International Journal of Mathematics Trends and Technology
ISSN: 2231-5373/ https://doi.org/10.14445/22315373/IJMTT-V72I10P110

Volume 72 Issue 10, 116-124, October 2026
© 2026 Seventh Sense Research Group®

Original Article

Algebraic Structures in Petri Net Theory: Monoid, Categorical and Max-plus Approaches Reviewed, with Computationally Checked Examples

Saithalavi P[1], Girish K. P[2]

[1]*Department of Mathematics, MES Keveeyam College, Valanchery, University of Calicut, Kerala, India.*
[2]*Department of Mathematics, MES Kalladi College(Autonomous), Mannarkkad, University of Calicut, Kerala, India.*

[2]*Corresponding Author : girikalam@gmail.com*

Received: 29 August 2026 Revised: 02 October 2026 Accepted: 19 October 2026 Published: 29 October 2026

Abstract *- Petri nets model concurrent systems, and their theory combines graph-theoretic, linear-algebraic, semigroup-theoretic, categorical and max-plus methods whose results are often quoted in simplified form. This paper examines three such statements and places them in a critical comparison of the algebraic strands of the theory. First, it proves that the minimal-support invariants of a net generate all invariants over the non-negative rationals but not over the integers, and it determines the Hilbert basis of an explicit three-place net. Second, it proves that the circuit vectors of a marked graph are invariants, so that differences of reachable markings are orthogonal to them and not combinations of them, which corrects a common misstatement of the reachability criterion. Third, it replaces the claim that reachability sets are semilinear by the precise statement that this holds only in low dimension. A validation study with two independent implementations tests the classical circuit criteria for marked graphs on 89 837 nets of an exhaustive class and on 1 000 random nets, and tests the max-plus cycle-time statement on 300 random timed event graphs with an exact rational test; it supports the published theorems on the nets tested without proving them. The comparison identifies, for each strand, its established results, its limits and its open problems. The clarifications give a more precise basis for the algebraic treatment of nets than the formulations in common use.*

Keywords *- Automorphism Groups, Concurrency Theory, Invariants, Prefix Codes, Reachability.*

1. Introduction

A Petri net is at once a picture and a piece of algebra. As a picture, it is a bipartite directed graph of places and transitions joined by weighted arcs, on which tokens move according to one firing rule. As algebra, it is a vector addition system: firing a transition adds an integer vector to the current marking, subject to non-negativity, so that the markings reachable from an initial marking lie in that marking plus the monoid generated by the transition vectors. Petri's dissertation already contained both aspects [1], and the tension between the operational and the structural view has shaped the theory since [2]. Standard texts are [3–6], and the classical linear-algebraic techniques are surveyed in [7, 8].

Four algebraic strands can be distinguished. The classical strand analyses nets through the incidence matrix and its invariants [2]. A second strand, developed in Kunimochi's dissertation [9], treats net languages, prefix codes and automorphism groups with semigroup theory. A third, beginning with Meseguer and Montanari [10] and Winskel [11], treats nets as monoids and leads to categories of open nets [12]. A fourth treats timed nets over the max-plus semiring [13–15].

These strands are usually presented separately, and each is often quoted in simplified form. The research gap has three parts. No single account compares the strands with respect to what they establish, where they stop and what remains open. Several statements that circulate in the secondary literature are imprecise: the generation of invariants by minimal-support vectors, the characterisation of reachability in marked graphs, and the description of reachability sets as semilinear. Moreover, the finite statements on which the classical theory relies are rarely checked independently, and the checks that exist are not documented well enough to be repeated.

The paper addresses these three parts. Its original contributions are limited and are of two kinds that are kept apart. The first kind consists of mathematical clarifications that are proved in the text, namely Propositions 1–3 and Remark 1. The

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

second is a validation study that provides evidence for published theorems on finite classes of nets and does not prove them. A critical comparison of the four strands (Table 1) supplies the setting and is based on published results.

Section 2 describes the literature base, the definitions and the design of the validation. Section 3 presents the results, the comparison of the strands and a discussion with open problems, and Section 4 concludes.

2. Materials and Methods

2.1. Literature Base and Comparison Method

The literature base was assembled narratively and not by a systematic database search, and it is anchored on two documents: Murata's survey [2], the standard reference for the classical behavioural and structural theory, and Kunimochi's dissertation [9], which develops an algebraic apparatus of semigroups with length, free monoids, automorphism groups of net structures and prefix codes. Because these two sources are old and single, the literature base was extended to recent work in three steps. Recent categorical, max-plus and compositional papers (2018–2026) were located by following citations from the anchor documents and from the categorical literature, and by web searches in October 2026; the bibliographic details of every recent item were checked against publisher or repository records, and several errors in earlier citations of these items were corrected. Foundational results on reachability and on invariants were added from the standard literature [8, 16–19]. A work was included if it concerns the algebraic structure of Petri nets or the algebraic computation of their properties and its details could be confirmed. The synthesis classifies each strand by its main objects, its established results, its limitations and its open problems (Table 1). The literature base can miss work outside these citation networks, and no database search with logged queries was performed.

Table 1. Critical comparison of the algebraic strands of Petri net theory

Strand	Established	Limits	Open
Linear algebra [2, 8]	invariants, state equation, structural properties, marked-graph theorems	conditions necessary but not sufficient; rational versus integer generation	structure of invariant monoids by subclass
Semigroups, languages, codes [9]	twelve language classes, prefix codes, every finite group is some $\mathrm{Aut}(N)$	labelled nets only; few citations; inclusion questions open	semigroup analogues; inclusion of code classes
Categorical [10–12, 20–22]	free monoidal categories, open nets, compositional semantics, classification of additive invariants	liveness is not compositional; tooling immature	timed and stochastic open nets
Max-plus and stochastic [13–15, 23, 24]	linear recurrences, cycle time, transience bounds	event graphs only; choice needs other tools	nets with choice
Reachability [16–18, 25, 26]	decidability, Ackermann-completeness	non-semilinear sets; very high complexity	tractable subclasses

2.2. Definitions, Firing Rule and State Equation

The definitions follow Murata [2].

Definition 1. A *Petri net* is a 5-tuple $PN = (P, T, F, W, M_0)$ where $P = \{p_1, \dots, p_m\}$ is a finite set of places, $T = \{t_1, \dots, t_n\}$ a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ the flow relation, $W: F \to \{1,2,3, \dots\}$ the weight function and $M_0: P \to \{0,1,2, \dots\}$ the initial marking, with $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

A transition t is *enabled* at a marking M if each input place p of t holds at least $W(p,t)$ tokens; firing removes $W(p,t)$ tokens from each input place and adds $W(t,p)$ tokens to each output place. Iterating the rule from M_0 gives the reachability set $R(M_0)$. The incidence matrix is $A = [a_{ij}]$ with $a_{ij} = a_{ij}^{+} - a_{ij}^{-}$ the net change in place j caused by one firing of transition i (rows are transitions). With x the firing count vector of a sequence leading from M_0 to M_d,

$$A^{\mathsf{T}} x = M_d - M_0. \tag{1}$$

Solvability of Equation 1 in non-negative integers is necessary but not sufficient for reachability (Example 1).

2.3. Design of the Computational Validation

Status. The computations below test published theorems on finite classes of nets. A computation without exception supports a theorem but does not prove it, and no result of this section is used as a premise of a proof in the paper. Their purpose is to exclude transcription or reading errors and to show where a loosely quoted statement fails.

Random generation and sampling criteria. Marked graphs: the number of transitions n is chosen uniformly from $\{3,4\}$; the net contains the ring of arcs $t_i \to t_{i+1}$ (so it is strongly connected) and one to three further arcs with uniformly chosen endpoints (parallel arcs and self-loops allowed); the number of tokens of each place is drawn uniformly from the list $(0,0,1,1,2)$; a net is rejected if the total number of tokens exceeds five. Of 1 385 attempts, 385 were rejected for the token cap, and 1 000 nets were retained (598 live, 402 not live). Timed event graphs: $n \in \{3,4,5\}$, the ring plus one to four extra arcs, the number of tokens of each place drawn uniformly from the list $(0,1,1,2)$integer delays uniform on $1, \ldots, 9$; a net is rejected if some circuit carries no token. Of 365 attempts, 65 were rejected and 300 retained. The pseudo-random generator is the Python random module with seeds 20261014 (marked graphs) and 20261015 (timed graphs).

Exhaustive class. In addition to the random samples, all strongly connected marked-graph structures with two or three transitions and at most six places, and with four transitions and four places (arc multisets, parallel arcs and self-loops allowed), were enumerated: 623 structures, with every marking of entries at most two and at most four tokens, giving 89 837 nets (21 057 live, 68 780 not live).

Environment and independent validation. Python 3.12.3, NetworkX 3.6.1 and NumPy 2.4.4 on Linux x86-64; the validation of Table 2 takes about 30 seconds on one core. For marked graphs, the truth is established by exhaustive reachability graphs, computed by two independent implementations of the firing rule (arc lists with breadth-first search, and pre- and post-incidence matrices with depth-first search); liveness is decided by two independent tests (forward closure of the reachability graph, and the criterion that every bottom strongly connected component of the reachability graph contains an edge labelled by every transition). For timed event graphs, three independent computations are compared: simulation of the recurrence for 5 000 firings, the maximum circuit ratio from enumeration of all elementary circuits, and the exact parametric test of Section 3.6 with rational arithmetic and tolerance 10^{-6}.

3. Results and Discussion

3.1. Reachability Sets, Decidability and Classical Methods

Remark 1. A Petri net with m places is a vector addition system of dimension m, and the statement that its reachability set is semilinear holds only in low dimension. Hopcroft and Pansiot proved that reachability sets of vector addition systems of dimension at most five are effectively computable semilinear sets and gave an example of a vector addition system of dimension six whose reachability set is not semilinear; for vector addition systems with states, reachability sets are semilinear in dimension two, and a three-dimensional counterexample exists [16]. In general, therefore, the markings reachable from M_0 lie in M_0 plus the monoid generated by the transition vectors, but they do not form a semilinear set, and the decidability of reachability [17, 18] cannot be obtained by describing the reachability set in closed form.

Decidability and complexity. Reachability is decidable [17, 18], requires at least exponential space [27], is not elementary [28], is not primitive recursive [25] and is Ackermann-complete [26]; surveys of decidability issues are given in [19]. Boundedness and coverability are settled by the coverability tree of Karp and Miller [29], which collapses unboundedly growing components to a symbol ω; it decides boundedness and safeness but does not by itself settle liveness or general reachability for unbounded nets, because ω discards arithmetic information [2].

Reduction rules. Six elementary transformations (fusion of series and parallel places or transitions, elimination of self-loop places and transitions) preserve liveness, safeness and boundedness [2]. They can be read as identifying behaviourally equivalent nets, which anticipates the categorical treatments of Section 3.5.

3.2. Invariants and their Generation

Invariants. For a net with incidence matrix A, an *S-invariant* is a non-negative integer vector y with $Ay = 0$; then. $M^{\mathsf{T}}y$ is constant on $R(M_0)$. A *T-invariant* is a non-negative integer vector x with $A^{\mathsf{T}}x = 0$. An element of $\{y \in \mathbb{N}^m : Ay = 0\}$ has *minimal support* if it is non-zero and no non-zero element has a strictly smaller support. The set of S-invariants is a finitely generated commutative monoid, because it is the set of non-negative integer solutions of a homogeneous linear system and has a finite Hilbert basis by Gordan's lemma [30]. The relation between minimal-support invariants and generation is as follows [8].

Proposition 1. Let $\mathcal{C} = \{y \in \mathbb{Q}_{\geq 0}^m : Ay = 0\}$. Every element of $\mathcal{C}$ is a non-negative rational linear combination of minimal-support elements of $\mathcal{C}$.

Proof. Induction on the size of the support of $y \in \mathcal{C}$, $y \neq 0$. Among the non-zero elements of $\mathcal{C}$ whose support is contained in $\mathrm{supp}(y)$ choose z with minimal support; it is a minimal-support element of $\mathcal{C}$. Put $t = \min\{y_i/z_i : i \in \mathrm{supp}(z)\} > 0$, a

rational number. Then $y - tz \in \mathcal{C}$ has a support strictly smaller than $\mathrm{supp}(y)$, because the minimum is attained at some index. By induction, $y - tz$ is a non-negative rational combination of minimal-support elements, hence so is $y = tz + (y - tz)$. ▫ □

Proposition 1 holds with rational coefficients only; the assumptions are that $\mathcal{C}$ is the rational cone and that the coefficients may be non-integral. The next proposition shows that integrality cannot be added.

Proposition 2. Let $A = (-1, -1, 2)$ be the incidence matrix of the net N_1 of Figure 1(a) and let $\mathcal{M} = \{y \in \mathbb{N}^3 : y_1 + y_2 = 2y_3\}$ be its monoid of S-invariants. Then (i) the minimal-support elements of $\mathcal{M}$ are, up to scaling, $(2,0,1)$ and $(0,2,1)$; (ii) the Hilbert basis of $\mathcal{M}$ is $\{(2,0,1), (0,2,1), (1,1,1)\}$; (iii) $(1,1,1)$ is not a non-negative integer combination of $(2,0,1)$ and $(0,2,1)$, so the minimal-support invariants do not generate $\mathcal{M}$.

Proof. (i) A solution with $y_3 = 0$ has $y_1 = y_2 = 0$, so every non-zero solution has $y_3 > 0$. A solution with support $\{1,3\}$ has $y_2 = 0$ and $y_1 = 2y_3$, hence is a multiple of $(2,0,1)$; similarly support $\{2,3\}$ gives multiples of $(0,2,1)$. A solution with full support contains these supports and is not of minimal support. (iii) A non-negative integer combination $a(2,0,1) + b(0,2,1) = (2a, 2b, a + b)$ has an even first coordinate, whereas the first coordinate of $(1,1,1)$ is odd. (ii) Let $y \in \mathcal{M}$ be non-zero; $y_1 + y_2$ is even, so y_1, y_2 have the same parity. If both are even, $y = \frac{y_1}{2}(2,0,1) + \frac{y_2}{2}(0,2,1)$ because $y_3 = (y_1 + y_2)/2$. If both are odd then $y_3 \geq 1$ and $y - (1,1,1) \in \mathcal{M}$; repeating the step reduces y to the even case after finitely many steps. So $\mathcal{M}$ is generated by the three vectors. None of them is a sum of two non-zero elements of $\mathcal{M}$, because every non-zero element has $y_3 \geq 1$, whereas each of the three vectors has $y_3 = 1$; hence they form the Hilbert basis. ▫ □

The invariant $(1,1,1)$ expresses that the total number of tokens is conserved: from $M_0 = (3,3,0)$ the reachable markings are $(3,3,0), (2,2,2), (1,1,4), (0,0,6)$.

Example 1. Let N_2 of Figure 1(b) have places p_1, p_2 and transitions $t_1 : p_2 \to p_1$ and $t_2 : p_1 \to 2p_2$, with $M_0 = (0,0)$. The firing vector $x = (1,1)$ satisfies Equation 1 with $M_d = (0,1)$, but neither transition is enabled at M_0, so $R(M_0) = \{M_0\}$. The cycle $t_1 t_2$ has a positive net effect but needs a token to start. A search over pure nets with at most three places and at most three transitions, arc weights at most two and initial markings with entries at most two finds this net first; it shows that the state equation cannot replace the analysis of the order of firings.

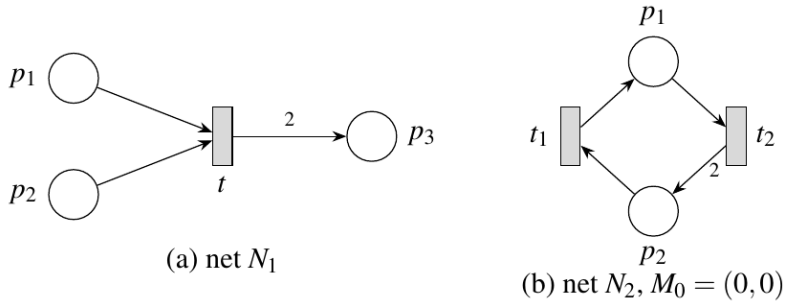

Fig. 1 The nets of Proposition 2 and Example 1; arc weights are 1 unless marked

3.3. Marked Graphs

Restricting the local structure of a net yields the hierarchy of state machines, marked graphs, free-choice nets, extended free-choice nets and asymmetric-choice nets, with state machines and marked graphs inside the free-choice nets, then extended free-choice, asymmetric-choice and, finally, ordinary nets [2, 6]. Marked graphs (every place has one input and one output transition) have the cleanest theory, going back to Commoner, Holt, Even and Pnueli [31]. For a strongly connected marked graph with initial marking M_0 the following are established there.

1) M_0 is live if and only if every directed circuit carries at least one token.
2) A *live* marking is safe if and only if every place lies on a directed circuit carrying exactly one token.
3) From a live marking M_0, a marking M is reachable if and only if M and M_0 carry the same number of tokens on every directed circuit.

Statement 3 is sometimes paraphrased as saying that the difference of two markings is a linear combination of fundamental circuits. That formulation is incorrect, and the correct relation is the following elementary fact, which also gives the necessity direction of statement 3.

Proposition 3. Let N be a marked graph, C a directed circuit, and $y_C \in \{0,1\}^m$ the indicator vector of the places on C. Then $Ay_C = 0$, so y_C is S-invariant, and every difference $M - M_0$ of reachable markings satisfies $y_C^{\mathrm{T}}(M - M_0) = 0$. The difference lies in the image of A^{T} Moreover, is orthogonal to the circuit vectors.

Proof. Let t be a transition on C. Exactly one input place and one output place of t lie on C (consecutive places of the circuit), so the row of A for t has -1 at the first and $+1$ at the second on the places of C (the entries cancel if the two coincide). If t is not on C, no place of C is adjacent to t, because each place has a unique input and a unique output transition, and for places of C these lie on C. Hence every row of A has zero sum over the places of C, that is $Ay_C = 0$. By Equation 1, $M - M_0 = A^{\mathrm{T}}x$ and $y_C^{\mathrm{T}}A^{\mathrm{T}}x = (Ay_C)^{\mathrm{T}}x = 0$. ▫ □

The sufficiency direction of statement 3 (that equal circuit counts imply reachability from a live marking) is the deep part of the theorem of Commoner et al. and is cited, not reproved. Structural properties other than these (structural liveness, boundedness, conservativeness, repetitiveness, consistency, controllability) are characterised by alternatives of Minkowski–Farkas or Stiemke type [2]; the net-theoretic content lies in the construction of A, and the proof content is algebraic.

3.4. Semigroups, Monoids, Codes and Automorphism Groups

Kunimochi's dissertation [9] axiomatises net languages and codes algebraically, and it is summarised here because it shows what an algebraic foundation looks like. The statements below are reported from [9] and were not rechecked.

Free monoids and languages. For an alphabet X (a labelling of transitions), the words over X form the free monoid. X^*. A labelled net induces a language $L(N, M_0) \subseteq X^*$. Twelve classes of Petri net languages are distinguished by four final-marking conditions and three labelling conventions; every Petri net language is context-sensitive, while some, such as $\{a^n b^n c^n \mid n \geq 0\}$, are not context-free [2].

Codes. A subset $L \subseteq X^*$ is a code if it freely generates L^*, and a prefix code if no word is a proper prefix of another. Kunimochi studies the C-type codes defined from nets and reports that such a code is a maximal prefix code (or empty) if and only if every transition is enabled at every marking reachable by a positive firing sequence, a condition equivalent for input-ordinary nets to absence of deadlock [9]. A question about maximality of a code thus becomes a behavioural property. The dissertation further reports that maximality is not preserved under concatenation or a "parallel" operation but is preserved under an "interruption" operation, and that the inclusion between the classes of codes of general and of input-ordinary nets, open in general, holds for nets with at most two places or one transition.

Automorphism groups. An isomorphism of a net structure $N = (P, T, W)$ is a pair of bijections of places and of transitions that preserves arc weights; the automorphisms form a group $\mathrm{Aut}(N)$. Using transformation nets built from transformation semigroups and the right regular representation, it is shown in [9] that every finite group is the automorphism group of some net structure.

Critical remarks. The strengths of this line are that maximality of a code becomes equivalent to deadlock-freedom and that every finite group is realised as a net symmetry group. Its limits are that the results concern labelled nets and language semantics, that the dissertation is a single source that has been little cited next to the graph-theoretic mainstream, and that its inclusion question for code classes is open.

3.5. Categorical and Compositional Algebra

Nets as monoids. Meseguer and Montanari showed that Petri nets are monoids: the commutative monoid of markings together with the pre- and post-condition maps presents a free symmetric strict monoidal category whose morphisms are firing sequences modulo the interchange law of independent transitions [10]; Winskel related nets, algebras and morphisms compositionally [11], and Sassone studied the category of net computations [20].

Open nets. Baez and Master (2020) [12] extend this to open Petri nets, with input and output interfaces given by a cospan of finite sets. Composition glues outputs to inputs by a pushout of place sets; isomorphism classes of open nets are the morphisms of a symmetric monoidal category, and strict composition requires a symmetric monoidal double category. Two double functors give semantics, an operational one assigning a category of processes and a reachability one assigning a relation between input and output markings. A general account of nets as free constructions is given in [21]. Bumpus et al. (2025) [22] classify the additive invariants of open nets, that is, $\mathbb{N}$-valued invariants additive under sequential and parallel composition, which are determined by their values on single-transition nets together with, for monically open nets, transitionless nets.

Aduddell et al. apply compositional methods to biochemical regulatory networks [32], and Alahmadi et al. extend structured occurrence nets, which represent causality and concurrency of evolving systems, to acyclic nets [33].

Critical remarks. The categorical view gives a principled account of modularity: a property preserved by the chosen composition can be verified on the components and inherited by the composite. The qualification is essential, because liveness and reachability are not in general preserved by composition. The programme is also young and its tooling immature, and it concentrates on untimed place/transition nets; the interaction of open-net semantics with the timed and stochastic semantics of Section 3.6 is unresolved.

3.6. Semiring and Max-Plus Structures

For timed marked graphs (event graphs), the firing times obey linear recurrences over the max-plus semiring. $(\mathbb{R} \cup \{-\infty\}, \max, +)$ [13–15]. Let each transition t_i have a delay d_i, and let the place from t_i to t_j carry m initial tokens; the earliest firing times satisfy $x_j(k) = \max_{(i,m)}(x_i(k-m) + d_i)$ with $x_i(k) = 0$ for $k \leq 0$. If every circuit carries a token, the growth rate of $x_j(k)/k$ is the *cycle time*, the largest ratio $D(C)/M(C)$ of total delay to total token count over the directed circuits C, and it equals the max-plus eigenvalue of the associated matrix [14]. This is the algebraic counterpart of the minimum cycle time bound for marked graphs given by Murata through minimal-support S-invariants [2]. The maximum circuit ratio can also be characterised without enumerating circuits: λ^* is the least λ such that no circuit has $\sum(d - \lambda m) > 0$, which is a longest-path (Bellman–Ford) test [34].

Recent work. Allamigeon, Boyet and Gaubert study timed Petri nets with a hierarchical structure, in the continuous (fluid) approximation, for emergency call centres, and show that a counter function coincides with the value function of a semi-Markov decision problem and obtain an explicit bound on the transience time [23]. Abdul Hameed, Lassoued and Schwung relate the max-plus representation of a timed event graph to a one-layer perceptron so that its timing parameters can be learned from data [24].

Stochastic nets. If transition delays are exponentially distributed, the reachability graph of a bounded stochastic net is isomorphic to a finite continuous-time Markov chain, and steady-state measures solve $\Pi Q = 0$, $\sum_i \pi_i = 1$ for the rate matrix Q [2].

Critical remarks. The max-plus description is exact for event graphs but not for nets with choice, where the earliest-firing recurrence is not linear over the semiring; the recent applications [23, 24] work with approximations (fluid models, learned parameters), so their guarantees are those of the approximation.

3.7. Results of the Computational Validation

Table 2 summarises the results. In every marked graph of the exhaustive class (89 837 nets, 21 057 of them live) and of the random sample (1 000 nets, 598 live), the three circuit criteria held without exception, the two implementations of the firing rule produced the same reachability sets, and the two liveness tests agreed. For the 300 timed event graphs, the exact parametric test confirmed the maximum circuit ratio in every case, and the simulated growth rate differed from it by at most 0.0043. These outcomes support the published theorems on the nets tested; they do not prove them, and the limits are listed in the last column of Table 2.

Table 2. Computational validation: tests, independent contribution and limits (programs, nets and outputs: Online Resource 1)

Statement	Test and outcome	Independent contribution	Limit
Minimal-support invariants generate all invariants.	Hilbert basis of N_1 computed by enumeration up to 8 and proved (Prop. 2)	shows the integer failure	one net; proof is the evidence
State equation sufficient for reachability	search over pure nets with at most 3 places and 3 transitions; net N_2 found	explicit counterexample	small search space
Marked graphs: liveness iff every circuit marked [31]	89 837 exhaustive nets and 1 000 random nets: no exception, no implementation disagreement	confirms the statement as quoted	at most 4 transitions, 6 places

Live marked graph safe iff every place on a one-token circuit [31]	21 057 live nets exhaustive and 598 random: no exception	shows that liveness is a necessary hypothesis	as above
Reachability iff equal circuit counts [31]	reachable set equals predicted set in all live nets	confirms the criterion as corrected	as above
Cycle time = maximum circuit ratio [14]	300 random nets: exact test passed in all; simulation deviation at most 0.0043	three-way agreement	delays and tokens small; event graphs only

3.8. Discussion and Open Problems

The case for algebraic structures. The comparison supports a measured case for treating explicit algebraic structures as a research programme. They unify criteria that are otherwise proved separately: maximality of a net code is equivalent to deadlock-freedom [9]. They give principled compositionality [12, 21, 22]. They turn performance questions into eigenvalue or null-space computations over the right semiring [2, 14]. Moreover, they connect nets to automata theory, coding theory and group theory. The case is not a claim of necessity, and the following limits apply: many algebraic reformulations restate known results and pay off only when they yield a new theorem; no algebraic structure removes the complexity barrier of reachability [26]; compositional verification helps only for properties preserved by the composition at hand; and the categorical programme is young.

How this study differs from earlier accounts. Murata's survey [2] and the lecture notes [7, 8] treat the classical linear-algebraic theory, and recent papers each treat one strand. The present study compares the strands under common headings (Table 1), separates reported from proved and from computed statements, and corrects three imprecise statements with proofs. The corrections are modest, but they matter for anyone who builds on the quoted formulations: the integer failure of minimal-support generation (Proposition 2) affects algorithms that enumerate invariants by their supports; the orthogonality formulation (Proposition 3) affects the use of circuit counts as invariants; and the semilinearity remark (Remark 1) affects any argument that treats reachability sets as semilinear.

Open problems.

1) The inclusion between the classes of C-type codes of general and of input-ordinary nets is open beyond nets with at most two places or one transition [9].
2) Which finite semigroups, not only groups, arise as automorphism or endomorphism monoids of net structures appears to be open.
3) A compositional framework that joins the open-net semantics with the semiring-based timed and stochastic semantics is missing; progress exists for compositional invariants of untimed open nets [22].
4) A systematic description of the Hilbert bases of S- and T-invariants for net subclasses, and of when they coincide with the minimal-support invariants, would sharpen the algebraic use of invariants (Proposition 2).

4. Conclusion

The algebraic theory of Petri nets consists of four strands of different maturity: a classical linear-algebraic theory that is established but whose standard formulations need care, a semigroup and code theory that is established for labelled nets and little cited, a categorical theory that is promising for compositional modelling but young, and a max-plus theory that is exact for event graphs only. The original contributions of this paper are limited: three mathematical clarifications with proofs, and computational validation that supports published theorems on the nets tested without proving them. The study recommends an explicit algebraic foundation as a worthwhile research programme, with the limits stated above, and recommends that classical statements be quoted with their precise hypotheses.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

The authors received no financial support for the research, authorship and publication of this paper.

Author Contribution Statement

SP and GKP designed the study and wrote the manuscript. GKP carried out the computations. Both authors read and approved the manuscript.

Data Availability Statement

The programs, the generated nets (exhaustive class and random samples) and the outputs of all computations are provided as Online Resource 1.

Use of Generative AI

During the preparation of this manuscript, the authors used Claude (Anthropic) to draft and revise the text, to check bibliographic details against online records, and to write and run the programs described in Section 2.3. The authors reviewed, edited and verified all AI-assisted content and take full responsibility for its accuracy and originality.

Acknowledgments

The authors thank the Department of Mathematics, MES Keveeyam College, Valanchery, and the Research & Post Graduate Department of Mathematics, MES Kalladi College, Mannarkkad, both affiliated to the University of Calicut, Kerala, India, for institutional support.

References

[1] Carl Adam Petri, "*Communication with Automata*," Ph.D. Dissertation, Institute for Instrumental Mathematics, University, Bonn, Germany, 1962. [Google Scholar]

[2] Tadao Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541-580, 1989. [CrossRef] [Google Scholar] [Publisher Link]

[3] James Lyle Peterson, *Petri Net Theory and the Modeling of Systems*, Englewood Cliffs, NJ, USA: Prentice-Hall, 1981. [Google Scholar]

[4] Wolfgang Reisig, *Petri Nets: An Introduction*, Monographs on Theoretical Computer Science, Heidelberg, Germany: Springer, vol. 4, 1985. [CrossRef] [Google Scholar] [Publisher Link]

[5] Christophe Reutenauer, *The Mathematics of Petri Nets*, Prentice Hall United States, 1990. [Google Scholar] [Publisher Link]

[6] Jorg Desel, and Javier Esparza, *Free Choice Petri Nets*, Cambridge University Press, 1995. [Google Scholar] [Publisher Link]

[7] Jörg Desel, and Wolfgang Reisig, *Place/Transition Petri Nets*, Lectures on Petri Nets I: Basic Models, Springer, vol. 1491, pp. 122-173, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[8] Manuel Silva, Enrique Terue, and José Manuel Colom, *Linear Algebraic and Linear Programming Techniques for the Analysis of Place/Transition Net Systems*, Lectures on Petri Nets I: Basic Models, Springer, vol. 1491, pp. 309-373 1998. [CrossRef] [Google Scholar] [Publisher Link]

[9] Yoshiyuki Kunimochi, "*Algebraic Properties of Petri Net Languages and Codes*," Ph.D. Dissertation, Debrecen University, Hungary, 2009. [Google Scholar] [Publisher Link]

[10] José Meseguer, and Ugo Montanari, "Petri Nets are Monoids," *Information and Computation*, vol. 88, no. 2, pp. 105-155, 1990. [CrossRef] [Google Scholar] [Publisher Link]

[11] Glynn Winskel, "Petri Nets, Algebras, Morphisms, and Compositionality," *Information and Computation*, vol. 72, no. 3, pp. 197-238, 1987. [CrossRef] [Google Scholar] [Publisher Link]

[12] John C. Baez, and Jade Master, "Open Petri Nets," *Mathematical Structures in Computer Science*, vol. 30, no. 3, pp. 314-341, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Gary Cohen et al., "A Linear-System-Theoretic View of Discrete-Event Processes and its use for Performance Evaluation in Manufacturing," *IEEE Transactions on Automatic Control*, vol. 30, no. 3, pp. 210-220, 1985. [CrossRef] [Google Scholar] [Publisher Link]

[14] François Baccelli et al., *Synchronization and Linearity: An Algebra for Discrete Event Systems*, Wiley, 1992. [Google Scholar]

[15] Bernd Heidergott, Geert Jan Olsder, and Jacob W. van der Woude, *Max Plus at Work: Modeling and Analysis of Synchronized Systems*, Princeton University Press, vol. 13, 2006. [Google Scholar] [Publisher Link]

[16] John Hopcroft, and Jean-Jacques Pansiot, "On the Reachability Problem for 5-Dimensional Vector Addition Systems," *Theoretical Computer Science*, vol. 8, no. 2, pp. 135-159, 1979. [CrossRef] [Google Scholar] [Publisher Link]

[17] S. Rao Kosaraju, "Decidability of Reachability in Vector Addition Systems," *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pp. 267-281, 1982. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ernst W. Mayr, "An Algorithm for the General Petri Net Reachability Problem," *SIAM Journal on Computing*, vol. 13, pp. 441-460, 1984. [CrossRef] [Google Scholar] [Publisher Link]

[19] Javier Esparza, and Mogens Nielsen, "Decidability Issues for Petri Nets: A Survey," *arXiv preprint*, pp. 1-20, 2024. [Google Scholar]

[20] Vladimiro Sassone, "On the Category of Petri Net Computations," *Colloquium on Trees in Algebra and Programming*, pp. 334-348, 1995. [CrossRef] [Google Scholar] [Publisher Link]

[21] John C. Baez et al., "Categories of Nets," *Proceedings of 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, Rome, Italy, pp. 1-13, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[22] Benjamin Merlin Bumpus et al., "Additive Invariants of Open Petri Nets," *Compositionality*, vol. 7, pp. 1-20, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Xavier Allamigeon, Marin Boyet, and Stéphane Gaubert, "Computing Transience Bounds of Emergency Call Centers: A Hierarchical Timed Petri Net Approach," *International Conference on Applications and Theory of Petri Nets and Concurrency*, Bergen, Norway, pp. 90-112, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mohammed Sharafath Abdul Hameed, Sofiene Lassoued, and Andreas Schwung, "Learnable Petri Net Neural Network using Max-Plus Algebra," *Machine Learning and Knowledge Extraction*, vol. 7, no. 3, pp. 1-27, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[25] Jérôme Leroux, "The Reachability Problem for Petri Nets is Not Primitive Recursive," *IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, Denver, CO, USA, pp. 1241-1252, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] Wojciech Czerwiński, and Łukasz Orlikowski, "Reachability in Vector Addition Systems is Ackermann-Complete," *IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, Denver, CO, USA, pp. 1229-1240, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Richard J. Lipton, "*The Reachability Problem Requires Exponential Space*," Research Report 62, Department of Computer Science, Yale University, 1976. [Google Scholar]

[28] Wojciech Czerwiński et al., "The Reachability Problem for Petri Nets is Not Elementary," *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 24-33, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[29] Richard M. Karp, and Raymond E. Miller, "Parallel Program Schemata," *Journal of Computer and System Sciences*, vol. 3, no. 2, pp. 147-195, 1969. [CrossRef] [Google Scholar] [Publisher Link]

[30] Alexander Schrijver, *Theory of Linear and Integer Programming*, Wiley, 1986. [Google Scholar]

[31] Frederic Commoner et al., "Marked Directed Graphs," *Journal of Computer and System Sciences*, vol. 5, no. 5, pp. 511-523, 1971. [CrossRef] [Google Scholar] [Publisher Link]

[32] Rebekah Aduddell et al., "A Compositional Account of Motifs, Mechanisms, and Dynamics in Biochemical Regulatory Networks," *Compositionality*, vol. 6, no. 2, pp. 1-30, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[33] Mohammed Alahmadi et al., "Structured Acyclic Nets," *arXiv*, pp. 1-35, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] Eugene L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976. [Google Scholar] [Publisher Link]