Information Sciences 363 (2016) 198–220

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A survey of siphons in Petri nets

GaiYun Liu [a,b,*], Kamel Barkaoui [b]

[a] School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, PR China
[b] Cedric Lab and Computer Science Department, Conservatoire National des Arts et Métiers, Paris 75141, France

ARTICLE INFO

Article history:
Received 31 December 2014
Revised 28 June 2015
Accepted 21 August 2015
Available online 29 August 2015

Keywords:
Concurrent system
Petri net
Deadlock
Siphon
Supervisor

ABSTRACT

Petri nets have gained increasing usage and acceptance as a basic model of asynchronous concurrent systems since 1962. As a class of structural objects of Petri nets, siphons play a critical role in the analysis and control of systems modeled with Petri nets. This paper surveys the state-of-the-art siphon theory of Petri nets including basic concepts, computation of siphons, controllability conditions, and deadlock control policies based on siphons. Some open problems on siphons are discussed, such as the maximally permissive supervisor design problems based on siphons and the application of siphons to robust supervisory control. This survey is expected to serve as a reference source for the growing number of Petri net researchers and practitioners.

1. Introduction

In 1962, Petri invented a net-theoretic approach to model and analyze communication systems in his thesis [146]. This model was based on the concepts of asynchronous and concurrent operation by the parts of a system and the realization that relationships between the parts could be represented by a net. A large amount of research has been done on both the nature [153] and the application of Petri nets, and their application seems to be expanding [43,51,147]. Petri nets have been proven to be very useful in the modeling, analysis [202,203], simulation, and control of concurrent systems [34,35,66–81,92,94,98,108,111,113,123,124,133,137,150,151,165,183,184,189,196–198,205–213].

The concurrent flow of multiple jobs in a resource allocation system (RAS), which all compete for a finite set of resources, can lead to a deadlock. A deadlock occurs when a set of jobs are in a "circular wait" state [122], where each job in the set is waiting for a resource being held by another job in the set while occupying a resource that is, in turn, needed by one of the other jobs [3]. The notion of partial or total deadlock is frequent and validation before implementation is preferable to reduce the risks.

Due to the easy and concise description of the concurrent execution of processes and the resource sharing by Petri nets, many methods to verify deadlock-freeness and to synthesize controllers using structural theory or reachability graph analysis have been proposed over the past two decades. Traditionally, a deadlock control policy can be evaluated by three performance criteria: structural complexity, behavioral permissiveness, and computational complexity [25–27]. A maximally permissive supervisor has great potential to lead to high utilization of system resources. A supervisor with a simple structure can decrease the hardware and software costs. A policy with low computational complexity means that it can be applied to real-world large-sized systems. Many researchers have developed deadlock control algorithms that can obtain liveness-enforcing supervisors with maximal permissibility, a simple supervisory structure, and low computational complexity. Generally, deadlock control policies based on the state space analysis can approach the maximal permissive behavior, but suffer from the state explosion problem [28–31,140,141].

* Corresponding author. At: School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, PR China. Tel.: +8613572429091.
E-mail addresses: lg_2005@163.com, gaiyunliu@gmail.com (G.Y. Liu), kamel.barkaoui@cnam.fr (K. Barkaoui).

http://dx.doi.org/10.1016/j.ins.2015.08.037
0020-0255/

On the contrary deadlock control policies based on the structural analysis of Petri nets avoid in general the state explosion problem successfully [6,7,11]. Resource circuits and siphons are two typical structural objects which have been widely studied in deadlock control area. This paper will focus on siphons and their applications to the development of deadlock control policies.

The idea behind siphon was first introduced by Commoner and Holt in 1971 for a restricted class of Petri nets called Free-Choice Petri nets [37]. Then Hack in his thesis showed important results for the class of Free-Choice Petri nets and solves the deadlocks (i.e., siphons) and unpredictability problem for a restricted class of systems called Production Schemata [56]. Most of the studies on Petri nets around this time were in the form of theses, dissertations, and reports that are neither readily available nor in wide circulation. From the internet we can find the first paper which gave the name "siphon" to this special structural objects in 1975 in [14]. A nonempty subset of places S in a net N is called a siphon (also known as a deadlock or co-trap) if ${}^{\bullet}S \subseteq S^{\bullet}$, i.e., every transition having an output place in S has an input place in S. We use a siphon instead of a deadlock since the latter may lead to terminological misunderstandings and it is used for a circular waiting condition or behavior in computer science [38,139].

To acquaint the reader with Petri nets, in Section 2, we first present their formal definition and the related concepts, including structural and behavioral properties. The concepts of elementary and dependent siphons are given in this section. A number of important subclasses of Petri nets and their relationships are discussed. In Section 3, different methods of siphon computation are recalled. Section 4 presents controllability conditions of siphons. Section 5 reviews deadlock control policies based on siphons, including the enumeration of strict minimal siphons, mathematical programming techniques, elementary siphons, deadlock control combined with reachability graph analysis and that based on siphons in Gadara nets, and robust deadlock control policies. Their advantages and limitations are discussed. Section 6 shows some open problems. Section 7 concludes this survey.

Because of the breadth of the applications and depth of the investigations into Petri nets, we touch on the results available in the literature. We refer the interested reader to the original works cited in the references for the proofs and details of the related research.

2. Preliminary

2.1. Basics of Petri nets

A generalized Petri net [109,139] is a four-tuple $N = (P, T, F, W)$ where P and T are finite and non-empty. P is the set of places and T is the set of transitions with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(f) > 0$ if $f \in F$ and $W(f) = 0$ otherwise, where $\mathbb{N} = \{0, 1, 2, \dots\}$. If $\forall f \in F, W(f) = 1$, then $N = (P, T, F, W)$ is called an ordinary net and denoted as $N = (P, T, F)$. A Petri net $N = (P, T, F, W)$ is pure (self-loop free) if $\forall x, y \in P \cup T$, $W(x, y) > 0$ implies $W(y, x) = 0$. A pure Petri net $N = (P, T, F, W)$ can be represented by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$.

A marking M of a Petri net N assigns to each place a nonnegative integer. To facilitate linear algebraic analysis, a marking M is usually treated as a $|P|$-vector. $M(p)$ denotes the number of tokens in place p. For economy of space, $\Sigma_{p \in P} M(p)p$ is used to denote vector M. Place p is marked at M if $M(p) > 0$. Given a subset $S \subseteq P$, the sum of tokens in all the places in S is denoted by $M(S)$ with $M(S) = \sum_{p \in S} M(p)$. S is marked (unmarked) at M if $M(S) > 0$ ($M(S) = 0$). (N, M_0) is called a net system and M_0 is called an initial marking of N.

A transition t is enabled (disabled) at M if $\forall p \in {}^{\bullet}t$ ($\exists p \in {}^{\bullet}t$), arc (p, t) is enabled (disabled). This fact can be denoted by $M[t\rangle$. Firing an enabled transition t reaches a new marking M' such that for each place p, $M'(p) = M(p) - W(p, t) + W(t, p)$, denoted by $M[t\rangle M'$. Marking M' is said reachable from M if there exists a finite sequence of transitions $\sigma = t_0 t_1 \dots t_n$ and markings M_1, $M_2, \dots,$ and M_n such that $M[t_0\rangle M_1[t_1\rangle \dots M_n[t_n\rangle M'$ holds. Specially, when σ is an empty sequence, we have $M[\sigma\rangle M$. The set of markings reachable from M_0 in N is called the reachability set of the marked Petri net (N, M_0) and denoted as $R(N, M_0)$. Marking $M \in R(N, M_0)$ is legal if $M_0 \in R(N, M)$.

Given a marked Petri net (N, M_0), a transition $t \in T$ is live at M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $M'[t\rangle$ holds. A transition $t \in T$ is said to be dead at marking $M \in R(N, M_0)$, if $\nexists M' \in R(N, M)$, $M'[t\rangle$. (N, M_0) is live at M_0 if $\forall t \in T$, t is live at M_0. Otherwise, (N, M_0) is non-live. (N, M_0) is deadlocked at M if $\nexists t \in T, M[t\rangle$, where $M \in R(N, M_0)$ and M is called a dead marking. (N, M_0) is deadlock-free if $\forall M \in R(N, M_0)$, $\exists t \in T$, $M[t\rangle$.

A marked net is bounded if $\exists k \in \mathbb{N}$, $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq k$. A net N is structurally bounded if it is bounded for any initial marking.

A P(T)-vector is a column vector $I(J) : P(T) \to \mathbb{Z}$ indexed by P(T), where $\mathbb{Z}$ is the set of integers. P-vector I is called a P-invariant (place invariant [193]) if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. Here $\mathbf{0}$ is a zero vector. A P-vector I is denoted by $\Sigma_{p \in P} I(p)p$ for economy of space. For example, $I = (1, 0, 0, 0, 2)^T$ is written as $I = p_1 + 2p_5$. P-invariant I is a P-semiflow if every element of I is non-negative. $||I|| = \{p \in P | I(p) \neq 0\}$ is called the support of I. $||I||^+ = \{p \in P | I(p) > 0\}$ denotes the positive support of P-invariant I and $||I||^- = \{p \in P | I(p) < 0\}$ denotes the negative one. I is called a minimal P-invariant if $||I||$ is not a proper superset of the support of any other and the greatest common divisor of its elements is one. If I is a P-invariant of (N, M_0), then $\forall M \in R(N, M_0)$, $I^T M = I^T M_0$.

Let $x \in P \cup T$ be a node of net N. ${}^{\bullet}x = \{y \in P \cup T \mid (y, x) \in F\}$ is called the preset of x, while $x^{\bullet} = \{y \in P \cup T \mid (x, y) \in F\}$ is called its postset. This notation can be extended to a set of nodes as follows: given $S \subseteq P \cup T$, ${}^{\bullet}S = \cup_{x \in S} {}^{\bullet}x$ and $S^{\bullet} = \cup_{x \in S} x^{\bullet}$. Let S be a non-empty subset of places. $S \subseteq P$ is a siphon (trap) if ${}^{\bullet}S \subseteq S^{\bullet}$ ($S^{\bullet} \subseteq {}^{\bullet}S$) holds.