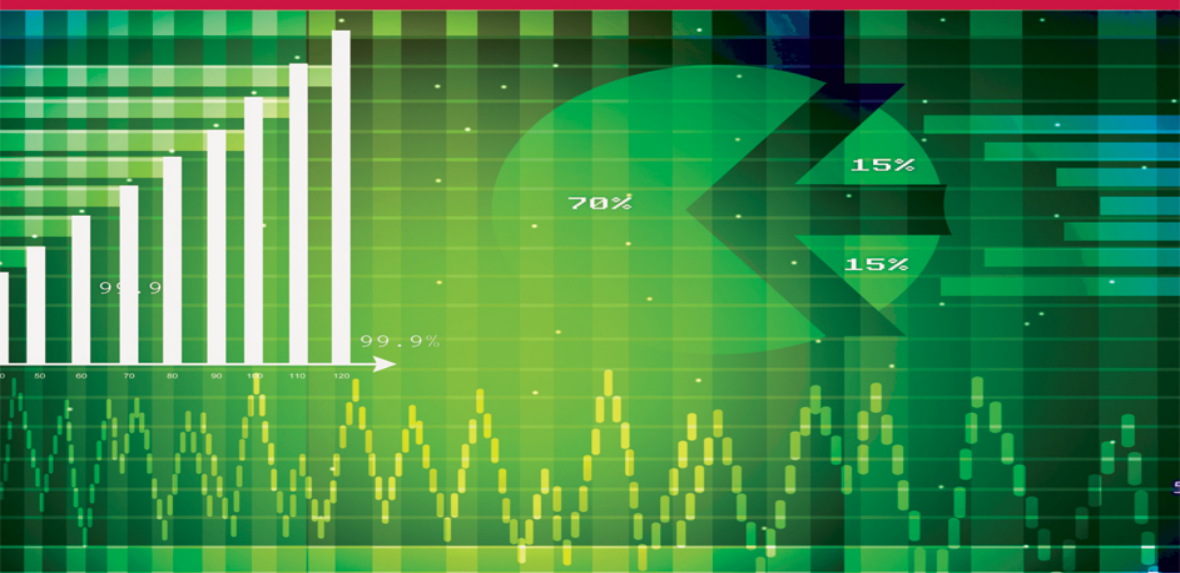


SYSTEMS AND INDUSTRIAL ENGINEERING SERIES

SYSTEMS DEPENDABILITY ASSESSMENT SET



Volume 1

Systems Dependability Assessment

Benefits of Petri Net Models

**Jean-François Aubry
Nicolae Brinzei
Mohammed-Habib zouni**

ISTE

WILEY

Systems Dependability Assessment

Systems Dependability Assessment Set

coordinated by
Jean-François Aubry

Systems Dependability Assessment

Benefits of Petri Net Models

Jean-François Aubry
Nicolae Brinzei
Mohammed-Habib Mazouni

ISTE

WILEY

First published 2016 in Great Britain and the United States by ISTE Ltd and John Wiley & Sons, Inc.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms and licenses issued by the CLA. Enquiries concerning reproduction outside these terms should be sent to the publishers at the undermentioned address:

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

© ISTE Ltd 2016

The rights of Jean-François Aubry, Nicolae Brnzei and Mohammed-Habib Mazouni to be identified as the authors of this work have been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

Library of Congress Control Number: 2015960014

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN 978-1-84821-991-5

Contents

Introduction	xi
Part 1. Short Review of Petri Net Modeling	1
Introduction to Part 1	3
Chapter 1. Autonomous Petri Nets	5
1.1. Unmarked Petri nets	5
1.1.1. Definitions	5
1.1.2. Drawing	6
1.1.3. Other definitions	7
1.2. Marking of a PN	7
1.2.1. Order relation on markings	8
1.2.2. Enabled transition	9
1.3. Dynamics of autonomous PNs	9
1.3.1. Firing of a transition	9
1.3.2. Transition matrix	11
1.3.3. Firing sequence	11
1.3.4. Reachable marking	12
1.3.5. Fundamental equation	12
1.3.6. Properties of PN	14
1.3.7. Other properties	14
1.3.8. Invariants in a PN	15
1.3.9. Reachability graph	16

Chapter 2. Petri Nets and Event Languages	19
2.1. Labeled PNs	19
2.1.1. Formal definition	19
2.1.2. Generated and marked languages	20
2.2. Example	21
Chapter 3. Comparison Petri Nets – Finite State Automaton	25
3.1. Language expression	26
3.2. Building of the models	27
3.2.1. Synchronization of submodels	28
3.2.2. Resource sharing	29
3.2.3. Construction by refinement	30
3.3. Compactness of the model	32
Chapter 4. Some Extensions of Petri Nets	35
4.1. PN with inhibitor arcs	35
4.2. Timed PN	36
4.2.1. P-timed Petri nets	37
4.2.2. T-timed Petri nets	37
4.3. Synchronized PN	38
4.4. Timed synchronized PN	40
4.5. Interpreted PN	41
4.6. Colored PN	42
4.6.1. Introduction example	42
4.6.2. Formal definition	45
4.6.3. A dedicated software CPN Tools	46
Conclusion to Part 1	51
Part 2. A Formal Approach to Risk Assessment	53
Introduction to Part 2	51
Chapter 5. Ontology-based Accidental Process	61
5.1. Preliminary definitions	61
5.2. Elementary entities: HSE and VTE	63

5.2.1. Hazard supplier entity (HSE)	63
5.2.2. Vulnerable target entity (VTE)	63
5.3. Elementary situations and elementary events	64
5.3.1. State versus situation	64
5.3.2. Initial situation (IS)	64
5.3.3. Initiating event (IEv)	64
5.3.4. Hazard situation (HS)	65
5.3.5. Exposure event (EEv)	65
5.3.6. Exposure situation (ES)	65
5.3.7. Accident situation	65
5.3.8. Hazardous (feared) event (HEv)	65
5.4. Conclusion	66
Chapter 6. Petri Net Modeling of the Accidental Process	67
6.1. Elementary process	68
6.2. Sequence of elementary processes	71
6.3. Modeling the action of a safety barrier	71
6.4. Modeling of a cumulative process	73
6.5. PN as a support for risk assessment	75
6.5.1. Modeling of the damage	75
6.5.2. Modeling of the event frequencies	75
6.5.3. CPN Tools implementation	77
6.5.4. Evaluation rule of the risk	83
6.6. Conclusion	86
Chapter 7. Illustrative Example	87
7.1. Functional description	87
7.2. Building of an accidental process	88
7.2.1. First elementary process	88
7.2.2. Second elementary process	91
7.2.3. Parallel process	92
7.2.4. The whole model	92
7.3. Conclusion	94

Chapter 8. Design and Safety Assessment Cycle	95
8.1. Five essential steps	95
8.2. Ontological interest	98
Conclusion to Part 2	101
Part 3. Stochastic Petri Nets	103
Introduction to Part 3	105
Chapter 9. Basic Concept	107
9.1. Introductory example	107
9.2. Formal definition	108
Chapter 10. Semantics, Properties and Evolution Rules of an SPN	111
10.1. Conservatism properties	112
10.1.1. Conservatism of the mean marking in steady state	112
10.1.2. Conservatism of the flow in steady state	113
10.2. Mean sojourn time in a place of a SPN	113
10.3. Equivalent Markov process	114
10.4. Example of SPN for systems dependability modeling and assessment	116
Chapter 11. Simplification of Complex Models	121
11.1. Introduction	121
11.2. System modeling	122
11.3. Presentation of the quantitative analysis method	124
11.3.1. Steps to obtain an aggregated Markov graph	124
11.3.2. Toward a direct establishment of a reduced Markov graph	137
11.4. Example	137
11.4.1. Failure modeling	138
11.4.2. Study of the different functional and hardware solutions	139
11.4.3. Evaluation of the weighting coefficients from the Petri nets	144
11.4.4. Conclusion	147

Chapter 12. Extensions of SPN	149
12.1. Introduction	149
12.2. Relationship between stochastic Petri nets and stochastic processes	150
12.3. The transition firing policy	151
12.4. Associated stochastic processes	151
12.4.1. Temporal memory based on resampling	152
12.4.2. Temporal memory based on age memory or on enabling memory	153
12.4.3. Stochastic process underlying a stochastic PN	154
12.4.4. Embedded Markov chain of the stochastic process	157
12.4.5. Application to a case study	159
12.5. Synchronization problem in generalized stochastic Petri nets	162
12.5.1. GSPN with internal synchronization	162
12.5.2. SPN with predicates and assertions	164
12.6. Conclusion	168
 Part 4. Applications of Stochastic Petri Nets to Assessment Problems in Industrial Systems	 169
 Introduction to Part 4	 171
 Chapter 13. Application in Dynamic Reliability	 175
13.1. Presentation of the system and hypothesis	175
13.2. System modeling with Petri net	177
13.3. Methodology application	179
13.4. Construction of an aggregated Markov graph	180
13.5. Conclusion	185
 Chapter 14. Classical Dependability Assessment	 187
14.1. Availability study of a nuclear power plant subsystem	187
14.1.1. CPN modeling	188
14.1.2. Reliability and dependability assessment	192
14.1.3. Conclusion	196
14.2. Common causes failures in nuclear plants (safety oriented)	197

14.2.1. The Atwood model	197
14.2.2. Case study	199
14.2.3. Probabilistic dependability assessment	208
14.2.4. Conclusion	212
Chapter 15. Impact of Failures on System Performances	213
15.1. Reliability evaluation of networked control system	213
15.1.1. Statement of the problem	213
15.1.2. Reliability criteria of an NCS	215
15.1.3. Elements of modeling	216
15.1.4. Simulation and results	225
15.1.5. Evaluation of reliability	230
15.1.6. Conclusion	230
15.2. Railway signaling	231
15.2.1. Introduction	231
15.2.2. Interest	233
15.2.3. Signaling system specifications	234
15.2.4. Elements to be modeled	235
15.2.5. Architecture of the model	236
15.2.6. Example of an elementary model	237
15.2.7. Incident generation	239
15.2.8. Results	239
15.2.9. Conclusion	242
Conclusion	245
Appendix	247
Bibliography	251
Index	261

Introduction

In the first book of this series [AUB 15], finite state automata were introduced as an efficient model for the study of reliability and dependability of systems as well in static as in dynamic context. We saw that this type of model requires either an *a priori* exhaustive knowledge of the possible states of the system or its formal construction by operations starting from the models of its components. This is unfortunately sometimes not possible. For example, during the design of a system these states are not known in advance. It is however useful to make a predictive dependability assessment in order to select the best solution among some propositions. Petri nets may be an interesting way to answer such problems. Widespread in the field of automatic control, especially for the modeling of discrete event systems, Petri nets were introduced in the field of dependability studies with a noticeable success. The objective of this book is not to present all of the forms of Petri nets used in dependability assessment but instead to focus on the most interesting ones. Before their description, we give a preliminary formal description of the different successive models of Petri nets which led to the advent of their use in the dependability field. Of course, it is not just a matter of exhaustively describing the existing variants of the basic models which are today hardly countable. In the same way, we will not demonstrate all the mathematical properties of these models and we will refer the reader to the essential basic works on the subject. After the introduction of the

basic models called “autonomous Petri nets” and the comparison with the finite state automata especially in terms of event language expression, we will present the fundamental models of non-autonomous Petri nets to take account of the time and of an external environment, such models giving an opening to the study of hybrid systems. Relying on these timed and synchronized Petri nets, we will describe a systematic method of risk analysis based on an ontological approach whose elements are entities (supplier or target of hazard), their successive states and the events corresponding to these state changes. From the proposed model, a risk assessment may be deduced by simulation thanks to the introduction of random event generators. This approach is illustrated by an example from the railway transportation field. The need of models, integrating the stochastic character of elements (in this case, events) and allowing an analytical solution instead of simulation, leads to the introduction of stochastic Petri nets modeling and its equivalence conditions with Markov or some extensions of Markov models. We then show how, under some conditions, complex models may be simplified by a distribution of the global model on the two formalisms: stochastic Petri nets and Markov processes. Numerous extensions of Petri nets have been proposed; we recall the most significant ones and the conditions of their Markov process equivalence. To complete the book, we present some modeling examples using different available software tools. These examples are issued from different application domains.

Writing this book would not have been possible without the contribution of colleagues and of PhD and Master students who investigated some related aspects. All of these contributions have been the subject of publications and are referenced in the text. We would like to extend our thanks to G. Babykina, P. Barger, G. Deleuze, L. Gérard, R. Ghostine, D. Jampi, J. Lalouette, R. Schoenig, J-M. Thiriet and N. Villaume.

Jean-François AUBRY
Nicolae BRINZEI
Mohammed-Habib MAZOUNI
December 2015

PART 1

Short Review of Petri Net Modeling

Introduction to Part 1

Petri nets (denoted as PN in this book) were introduced by Carl Adam Petri in 1962 [PET 62]. As finite state automata (FSA) described in Volume 1 of this book series [AUB 15], PNs are intended to describe discrete event systems but contrary to FSAs, the transition function is explicitly described in PNs. Adding the suggestive and intuitive graphic representation, we can say that PN is a more powerful model than FSA to describe discrete event systems, due to the fact that an FSA may always be transposed into PN whereas PNs, for example, do not always have a finite state number. We will show here that the notion of language, set of all the possible event sequences in a system, may also be associated with a PN and that the class of these languages is wider than regular languages associated with FSAs.

Like for FSAs, PNs were the subject of multiple extensions at first to move them from the abstraction level, where only event sequencing is considered, to the level taking time into account. Timed PNs were defined to describe behavior of deterministic time systems. Following extensions, called non-autonomous PNs, associated with a PN, an external environment is needed in order to consider synchronization events, continuous variables, especially to describe controlled systems. All these models at various levels have an interest to model problems in the dependability assessment of systems.

Autonomous Petri Nets

1.1. Unmarked Petri nets

1.1.1. Definitions

A unmarked PN is a bipartite oriented 1-graph¹ provided with a mapping ϖ from the set of arcs to the positive integer set \mathbb{N}^+ :

$$Q = \langle P, T, A, \alpha, \beta, \varpi \rangle$$

– P and T are two disjoint subsets of nodes: $P \cap T = \emptyset$:

- P is the Place subset with a finite cardinal p ;

- T is the Transition subset with a finite cardinal t .

– A is the set of Arcs, α and β are the mappings associating with each arc, its origin and its goal nodes, respectively, so that:

$$\forall a \in A, \quad \text{if } \alpha(a) \in T \text{ then } \beta(a) \in P$$

$$\text{if } \alpha(a) \in P \text{ then } \beta(a) \in T$$

– ϖ is a mapping or weighting function associating an integer with each arc, $\varpi : A \rightarrow \mathbb{N}^+$.

If \mathbb{N} is reduced to $\{1\}$, the PN is of ordinary type (or state transition graph), otherwise the PN is of generalized type.

¹ An oriented 1-graph is such that only one arc may be associated with a couple {origin node and destination node}.

Practically, rather than this formalism directly issued from the graph theory, we will use a definition where A does not explicitly appear. As it is an 1-graph, it consists of considering all the couples (P_i, T_j) or (T_i, P_j) and two applications w^- and w^+ .

The PN is then defined as:

DEFINITION 1.1.– An unmarked PN or Place/Transition (P/T) net is a 4-uple $Q = \langle P, T, w^-, w^+ \rangle$ [DAV 89] where:

- P is the set of places (finite cardinal p);
- T is the set of transitions (finite cardinal t);
- $w^-(P_i, T_j) : P \times T \rightarrow \mathbb{N}$ is the backward transition function;
- $w^+(P_i, T_j) : T \times P \rightarrow \mathbb{N}$ is the forward transition function.

The value “0” associated with the couple (P_i, T_j) by w^- or w^+ means that there is no arc between P_i and T_j or T_j and P_i . If the value $k \in \mathbb{N}^+$ is associated with this couple by w^- , respectively w^+ , then one arc oriented from P_i to T_j , respectively from T_j to P_i , exists between these nodes with the valuation k .

REMARK 1.1.– Another possibility is to define a PN as an n -graph (n arcs may exist between two nodes), an arc of weight n being replaced by n arcs each of them having the weight one.

1.1.2. Drawing

In the drawing of a PN, places and transitions are, respectively, represented by circles and streaks (or filled or empty rectangles) and the arcs are arrows to which the weights are attached. Figure 1.1 shows an example of PN with three places and two transitions respectively named as P_1, P_2, P_3, T_1, T_2 . From this figure, we can write $w^-(P_1, T_1) = 1$, $w^-(P_3, T_2) = 2$, $w^-(P_2, T_1) = 2$, $w^+(P_3, T_1) = 2$, $w^+(P_1, T_1) = 1$, $w^+(P_2, T_2) = 3$.

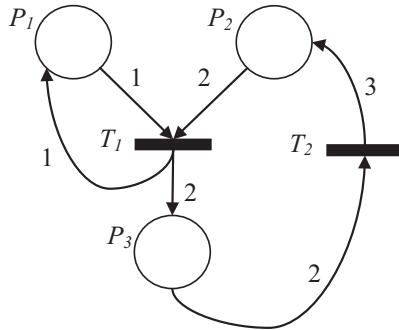


Figure 1.1. The drawing of a PN

1.1.3. Other definitions

Some of other definitions concerning particular cases of PN are summarized in the Appendix, section A.1.

1.2. Marking of a PN

Notations [DAV 89]:

– $I(T_j) = \{P_i \in P | w^-(P_i, T_j) > 0\}$ is the set of the input places of T_j ;

– $O(T_j) = \{P_i \in P | w^+(P_i, T_j) > 0\}$ is the set of the output places of T_j ;

– $I(P_i) = \{T_j \in T | w^+(P_i, T_j) > 0\}$ is the set of the input transitions of P_i ;

– $O(P_i) = \{T_j \in T | w^-(P_i, T_j) > 0\}$ is the set of the output transitions of P_i .

For example, in Figure 1.1, $I(T_1) = \{P_1, P_2\}$ and $O(T_1) = \{P_3, P_1\}$.

The marking is a notion resulting from the association of tokens with the places of the PN. The position in the places of these tokens will evolve to represent the dynamics of the described system. This evolution is performed according to a set of rules described in section 1.3.

DEFINITION 1.2.– A marked PN is a couple $R = \langle Q, M_0 \rangle$ where Q is an unmarked PN and M_0 is an initial marking.

The marking M of a PN at a given instant is a p -sized columnar vector of integers (p is the place number of the PN), each of its component being the marking (or charge) of the place P_i that is to say the number of tokens inside P_i at the considered time instant:

$$M \in \mathbb{N}^p \quad M^T = [M(P_1), M(P_2), \dots, M(P_p)]$$

The initial marking M_0 is the marking at time $t = 0$.

Figure 1.2 shows the initial marking of the PN of Figure 1.1 with: $M_0^T = [1, 2, 0]$.

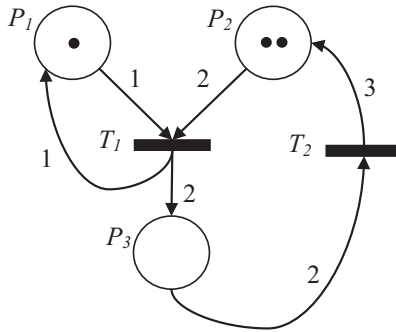


Figure 1.2. A marked PN

1.2.1. Order relation on markings

Let us consider two markings M_1 and M_2 of a PN.

We define the order relation between these markings as follows:

- $M_1 \geq M_2 \iff M_1(P_i) \geq M_2(P_i), \forall P_i \in P$;
- $M_1 > M_2 \iff M_1(P_i) \geq M_2(P_i), \forall P_i \in P$ and $\exists P_i | M_1(P_i) > M_2(P_i)$.

1.2.2. Enabled transition

The transition T_j is enabled for a given marking M if and only if:

$$M(P_i) \geq w^-(P_i, T_j), \forall P_i \in I(T_j)$$

In Figure 1.1, only the transition T_1 is enabled.

1.3. Dynamics of autonomous PNs

The previously defined notion of marking is the observation means of the evolution of the model. The position of the tokens will evolve according to a set of formal rules allowing the definition of some properties of the model. This will be recalled in the following, and more details may be found, for example, in [CAS 08, DAV 92, BES 01].

1.3.1. Firing of a transition

As PNs are models dedicated to discrete events systems, the firing of a transition may be considered as an event describing an elementary evolution of a system (see section 2.1 for the formal definition of labeled PN) characterized by the successive values of the marking before and after the firing. An enabled transition may be fired; from a given marking, each enabled transition could be fired but only one will be. The choice of the transition to be fired can be done arbitrarily. When a place has two output transitions their firings are in conflict. This notion of conflict (formally defined in Appendix A1.1) will be retrieved, for example, each time a failure occurs concurrently with a task activation or achievement. Some PN-dedicated software tools give the possibility of priority assignment to a transition concerned by a conflict, but this must be carefully handled to avoid the appearance of dead branches in the reachability graph. Two transitions $T_1, T_2 \in O(P_i)$ are not in conflict if they are not simultaneously enabled, which implies that these transitions have input places other than P_i .

The set of the enabled transitions must always be considered according to the current marking of the PN and not limited to a given place.

If M_b is the marking before the firing of T_j , the marking M_a after the firing is defined by:

$$\begin{aligned} -\forall P_i \notin I(T_j) \cup O(T_j) &\implies M_a(P_i) = M_b(P_i) \\ -\forall P_i \in I(T_j) - (I(T_j) \cap O(T_j)) &\implies M_a(P_i) = M_b(P_i) - w^-(P_i, T_j) \\ -\forall P_i \in O(T_j) - (I(T_j) \cap O(T_j)) &\implies M_a(P_i) = M_b(P_i) + w^+(P_i, T_j) \\ -\forall P_i \in I(T_j) \cap O(T_j) &\implies M_a(P_i) = M_b(P_i) - w^-(P_i, T_j) + w^+(P_i, T_j) \end{aligned}$$

The firing of the transition T_j subtracts in place P_i as many tokens as indicated by $w^-(P_i, T_j)$ and adds in place P_k as many tokens as indicated by $w^+(P_k, T_j)$.

Figure 1.3 shows the PN of Figure 1.2 after the firing of transition T_2 .

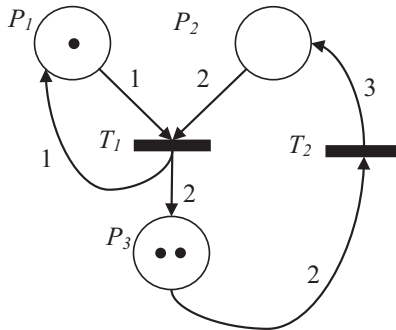


Figure 1.3. PN of Figure 1.2 after firing of transition T_2

1.3.2. Transition matrix

DEFINITION 1.3.– Let us define the backward matrix and forward matrix, as the following matrices with p lines and t columns:

$$W^- = \begin{bmatrix} w^-(P_1, T_1) & \cdot & \cdot & \cdot & w^-(P_1, T_t) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & w^-(P_i, T_j) & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad [1.1]$$

$$W^+ = \begin{bmatrix} w^+(P_1, T_1) & \cdot & \cdot & \cdot & w^+(P_1, T_t) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & w^+(P_i, T_j) & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ w^+(P_p, T_1) & \cdot & \cdot & \cdot & w^+(P_p, T_t) \end{bmatrix} \quad [1.2]$$

The transition matrix W is defined by:

$$W = W^+ - W^- \quad [1.3]$$

The transition matrix (p lines and t columns) is independent of the marking, each column simply shows the number of tokens to remove or add in a place when the corresponding transition fires. For Figure 1.1:

$$W = \begin{bmatrix} 1 & 0 \\ 0 & 3 \\ 2 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ -2 & 3 \\ 2 & -2 \end{bmatrix}$$

1.3.3. Firing sequence

DEFINITION 1.4.– A firing sequence is obtained when a set of transitions are successively fired, starting from an initial marking. It is represented by the concatenation of the successive names of the fired transitions.

If for example starting from the initial marking M_0 the transitions T_1 then T_2 are fired to give the marking M_2 , the sequence will be denoted as:

$$S = T_1T_2, \quad M_0 \xrightarrow{S} M_2$$

REMARK 1.2.– The transition set T provided with the concatenation operation and a neutral element may be considered as a monoid denoted by T^* . With such a notation, a firing sequence is one element of this monoid: $S \in T^*$. This notation will sometimes be used later.

DEFINITION 1.5.– Let S be a firing sequence feasible from a marking M_i , the characteristic vector of the sequence denoted as N is a t size vector, whose j^{th} component represents how many times the transition T_j is fired in the sequence S :

$$N \in \mathbb{N}^t, \quad N(T_j) = k \text{ if } T_j \text{ is fired } k \text{ times.}$$

1.3.4. Reachable marking

The M vector cannot take any value. From a given marking M_0 , it is possible to list all the possible firing sequences. The obtained marking after each of these sequences is a reachable marking.

Let us note that $R(M_0)$ is the set of the reachable marking from the initial marking M_0 :

$$R(M_0) = \{M_a \in \mathbb{N}^p : \exists S/M_0 \xrightarrow{S} M_a\}$$

1.3.5. Fundamental equation

In FSA, we defined the state changes by the mean of the transition function. In PNs, this function is defined as follows:

$$f : \mathbb{N}^p \times T \rightarrow \mathbb{N}^p$$

$f(M_k, T_j)$ is defined if and only if T_j is enabled, in this case, $f(M_k, T_j) = M_{k+1}$ with:

$$M_{k+1}(P_i) = M_k(P_i) - w^-(P_i, T_j) + w^+(T_j, P_i) \text{ for } P_i \in I(T_j) \cup O(T_j)$$

As for the FSAs, we can extend f from the domain $\mathbb{N}^p \times T$ to the domain $\mathbb{N}^p \times T^*$ (T^* being the monoid on the set T provided with the concatenation operation (see section 1.3.3)) and define for a given initial marking, the new obtained marking after a firing sequence of characteristic vector N .

We then obtain the fundamental matrix equation as:

$$M_k = M_i + W \cdot N \quad [1.4]$$

For Figure 1.2, let us imagine from the initial marking, the firing sequence T_2T_1 . After the firing of T_2 , the obtained marking is shown by Figure 1.3 and after the firing of T_1 it becomes as indicated by Figure 1.4.

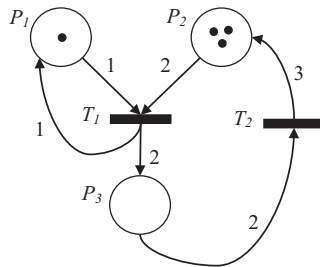


Figure 1.4. PN state of Figure 1.3 after firing of transition T_1

As the two components of the vector N are 1 and 1, each of the two transitions being fired one time, the obtained marking may be retrieved by the following calculus:

$$\begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -2 & 3 \\ 2 & -2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

1.3.6. Properties of PN

A set of definitions and properties are summarized here. For a complete description and formal demonstrations of properties, we can report to [DAV 89, CAS 08, DAV 92, BES 01, BRA 83]:

– *Boundedness*:

- a place of a PN is bounded for a given initial marking M_0 if for any accessible marking from M_0 the token number in this place remains finite. If $\forall M_n \in R(M_0), M_n(P_i) \leq k$ with $k \in \mathbb{N}$, then P_i is k -bounded,

- a PN is bounded for a given initial marking M_0 if all the places are bounded for M_0 .

If $\forall P_i \in P, \forall M_n \in R(M_0), M_n(P_i) \leq k$ with $k \in \mathbb{N}$, then the PN is k -bounded.

These properties are dependent of the initial marking but sometimes a PN may be structurally bounded, that is to say bounded for any initial marking.

– *Liveness*:

- a transition T_j is alive for a given marking M_0 if $\forall M_n \in R(M_0), \exists S : M_0 \xrightarrow{S} M_n / T_j \in S$ (there is always a firing of T_j),

- a PN is alive for a given marking M_0 if all its transitions are alive for M_0 .

– *Blocking*:

- a blocking is a marking from which any transition is enabled. It corresponds to an absorbing state,

- a PN is blocking free for a given initial marking M_0 if no marking $M_n \in R(M_0)$ is a blocking.

Liveness and blocking are properties dependant on the initial marking M_0 .

1.3.7. Other properties

Some other properties are summarized in Appendix A.1.

1.3.8. Invariants in a PN

1.3.8.1. Conservative component and marking invariant

It should be noted that sometimes the weighted sum of the markings of a subset of places remains constant. This is an invariant of this subset which is called conservative component of the PN. As it is independent of the initial marking, this is a property of the unmarked PN (the value of this constant may only depend on the initial marking). In most cases, this is the characteristic of a physical property of the modeled system.

A P -semi-flow is a vector F of integers of dimension p (number of places of the PN) so that:

$$F^T \cdot W = 0 \quad [1.5]$$

According to the fundamental equation $M_k = M_i + W \cdot N$ (for any accessible marking from M_i by a firing sequence S characterized by the vector N): $F^T \cdot M_k = F^T \cdot M_i + F^T \cdot W \cdot N$. If $F^T \cdot W = 0$ we obtain:

$$F^T \cdot M_k = F^T \cdot M_i \quad [1.6]$$

which is the marking invariant.

The integers of the vector F may be considered as weights assigned to the places of the PN. The subset of places whose weights are null is the PN conservative component support of the P -semi-low. It will be noted P_F .

Any linear combination of a semi-flow is itself a semi-flow.

Let $P_F = \{P_1, P_2, \dots, P_r\}$ be a conservative component of a PN and $F = [q_1, q_2, \dots, q_r]^T$ the corresponding weighting vector. All the places of the conservative component are bounded and we get: $M(P_i) \leq F^T \cdot M_0 / q_i$.

For example, it is easy to verify that in the PN of Figure 2.1 (see section 2.2) the subset of places $\{P_4, P_5\}$ is a conservative component, the sum of their marking is always equal to 1 (the initial marking of P_4).