Int. J. of Mathematical Sciences and Applications,
Vol. 1, No. 3, September 2011
Copyright © Mind Reader Publications
www.journalshub.com

A Graph Theoretic Approach: Petri net

Sunita Kumawat
Department of Applied Mathematics,
ASET, Amity university Haryana, Manesar
ksunita86@gmail.com skumavat@ggn.amity.edu

Abstract

This work attempts to understand some of the basic properties of Petri nets and their relationships to directed bipartite graphs. Different forms of directed graphs are widely used in computer science. Normally various names are given to these structures. E.g. directed acyclic graphs (DAGs), control flow graphs (CFGs), task graphs, generalized task graphs (GTGs), state transition diagrams (STDs), state machines, etc. Some structures might exhibit bi- similarity. The justification for this work is that Petri nets are based on graphs and have some similarities to them. Transforming Petri nets into graphs opens up a whole set of new interesting possible experimentations. Normally this is overlooked. Directed Graphs have a lot of theory and research associated with them. This work could be further developed and used for Petri net evaluation. The related works justifies the reasoning how and why Petri nets are obtained or supported using graphs. The transformation approach can be formal or informal. The main problem tackled is how graphs can be obtained from Petri nets. Possible solutions that use reduction methods to simplify the Petri net are presented. Different methods to extract graphs from the basic or fundamental Petri net classes are explained. Some examples are given and the findings are briefly discussed.

Keywords: Graphs Petri Nets, Transformation, Reduction

INTRODUCTION

Petri Nets are expressive graphical formalisms that serve to model discrete event behavior that takes place in different systems [12]-[15]. They are designed to model system behavior like: sequential behavior, concurrency, mutual exclusion, non-determinism, choice and conflict. Petri nets are classified into different classes ranging from elementary nets to higher order nets, colored Petri nets and object oriented nets. All these classes can be converted to time Petri nets. Ordinary Petri nets have a 'dual identity' they can be represented graphically or by using equations. These can be analyzed using mathematical models. Petri nets have at least three decades of use. Normally speaking, the analysis of Petri nets is based on i) structural properties and ii) behavioral properties [6]. The structural properties of Petri nets are suitable to understand the basic underlying structure. If the Petri net is viewed, basic structural features can be seen. E.g. the Petri net can be cycle free (acyclical) [9]. It could have bounded places, etc. On the other hand behavioral properties explain the behavior of the Petri net. These properties cannot be applied to all types of Petri nets especially if the net is unbounded or improperly designed. Some basic behavioral properties are i) reachability, ii) boundedness, iii) safeness, iv) conservativeness, v) liveness, vi) reversiblility, vii) repetitiveness, viii) home states. One of the salient points for using Petri nets is precisely the ability to transform them or obtain them from other formalisms or notations. Petri nets are classified as directed bi-partite graphs, definitely sharing some common properties with graphs. This means that they could be transformed into graphs and analyzed from this point of view. The work in this paper is restricted to the basic or fundamental classes of Petri nets.

PRELIMINARIES

A normal Petri net is basically defined as directed bipartite graph or bipartite digraph that can be basically represented as a five tuple A place-transition net (PT-Net) is a quadruplet $PN = \langle P, T, F, W, M_0\rangle$, where $P = \{p_1, p_2 . . . p_m\}$ is the set of places, $T = \{t_1, t_2, . . . , t_m\}$ is the set of transitions, such that $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs and W: $F \rightarrow \{1, 2 ...\}$ is the weight function. M_0 represents the initial marking

PROBLEM FORMULATION

The main problem that is dealt with in this paper is to try to examine how Petri nets can be converted into graphs for the purpose of analysis. It is possible to transform Petri nets into graphs. There are different ways how to obtain graphs from Petri nets. To obtain graphs from the Petri net the Petri net should have a reduced structure and has to be bounded. A possible solution it to reduce the class and structure of the Petri net before applying analysis methods and transformation of the Petri net structure into a graph. Sometimes there can be various issues, especially if the Petri net is too complex. It has to be reduced. I.e. it can be reduced structurally to a simpler model or class reduction could be performed. E.g. a more complex class can be reduced to a lower class by replacing or eliminating some properties or information.

PROBLEM SOLUTION

There are two aspects of the solution i.e. i) reduction of the Petri net and ii) explaining the possible transformations that can be done. Reduction might imply i) class reduction and ii) structural reduction. Once the Petri net is reduced it is possible to transform the Petri net to a graph by simply replacing the nodes and edges in the Petri net. Another simple way of obtaining a graph from a Petri net is by generating the marking graph or the reachability graph. Other possible methods could include replacing the Petri net node and edges.

A. Reducing the Petri net

Two possibilities are given. I.e. i) class reduction and ii) structural reduction. As previously stated, the best Petri nets used to obtain graphs have to be structurally reduced and limited.

B. Class Reduction

Class reduction can be defined as transformation of a higher order Petri net or net into a simpler net or a more basic class type. There are various ways how this transformation can be carried out. Normally class reduction necessitates the loss of information [11]. The resulting Petri net is simplified and it is more comprehensible and simple for transforming it into a directed graph. The basic structural features of the Petri net should still be retained. This might not always be the case.

C. Basic Structural Reduction

According to well known Petri net theory [5]-[7] it is possible to classify five or six main rules for Petri net reduction whilst retaining the main properties. Basically a subnet or structures of the Petri net are reduced or simplified. These basic reduction rules obtained in part from [5]. These rules are applicable to various Petri net classes. Normally applying these results in the loss of information, one issue is that these rules are applicable to basic constructs. If there are advanced constructs like inhibitor arcs, test arcs, conflict, choice, etc. reduction is not so simple. Other more complex rules can be specified if required. It is possible to use ideas from decomposition with Petri nets. I.e. a place can contain an entire subnet at a lower level.

i) Serial Place Fusion/Reduction: This rule refers to combining two sequential or serial places into one single place. Normally the given pattern is that of a place followed by an output arc to a transition. An outgoing arc connects to another place. then replace the places and shared output/input transition with one single place.

ii) Serial Transition Fusion/Reduction: This rule is similar to the Serial Place Fusion/Reduction but in this case serial transitions are considered. Here two sequential transitions are combined into a single transition. Normally the given pattern is that of a transition connecting to a place which connects to another transition in series. Then replace the transitions and shared output/input place with one single transition.

iii) Parallel Place Fusion/Reduction: This rule is applied to two places which connect to the same output transition and the same input transition. I.e. they are parallel places. This rule can be extended to more than two places which might be in parallel, then transform the sub graph by replacing all bounded parallel places with a single place which is the output place of the first transition and input place of the second transition.

iv) Parallel Transition Fusion/Reduction: The idea behind this rule is similar to iii) Parallel Place Fusion/Reduction. Here instead of parallel places, parallel transitions are considered. Parallel transitions normally connect to the same input and output places. This rule can similarly be extended to more than two transitions in parallel.

CONVERSION TO DIRECTED GRAPHS

Four different ways to convert a Petri net to a directed graph are listed and explained below.

A. Transitions as nodes. Places, Input/Output Arcs as Edges

Here the Petri net is converted into a graph and the transitions are replaced using nodes, whilst the places are their connecting input and output arcs are replaced with a single graph edge. The transformation of places into nodes and connecting transitions to edges works well only if places have single input points and single output (exit) points. For this type of conversion, it must be clearly explained that every place in the Petri net should have exactly one input arc and one output arc. If these conditions do not exist it is not possible to create a proper directed graph.

B. Places as nodes. Transitions, Input/Output Arcs as Edges

Here the Petri net is converted into a graph. The transitions and their connecting arcs are replaced using a single edge, places are replaced with nodes. For this type of conversion it must be clearly stated that a transition must have exactly one input arc and one output arc. If these conditions do not exist it is not possible to create a proper directed graph. The transformation of places into nodes and connecting transitions to edges works well only if places have single input points and single output (exit) points. Thus augmented marked graph Petri nets (AMG) or state-machine like Petri nets, etc. can easily be converted. Other more complex Petri net classes have to be reduced for conversion to be possible. Reduction methods presented should simplify the structure of the net.

C. Places and Transitions as Nodes, Input/Output Arcs as Edges

In this approach, when converting the Petri net the input and output arcs are replaced as graph edges and the places and transitions are replaced as nodes. It can be important to properly label the nodes.
I.e. in this case, there is not really any complex structural change in the Petri net. It is just that the places and transitions are represented as similar nodes. This implies that there is not any real change to the Petri net, it is only the structural representational notations of the net that have been modified.

D. Marking Graph

If the Petri net is structurally limited and its state space is limited (i.e. its markings are finite) it is possible to construct the marking graph for the net [6], [7]. Different algorithms can be used to construct the marking graph. In simple terms the marking graph represents all the possible states of the Petri net. The reachability tree or better known coverability tree for a restricted Petri net is easily constructed. The directed graph obtained from the Petri net can be used for different forms of analysis, which is often overlooked. This type of graph can become quite large if the Petri net has over one hundred states. A possible solution is to reduce the Petri net model and eliminate ambiguity.
The marking graph or reachability tree is a simple directed graph or digraph where the nodes or vertices represent a marking whilst the directed edges represent the transitions used to reach a particular marking. The reachability tree can be drawn as a marked directed graph of the form G = (V, E), where E = edges representing transitions and V= States or markings. There are various forms of the marking graph having different names but in essence they are similar. The delay of a transition can be represented on the edges. For the marking graph an adjacency matrix can be constructed.

EXAMPLES

Some simple examples illustrating the four conversion methods are given. A specific Petri net is taken and a corresponding directed graph is constructed using each one of the four methods. It can be assumed that the structural reduction rules previously defined have been applied as required. These are quite simple to comprehend and are self explanatory. Note that the resulting graph can obviously be drawn as required, i.e. the node or edge layout could be drawn aesthetically in different ways for visualization e.g. using rounded or flat edges, circles for nodes, etc. Fig. 1(a) shows a Petri net, complete with its marking graph Fig. 1(b) and a compacted or reduced form of marking graph Fig.1(c). Below the reduced marking graph the adjacency matrix for the marking graph has been given. For the marking graph in Fig. 1(a), the adjacency matrix is easily constructed. It is assumed that edges represent transitions.

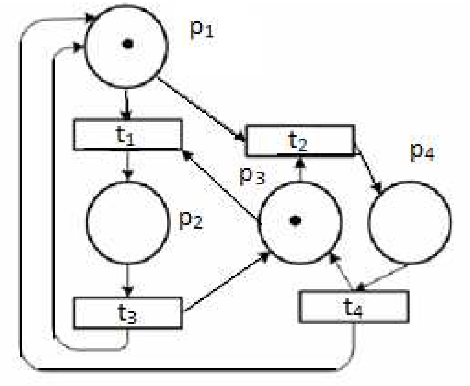

Figure 1(a)

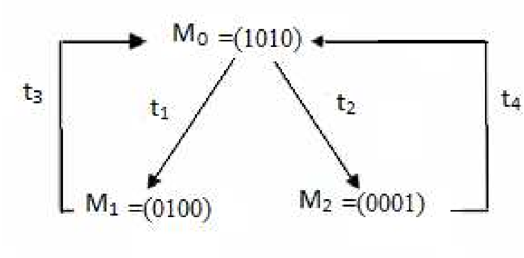

Figure 1(b)

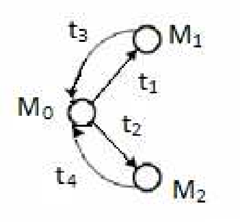

Figure 1(c)

$$\begin{pmatrix} 0 & t_1 & t_2 \\ t_3 & 0 & 0 \\ t_4 & 0 & 0 \end{pmatrix}$$

Figure 1(d)

RESULTS

For the marking graph in Fig. 1(a) the adjacency matrix can be constructed, given in Fig. 1(d). It is assumed that edges represent transitions . The given directed graph can be analyzed using the usual graph analysis techniques.

CONCLUSION AND FURTHER WORK

The limitations of this work are that here only basic or simple Petri nets have been considered for conversion into graphs. In reality only Petri nets that have a limited number of states or limited in structure can be easily converted.
The approach presented here shows only one way of looking at the Petri nets. The approach is applicable to certain classes of Petri nets like elementary nets, cause event nets (CE nets), state machine nets, augmented marked graph Petri nets and others, where the net has limited structure and deterministic behavior. It is possible to find that some structures are isomorphic to other structures derived from different conversion methods of the Petri net to directed graphs.

References

1. E. Kindler, R. Wagner, Triple Graph Grammers: Concepts, Extensions, Implementations and Application Scenarios, Technical Report TR-RI- 284, University of Paderborn, Paderborn, 2007, Available: http://www2.cs.uni-paderborn.de/cs/ag- schaefer/Veroeffentlichungen/Quellen/Papers/2007/tr-ri-07-284.pdf
2. A. Sathaye, B. Krogh, "Supervisor Synthesis for Real-Time Discrete Event Systems", Discrete Event Dynamic Systems, vol. 8 , issue 1, Springer, 1998, pp. 5 - 35.
3. A. Abellard, P. Abellard, Systolic Petri Nets, Petri Nets Applications, Pawel Pawlewski (Ed.), ISBN: 978-953-307-047-6, INTECH(2010), http://www.intechopen.com/articles/show/title/systolic-petri-nets
4. K.S. Cheung, K.O. Chow, "Compositional Synthesis of Augmented Marked Graphs", Control and Automation, ICCA, IEEE, 2007, pp. 2810- 2814.
5. M.B. Dwyer, L.A. Clarke, "A compact Petri net representation and its implications for analysis", IEEE Transactions on Software Engineering, vol. 22, issue 11, 1996, pp. 794 - 811.
6. T. Murata, "Petri nets: Properties, analysis and applications", Proc. of IEEE, vol.: 77, issue:4, 1989, pp. 541-580.
7. M. Zhou, K. Venkatesh, Modeling, Simulation, and Control of Flexible Manufacturing Systems A Petri Net Approach, Series in Intelligent Control and Intelligent Automation vol. 6 World Scientific, MA USA, 1999.
8. T. Gehrke, U.Goltz, H. Wehrheim, The Dynamic Models of UML: Towards a Semantics and its Application in the Development Process, TR. 11/98, University of Hildesheim, Germany, 1998.
9. G. Stemersch, R.K. Boel, "Structuring acyclical Petri Nets for Reachability Analysis and Control", International Journal of Intelligent Control and Systems, vol. 10, no. 2, 2005, pp. 175-187.

10. X. Xiaoxi, L. Cheng-Chew Lim, Transfer-Resource Graph and Petri-net for System-on-Chip Verification, Petri Nets Applications, Pawel Pawlewski (Ed.), ISBN: 978-953-307-047-6, InTech, Available from: http://www.intechopen.com/articles/show/title/transfer-resource-graph- and-petri-net-for-system-on-chip-verification
11. A. Spiteri Staines, "Supporting Requirements Engineering with Different Petri Net Classes",International Journal of Computers, NAUN, issue 4 vol 4, 2010, pp. 215-222.
12. V.Vlad, C.Ciufudean, A.Graur, C.Filote, "An example of modeling manufacturing systems using Petri nets and the IEC 61499 standard", 13th WSEAS Int. Conf. on Systems, Greece, 2009, pp. 357-363.
13. H.A. Ozkan, A. Aybar, "A Reversibility Enforcement Approach for Petri Nets Using Invariants, WSEAS Transactions on Systems, vol. 7, issue 6, 2008, pp.672-681.
14. K. Mun Ng, Z. Alam Haron, "Visual Microcontroller Programming Using Extended S-System Petri Nets", WSEAS Transactions on Computers, issue 6, vol. 9, 2010, pp. 573-582.
15. P. Strbac, M. Tuba, D. Simian, "Hierarchical model of a systolic array for solving differential equations implemented as an upgraded Petri net", WSEAS Transactions on Systems, vol. 8, issue 1, 2009, pp. 13-21.