

TOWARDS Z SPECIFICATION OF PLACE TRANSITION NETS

ZAHEER AHMAD¹, F AHMAD², MUDASSER NASEER³

¹IT Department, Govt. College of Commerce Shahdara, Lahore, Pakistan

^{2,3}Department of Computer Science, University of Central Punjab, Lahore Pakistan

zaheer190@gmail.com

ABSTRACT: Petri net formalism has dynamics and it is well suited for distributed or concurrent systems. However, it has a graphical representation in the form of a bipartite graph through which the type of data cannot be identified. This paper presents the Z specification of the net structure of place transition nets to provide the data semantics of graphical structure. This paper further addresses the Z specification of special sub classes of Petri nets, which include state machine, marked graphs and free choice nets.

Keywords: Petri net, Z notation, formal specification

1. Introduction: Petri net is a prominent formalism to design and analyze distributed and concurrent systems. Petri net has established its importance in the domain of verifying the communication protocols, manufacturing systems, safety critical systems, and any kind of software system. Petri net is the name of a class of net based formal methods which include elementary nets as low level nets, place/transition nets as an intermediate level nets and colored Petri nets as a class of high level nets [1].

Further, low and intermediate level nets can only model the control flow of a system being modeled through these kind of nets. However, high level nets are able to represent the data involved in a system being modeled [2]. Petri nets have the power of execution and a Petri net model of a system can be simulated on a paper and the behavior of the model can be analyzed and verified [3]. However, data abstraction and modification can't be observed in Petri nets as it has a graphical representation and such a formalism is able to presents the control flow of a system to be modeled. On the other hand, Z notation has the power of specifying data and its progress in any kind of a software system [8]. However, Z notation based formal method has not the ability of execution and henceforth the dynamic behavior cannot be observed. Z formal method is suitable to describe data types, sets of operation, labeling and constraints of a system. Further, such kind of formal method has the tool support for type checking and for theorem proving. An effort has been presented in [6] to enhance the Petri nets with Z. Another research work addressed the combination of Z and Petri nets in the form of PZ nets [7]. Further, Z doesn't support concurrency and not suitable for parallel system, therefore, some efforts has been reported in the literature to represent concurrent models through Z notation [9], [10], [11].

In order to provide the data semantics of graphical structure of Petri nets, this paper addresses the Z notation based representation of a net structure to study the data type and constraints imposed in the form of invariants. Further, this paper elaborates the structural components of Petri nets in form of axioms or schemas. Therefore, this effort gives the research direction of representing the data, types and constraints of a system to be modeled through Petri net. Moreover, through the Petri net model, dynamic behavior.

Rest of the paper is organized as follows, Section 2 describes the basic concepts and notation about Petri nets and Z notation. Section 3 discusses the formal definitions, axioms and schemas of net structural components of place transition net. Section 4 presents a Z specification of special sub classes of Petri nets using the specification provided in Section 3 while Section 5 gives some conclusive remarks.

2 Basic Terminology and concepts: Z formal language is purely typed language while it is based on first order logic and set theory. Following concepts has been taken from [8].

- $X \times Y$ (Cartesian Product) is the set of all ordered pairs (x,y) such that x is an element of X and y is an element of Y ;
- $\mathbb{P}X$ (power set) is the set of all subsets of X ;
- $X \leftrightarrow Y$ is the set of all binary relations between X and Y , defined by $\mathbb{P}(X \times Y)$;
- $x \mapsto y$ is a ‘Maplet’ from x to y , an alternative representation of the ordered pair (x, y) ;
- **dom** R denotes the domain of R , defined by $\{x: X \mid \exists y : Y . x \mapsto y \in R\}$.
- **ran** R denotes the range of R , defined by $\{y: Y \mid \exists x : X . x \mapsto y \in R\}$.
- $R \circ S$ is the composition of R and S , defined by $\{x: X; z: Z \mid \exists y: Y \bullet (x, y) \in R \wedge (y, z) \in S\}$
- **id** X is the identity relation on X , defined by $\{(x, x) \mid x: X\}$
- $X \twoheadrightarrow Y$ is the set of all partial functions from X to Y , defined by $\{f: X \leftrightarrow Y \mid x: X, y_1, y_2: Y \bullet (x, y_1) \in f \wedge (x, y_2) \in f \Rightarrow y_1=y_2\}$
- $X \rightarrow Y$ is the set of all total functions from X to Y , defined $\{f: X \leftrightarrow Y \mid \text{dom } f = X \bullet f\}$

Definition 1(Low level nets) [3] A Petri net (N, M_0) is a net PN with five tuples (P, T, F, W, M) where

- P is a finite set of places; T is a finite set of transitions.
such that $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$;
- F is the flow relation (set of arcs) between places and tranistions
such that $F \subseteq (P \times T) \cup (T \times P)$;
- W is a weight function that maps positive number to set of flow relations.
such that $W = F \rightarrow \{1, 2, 3 \dots\}$ is no of arcs associate with places and transitions.
- M is a function that maps non negative integers to set of place such that $M: P \rightarrow N$ and $M(p)$ represents no. of tokens in any place and M_0 is called initial marking of net.

3 Z Specification of the structure of Place Transition net

As given in Definition 1, that there are three structural components of place transition net, i.e. a place represented by circle, a transition represented by square and an arc (see Fig. 1). Therefore, a place transition net is a bipartite graph [5] which can be described through three sets which are, a set of places, a set of transition and a set of arcs (which can be described through a relation).

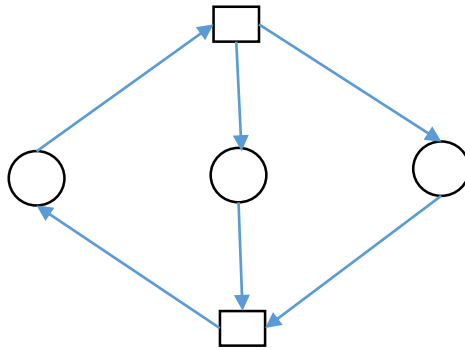


Figure 1: it shows the structure of a place transition net

Now the specification of the net structure is given as:

$[P, T]$

Set of arcs can be described in the form of a relation of type $(P \times T) \cup (T \times P)$:

$\text{Arc: } \mathbb{P}((P \times T) \cup (T \times P))$
$\text{domArc} = \{p:P; t:T \mid p \rightarrow t \in \text{Arc} \vee t \rightarrow p \in \text{Arc} \cdot p \vee t\} \Leftrightarrow \text{domArc} \subseteq P \cup T$ $\text{ranArc} = \{p:P; t:T \mid p \rightarrow t \in \text{Arc} \vee t \rightarrow p \in \text{Arc} \cdot p \vee t\} \Leftrightarrow \text{domArc} \subseteq P \cup T$

Pre-set of a place $p \in P$ is the set of those transitions having outgoing arcs to the p . Now we can specify pre-set for each place in P :

$\text{Pre}_p: \mathbb{P}T$ $p: P$
$\text{Pre}_p = \text{dom}(\text{Arc} \triangleright \{p\}) \Leftrightarrow \text{Pre}_p = \{p:P; t:T \mid t \rightarrow p \in \text{Arc} \cdot t\}$

Post set of a place $p \in P$ is the set of those transitions having incoming arcs from that p , Z specification of post set is give as:

$\text{Post}_p: \mathbb{P}T$ $p: P$
$\text{Post}_p = \text{ran}(\{p\} \triangleleft \text{Arc}) \Leftrightarrow \text{Post}_p = \{p:P; t:T \mid p \rightarrow t \in \text{Arc} \cdot t\}$

Similarly, preset of a transition $t \in T$ is the set of places having outgoing arcs to that t . Further, preset of a transition can be represented in Z specification, given below:

$\text{Pre}_t: \mathbb{P}P$ $t: T$
$\text{Pre}_t = \text{dom}(\text{Arc} \triangleright \{t\}) \Leftrightarrow \text{Pre}_t = \{p:P; t:T \mid p \rightarrow t \in \text{Arc} \cdot p\}$

Post set of a transition $t \in T$ is the set of those places having incoming arcs from that t , Z specification of post set of t is give as:

$$\begin{array}{l}
\text{Post}_t \\
\text{Post}_t: \mathbb{P}P \\
t: T \\
\hline
\text{Post}_t = \text{ran}(\{t\} \triangleleft \text{Arc}) \Leftrightarrow \text{Post}_t = \{p: P; t: T \mid t \rightarrow p \in \text{Arc} \cdot p\}
\end{array}$$

Further, in order to capture the dynamics of place transition nets, the enabling of a transition can be described in \mathbb{Z} , which is given below:

$$\begin{array}{l}
\text{Enabled}_t \\
\text{Marking}: P \rightarrow \mathbb{N} \\
\text{Arc}_W: \text{Arc} \rightarrow \mathbb{N} - \{0\} \\
\hline
\forall p: \text{Pre}_t \cdot \text{Marking}(p) \geq \text{Arc}_W(p, t)
\end{array}$$

By firing an enabled transition, new marking can be obtained as:

$$\begin{array}{l}
\text{New_Marking} \\
\text{Marking}: P \rightarrow \mathbb{N} \\
\text{Arc}_W: \text{Arc} \rightarrow \mathbb{N} - \{0\} \\
\hline
\text{New_Marking} = (\forall p: \text{Pre}_t \cdot \text{Marking}(p) - \text{Arc}_W(p, t)) \wedge (\forall p: \text{Post}_t \cdot \text{Marking}(p) + \text{Arc}_W(t, p)
\end{array}$$

4 Sub Classes of Place Transition nets

State Machine: it is an ordinary place transition net [4] where arc weight is one for each arc of the net.

Further, every transition as one and only one input place and one and only one output place. Further, \mathbb{Z} specification for a state machine is given as:

$$\begin{array}{l}
\text{State_Machine} \\
\text{Arc}_W: \text{Arc} \rightarrow \{1\} \\
\hline
\forall t: T \cdot \#\text{Pre}_t = \#\text{Post}_t = 1
\end{array}$$

Marked graph: in a marked graph each place has one and only one input transition as well as one and only one output transition, its specification is given below:

Marked_Graph

$$Arc_W: Arc \rightarrow \{1\}$$

$$\forall p: P \bullet \#Pre_p = \#Post_p = 1$$

Free choice net: in this type of a special net, either each place has one output transition or each transition has one

Free_Choice

$$Arc_W: Arc \rightarrow \{1\}$$

$$\forall p: P \bullet \#Post_p = 1 \vee \forall t: T \bullet \#Pre_t = 1$$

Conclusion: Petri net is a famous formal technology for systems engineering which has the power of execution and simulation. Further, Z specification language is a typed language with gives the data abstraction and refinement. Therefore there was a need to provide the data semantics of graphical structure of Petri nets. This paper addressed the Z notation based representation of a net structure to study the data type and constraints imposed in the form of invariants. Further, this paper elaborates the structural components of Petri nets in form of axioms or schemas. Therefore, this effort gives the research direction of representing the data, types and constraints of a system to be modeled through Petri net. Moreover, through the Petri net model, dynamic behavior.

REFERENCES:

- [1] Thiagarajan, P. S. (1987). Elementary net systems. In Petri nets: central models and their properties (pp. 26-59). Springer Berlin Heidelberg.
- [2] Jensen, K., Kristensen, L. M., & Wells, L. (2007). Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. International Journal on Software Tools for Technology Transfer, 9(3-4), 213-254.
- [3] Desel, J., & Reisig, W. (1998). Place/transition Petri nets. In Lectures on Petri Nets I: Basic Models (pp. 122-173). Springer Berlin Heidelberg.
- [4] Murata, T. (1989). Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 77(4), 541-580.
- [5] Ahmad, F., & Khan, S. A. (2013). Specification and verification of safety properties along a crossing region in a railway network control. Applied Mathematical Modelling, 37(7), 5162-5170.
- [6] Heiner, M., & Heisel, M. (1999). Modeling safety-critical systems with Z and Petri nets. In Computer Safety, Reliability and Security (pp. 361-374). Springer Berlin Heidelberg.
- [7] He, X. (2001). PZ nets—a formal method integrating Petri nets with Z. Information and Software Technology, 43(1), 1-18.
- [8] Woodcock, J., & Davies, J. (1996). Using Z: specification, refinement, and proof. Prentice-Hall, Inc.
- [9] Duke, R., & Smith, G. (1989). Temporal logic and Z specifications. Australian Computer Journal, 21(2), 62-66.
- [10] Clarke, E. M., & Wing, J. M. (1996). Formal methods: State of the art and future directions. ACM Computing Surveys (CSUR), 28(4), 626-643.
- [11] Evans, A. S. (1997). An improved recipe for specifying reactive systems in Z. In ZUM'97: The Z Formal Specification Notation (pp. 275-294). Springer Berlin Heidelberg.