ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse156922020.pdf
https://doi.org/10.30534/ijatcse/2020/156922020

Petri Nets Generating Array Languages

M.I.Mary Metilda[1] , D.Lalitha [2]
[1,2] Department of Mathematics, Sathyabama Institute of Science and Technology,
Chennai, India.
[1]metilda81@gmail.com, [2]lalkrish2007@gmail.com

ABSTRACT

Petri net models have been characterised to create array languages. In array generating Petri nets arrays are utilised as tokens in the initial places, instead of black dots. Triangular exhibits and triangular models are generally found in the literature in writing on picture preparing and scene investigation. Picture age should be possible from multiple points of view in formal dialects. Petri net models to create rectangular exhibits and hexagonal picture languages are also seen in the literature. We use this information from the literature in Petri nets generating diamond arrays. The catenation of upper cone arrays and lower cone arrays to a diamond array results in a similar diamond array.

Key words: Petri net, array languages, diamond array, upper cone array and lower cone array.

1 INTRODUCTION

Petri nets were introduced in the year of 1962 by Carl Adam Petri.[1]. Petri net models to produce rectangular arrays have seen in [2, 5]. In such Petri nets arrays are taken as tokens that reside in the places. Transitions are assigned with conditions. So when an enabled transition fires it removes an array from its input place and catenates an array and deposits it in the output place. By defining a labelling feature for transitions over an alphabet, the set of all firing sequences, beginning from a selected preliminary marking to a finite set of terminal markings, produces a language over the alphabet [3,4]. Hexagonal picture languages were also generated by Petri nets [6,7, 8].
Petri nets were also been used in generation of the password [9, 10], multi factor authentication [15,17], ATM PIN [16], CAPTCHA and *I*CAPTCHA [12-14] generation. Colour Petri nets and timed colour Petri nets were introduced for password generation[11]. Password and authentication issues are addressed by various researchers in finding better solution towards rectification [18-21]. Motivated by these concepts we have defined a notion of diamond arrays generated by Petri nets. In this concept upper cone array and lower cone arrays over a given alphabet are taken as tokens. Transition labelling is assigned with upper/lower catenation rules. Firing the transitions continuously, removes the initial array from the input place and joins the upper cone array or lower cone array according to the condition assigned with the transition and moves it to all the output places.

All such arrays reaching the output place is called language produced by the Petri Net structure. We name the resulting array as Diamond array token Petri Net structure(DATPNS). We can apply these concepts in our life in tile pasting and kolam generation.

2. PRELIMINARIES

We first recall the basics in Petri nets,array generating Petri nets and see the notations used.

DEFINITION 2.1
We define Petri net asN = [P,T,I,O] where $P = \{p_i\}$ i = 1,2,3,.....n is a finiteof non-empty places. $T = \{t_j\}$,j = 1,2,3.....m is a non-empty set of transitions. Iis aninitial function from which an arc starts and end at the transition.O is an outgoing function to which an arc ends which run from the transition.

DEFINITION 2.2
A Petri net is said to be a marked Petri net $[M_0]$ when tokens are assigned in the initial place. During the implementation of Petri net, position of these tokens will be changing continuously.

In this paper Diamond array, upper cone and lower cone arrays are used as tokens.

3. ARRAY GENERATING PETRI NETS (AGPNS)

In AGPN models the tokens are arrays over a given set of alphabets.The firing in these models depends upon the conditions assigned to the transition in the net.

DEFINITION 2.3
An AGPNS is a 5-tuple $N' = [N, \Sigma, \sigma, \mu_0, F]$ where N is a Petri net structure Σ is the set of alphabets, σ set of conditions given to the transitions, μ_0 is the initial array which assigned in the initial place and F is the final set $[F \subseteq P]$.

3.1 Notations that are used

Let Σ be the finite set of symbols. Σ_s^{**} Denotesthe array which is got from the set of elements of Σ . The magnitude of a Diamond array is calculated as follows.Let S be a Diamond array of magnitude "n". Let S' be the array that we get after joining the upper cone and lower cone. Adding S' with S the length of the array that reaches the end places is n+2.It is an $m \times m$ square array in which some positions (elements) will be blank.

Example 1: Let the diamond array be $S = \begin{matrix} & a & \\ a & a & a \\ & a & \end{matrix}$ Size of this array is n=3 which is defined by its total no of rows or columns(which are equal). It is similar to an 3 x 3 square array in which the elements in the positions(1,1),(1,3),(3,1)and (3,3) are blank.

Similarly let S = $\begin{matrix} & & a & & \\ & a & a & a & \\ a & a & a & a & a \\ & a & a & a & \\ & & a & & \end{matrix}$ The size of the array is n = 5(i.e. n+2) (number of rows or column which are equal) It is similar to a 5 x 5 square array in which some elements in the places (1,1),(1,2),(1,4),(1,5),(2,1),(2,5),(4,1),(4,5),(5,1),(5,2),(5,4)a nd(5,5) are blank.

4. CATENATION RULES

We use the following four catenation rules. The symbols which are assigned to the transitions are defined as follows.

1. $\wedge$ catenates the array in the upper direction.
2. $\vee$ catenatesthe array in the lower direction.
3. <catenatesthe array in the right hand direction.
4. >catenatesthe array in the right hand direction.

The arrays used in the net are over the alphabet{a,b} which are defined as follows.

$S = \begin{matrix} & a & \\ a & a & a \\ & a & \end{matrix}$, $B_1 = \begin{matrix} a & & a \\ & a & \\ a & & a \end{matrix}$,

Diamond upper cone

$B_2 = \begin{matrix} a & & a \\ & a & \end{matrix}$ $B_3 = \begin{matrix} b \\ b \\ b \\ b \end{matrix}$

lower cone Left cone

$B_4 = \begin{matrix} b \\ & b \\ b \end{matrix}$

Right cone

It is defined as S $\wedge$ B_1which joins the upper cone B_1 with the array S that come from the inicial place and takes the resulting array to the final place.

Let us consider the following example. The array in the initial place be $S = \begin{matrix} & a & \\ a & a & a \\ & a & \end{matrix}$

Let the catenation rule is define as $\sigma(t) = (S \wedge B_1)$. It removes S from its input place and joins the upper cone to its top and pushes the resulting array to its output place.The resulting array reaching the output place is S =

$\begin{matrix} & & a & & \\ & a & a & a & \\ a & a & a & a & a \\ & & a & & \end{matrix}$

4.1 Lower Catenation Rule

It is defined as S $\vee$ B_2 which joins the lower cone B_2 with the array S that come from the initial place and deposits it to the final place.

For example let S = $\begin{matrix} & & a & & \\ & a & a & a & \\ a & a & a & a & a \\ & & a & & \end{matrix}$ is an array that come from the input place and B_2 is the lower cone array define as in fig.A . Let the catenation rule be defined as $\sigma(t) = (S \vee B_2)$ Which removes the array S from its input place and joins B_2 in its lower side . The resulting array which is moved to the output place is S = $\begin{matrix} & & a & & \\ & a & a & a & \\ a & a & a & a & a \\ & a & a & a & \\ & & a & & \end{matrix}$

Which is the diamond array of sizen = 5, in which some positions are empty.

4.2 Left Catenation Rule

We define the left catenation rule as S <B_3which joins the left cone with S that come from the input place . Consider the following example. Let S be the array in the start place $S = \begin{matrix} & a & \\ a & a & a \\ & a & \end{matrix}$ Let the catenation rule is define as $\sigma(t) = (S < B_3)$. It removes S from its input place and joins

the left cone to itsleft side and pushes the resulting array to its output place. The resulting array is $S = \begin{matrix} & & b & & \\ & b & a & & \\ b & a & a & a & \\ & b & a & & \\ & & b & & \end{matrix}$.

4.3 Right Catenation Rule

We define the right catenation rule as $S > B_4$ which joins the right cone B_4 with S that come from the input place . Consider the array in P_1 $S = \begin{matrix} & b & & \\ b & a & & \\ a & a & a & \\ b & a & & \\ & b & & \end{matrix}$. Let the catenation rule define as $\sigma(t) = (S > B_4)$. It removes S from its input place and joins the right cone to its right hand side and puts the resulting array to its end place. The resulting array is

$$S = \begin{matrix} & & b & & \\ & b & a & b & \\ b & a & a & a & b \\ & b & a & b & \\ & & b & & \end{matrix}$$

Which is also a diamond array of size n = 5, in which some positions are empty.

5. SET OF RULES ASSIGNED FOR FIRING

We discuss the following three enabled transitions in DATPNS.

(i) A transition 't' which is not assigned any condition will be firing only in the case that all its initial places have similar arrays as tokens. In all the other cases the transition will not fire.

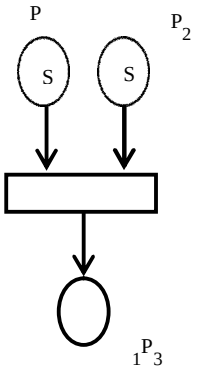

Figure 1 : Position of token before the implementation of the

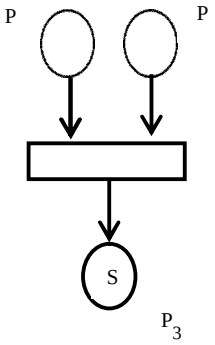

Figure 2 : Position of token after the implementation of the net

Fig.1 and fig.2 shows the position of tokens before and after the implementation of the net.

(ii) If the initialplaces of the transition do not have same arrays as tokens then the conditions to the transitions should be specified. If not the transition will not fire.When a transition fires it takes an array from the initial place and catenates the array specified with the transition and puts it in the output place.

6. PETRI NETS GENERATING DIAMOND ARRAY

Definition 3.1

Consider the Petri net model C = (P,T,I,O, Σ , μ_0 ,F) with diamond array over the set of alphabets Σ with the initial marking $\mu_0 : P \to \Sigma$ with the condition of at least one transition being Upper or lower cone rule and F is a finite set of end places $[F \subseteq P]$ then the Petri net is denoted as DATPNS.

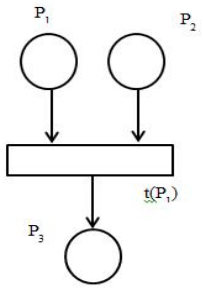

Fig.3 Position of tokens before firing with label

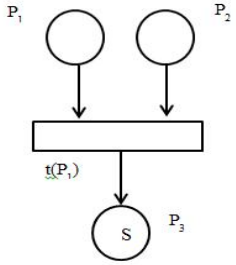

Fig.4 Position of tokens after firing with label

Definition 3.2

IfCisa DATPNS thenwedefineCas $L(C) = \{S \in \Sigma\}$ with arrays taken from the set of alphabets Σ ,and assigned in the initial place. All the conditions for the transitions are fixed. The group of arrays that are collected in the end place F are called the language produced by C.

The arrays which are used in the Petri net structure over the given set of alphabets reside in the input places. When all transitions fires, these arrays transformed from the initial place to any set of the final place. The language produced by the net is the collection of arrays arriving the final set of places.

6.1 Firing Rule

Let a transition't' have assigned a condition S*B, where * denotes any one of the following directions $\{\wedge, \vee, < and >\}$. S is a given Diamond array which is in the input place, B is predefined upperor lower or left or right cone . Then firing the transition removes S from its initial place and joins B and moves it to the output place

Example 2 Let $C' = [C, \Sigma, \sigma, \mu_0, F]$ be the net generating Diamond array, where C is the basic Petri net, Σ = {a} is the leater set, σ is the set of catenation rules assinged to the transitions, μ_0 is defined as the inceptive array assinged in the begining place. The array inceptive array in the begining place P_1 be S and $F = \{P_1\}$ is the sub set of final set. $I(t_1) = \{P_1\}$, $O(t_1) = \{P_2\}$, $I(t_2) = \{P_2\}$, $O(t_2) = \{P_3, P_1\}$. The arrays used in the net are taken from Fig.A.

7. PETRI NET GENERATING DIAMOND ARRAY

The starting array S is the Diamond shape of size 3 which is in the initial place P_1. On firing t_1 removes S from P_1 and joins the upper cone B_1 and put in the output P_2 enabling t_2 for firing. On firing t_2 removes the array from P_2 and joins the lower cone B_2 and deposits the resulting array in both P_3 and P_1. Moving the array to P_1 enables t_1 to fire.

The firing sequence $(t_1t_2)^n, n>0$ puts a Diamond spiral of size $2n+1$, $n>0$ in Place P3 and P1. The language generated by these DATPNs is a set of Diamond spirals. The resulting array is a Diamond array which is similar to an m x m matrix where m = 3, 5, 7, 9…….. in which some places are empty as explained earlier. The net which generate the diamond array is as follows.

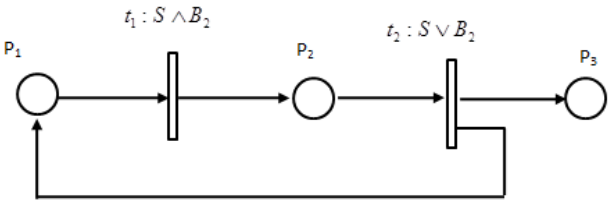

Figure 5: Petri net generating diamond array

8. THE RESULTING ARRAY GENERATED BY THE PETRI NET

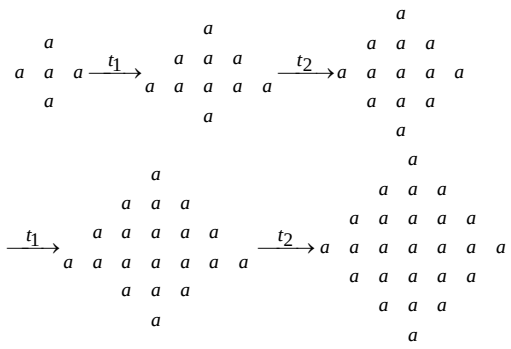

Figure 6: Diamond array

Example 3:Consider one more example. Let $S = \begin{matrix} & b & \\ b & b & b \\ & b & \end{matrix}$ which is in the place P_1 and $F = \{P_1\}$ is the subset of final set. $I(t_1) = \{ P_1 \}$, $O(t_1) = \{ P_2 \}$, $I(t_2) = \{ P_2 \}$, $O(t_2) = \{ P_3, P_4 \}$. $I(t_3) = \{ P_4 \}$ $O(t_3) = \{ P_5 \}$, $I(t_4) = \{ P_5 \}$, $O(t_4) = \{P_6, P_1\}$.

9. PETRI NET GENERATES THE ARRAY

The starting array S is the Diamond shape of size 3 which is in the initial place P_1. On firing t_1 removes S from P_1 and joins the upper cone B_1 and put in the output P_2 enabling t_2 for firing. On firing t_2 removes the array from P_2 and joins the lower cone B_2 and deposits the resulting array in both P_3 and P_4. On moving the array to P_4enables t_3 for firing. The arrays which are collected in P_1 will be a spiral of diamonds over an alphabet {b}. When t_3 fires, the array from P_4 is joined with B_3 in the left hand side and puts it in P_5. As soon as an array reaches P_5 it enables t_4 for firing. When t_5 fires B_4 is joined in the right hand side direction and puts the array in P_1 and P_6. An array reaching P_1 again enables t_1 for firing and so on. The resulting array in P_3 and P_6 are diamond spirals over the alphabets {a, b} alternatively. On firing $(t_1t_2t_3t_4)^n$, n = 1, 2, 3 …..puts a Diamond pattern of dimension $2n+1$, $n>0$ in Place P_3 and P_6. The language produced by by these DATPNs is a set of Diamond patterns.

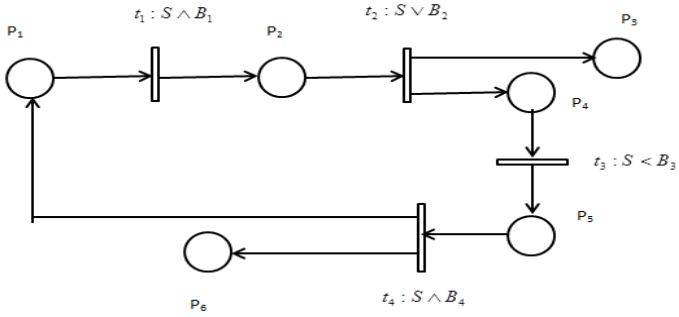

Figure 7: Petri nets generating Diamond array

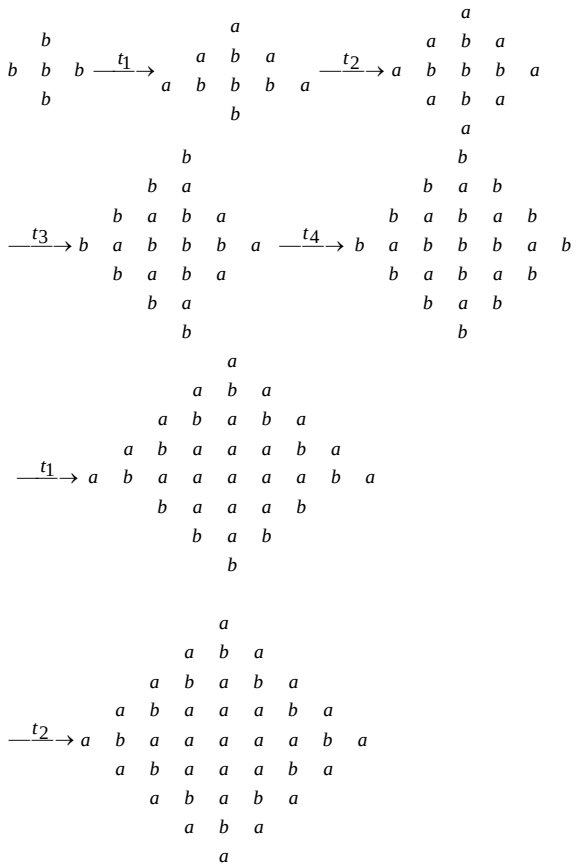

Figure 8: Diamond array

The resulting array is a Diamond array over a set of alphabets {a, b} which is similar to an m x m matrix where m = 3, 5, 7, 9…….. In which some places are empty as explained earlier. The Petri net producing the array is given below. The Resulting Array Generated by the Petri Net:

Theorem 4.1

Any regular diamond array language can be generated by DATPNS.

Proof:

If we are able to generate any language by a regular matrix grammar then we can generate it by a regular array grammar. These regular array grammars are able to generate by an ATPNS(array token Petri net structure). Therefore the diamond array language can be generated by Diamond array token Petri net structure (DATPNs). The firing sequences in DATPNs will be in the order $(t_1 t_2)^n$, $n > 0$. Therefore any regular diamond array language can be produced by DATPNS.

10. CONCLUSION

Petri net models have been characterized widely to produce array languages. In this Paper we have defined DATPNS which generates diamond array languages. Transitions are assigned a condition as upper or lower cone. On firing these transitions, upper and lower cones are catenated with the array that come from the input place and generate a diamond array language as a result.

REFERENCES

1. Peterson JL, *Petri Net theory and modelling of systems*, Prentice-Hall; Englewood Cliffs; 1981.
2. Lalitha D, Rangarajan K, Column and row catenation petri net systems, in Proceedings of Fifth IEEE International Conference on Bio-Inspired Computing: Theories and Applications; p. 1382-1387,2010. https://doi.org/10.1109/BICTA.2010.5645063
3. Lalitha. D, Rectangular array languages generated by a Colored Petrinet, Proceedings of IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2015. https://doi.org/10.1109/ICECCT.2015.7226095
4. Lalitha, D. Rectangular array languages generated by a Petri net,Advances in Intelligent Systems and Computing ,volume 332,Computational Vision and Robotics, pp 17-27.
5. D. Lalitha, K. Rangarajan, D.G. Thomas, Rectangular arrays and Petri nets, International Workshop on Combinatorial Image Analysis, Lecture Notes in Computer Science, volume 7655,pp 166-180. https://doi.org/10.1007/978-3-642-34732-0_13
6. D. Lalitha, K. Rangarajan, Petri Net Generating Hexagonal Arrays, International Workshop on Combinatorial Image Analysis, Lecture Notes in Computer Science, volume 6636,pp235-247. https://doi.org/10.1007/978-3-642-21073-0_22
7. Lalitha D, Rangarajan K, Thomas DG, Petri net generating hexagonal arrays, J.K. Aggarwal et al. (eds.) IWCIA 2011, *LNCS* 6636, p. 235-247, 2011, Springer-Verlag, Berlin, Heidelberg.
8. S. Kuberala, T.Kalyani, D.G.Thomasc, T.Kamaraj, Petri Net Generating Triangular Arrays 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015) Procedia Computer Science 57 (2015) 642 – 649. https://doi.org/10.1016/j.procs.2015.07.430
9. S.Vaithyasubramanian, A. Christy, D. Lalitha "Generation of Array Passwords Using Petri Net for Effective Network and Information Security" Advances in Intelligent Systems and Computing, ISSN: 2194-5357 Springer India, Vol.1, July 2014 pp189 – 200. https://doi.org/10.1007/978-81-322-2012-1_20.
10. S.Vaithyasubramanian, A. Christy, D. Lalitha "Two factor Authentication for Secured Login Using Array Password Engender by Petri net" Procedia Computer Science, Elsevier, ISSN: 1877-0509, Vol 48, 313 –318, 2015. https://doi.org/10.1016/j.procs.2015.04.187.
11. D. Lalitha, S. Vaithyasubramanian, K. Vengatakrishnan A. Christy, M. I. Mary Metilda (2018). A Novel Authentication Procedure for Secured Web Login using Coloured Petri Net. International Journal of Simulation: Systems, Science & Technology (Vol. 19, No. 6). DOI 10.5013/IJSSST.a.19.06.33.
12. Vaithyasubramanian, S. (2017). Array CAPTCHAs (Completely Automated Public Turing test to tell Computer and Human Apart). Asia Life Sci.-The Asian Int. J. Life Sci., 1, 247-254.
13. Vaithyasubramanian, S., Lalitha, D., & Kirubhashankar, C. K. (2019). Enhancing website security against bots, spam and web attacks using *l*CAPTCHA. International Journal of Computers and Applications, 1-7. https://doi.org/10.1080/1206212X.2019.1702285
14. Sai Kirthiga. G, Vaithyasubramanian. S (2016). Review on development of some strong visual CAPTCHAs and breaking of weak audio CAPTCHAs. 2016 IEEE International Conference on Information Communication and Embedded Systems, ICICES 2016, (Icices), 7–10. https://doi.org/10.1109/ICICES.2016.7518939.
15. Vaithyasubramanian. S, Christy. A & Saravanan. D (2015). Two factor authentications for secured login in support of effective information preservation and network security. ARPN Journal of Engineering and Applied Sciences, 10(5).
16. Vaithyasubramanian, S., & Christy, A., 2019. ATM PIN generation - a formal mathematical model to generate PIN using regular grammar, context free grammar and recognition through finite state machine, pushdown automata. Inderscience, International Journal of Internet Protocol Technology, Vol.12, No.1, pp.11-15, doi: 10.1504/IJIPT.2019.098485.
17. Vaithyasubramanian, S., Christy, A., & Saravanan, D. (2016). Access to network login by three-factor authentication for effective information security. The Scientific World Journal, 2016. https://doi.org/10.1155/2016/6105053
18. A. Christy , S.Vaithyasubramanian, Viji Amutha Mary , Naveen Renold J. (2019), "Artificial Intelligence based

Automatic Decelerating Vehicle Control System to avoid Misfortunes", International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No.6, Pp. 3129-3134.
https://doi.org/10.30534/ijatcse/2019/75862019

19. Jerome P. Songcuan , Ariel M. Sison , Ruji P. Medina. (2019), "Towards Usability Evaluation of Jumbled PassSteps", International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No.4, Pp. 1032-1037.
https://doi.org/10.30534/ijatcse/2019/08842019

20. Gandhimathi Amirthalingam , Harrin Thangavel. (2019), "Multi-Biometric Authentication Using Deep Learning Classifier for Securing of Healthcare Data", International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No.4, Pp. 1340-1347.
https://doi.org/10.30534/ijatcse/2019/48842019

21. Muawia A. Elsadig , Abdulrahman Altigani , Mohammed Abuelaila Ali Baraka. (2019), "Security Issues and Challenges on Wireless Sensor Networks", International Journal of Advanced Trends in Computer Science and Engineering, Vol. 8, No.4, Pp. 1551-1559.
https://doi.org/10.30534/ijatcse/2019/78842019