INFORMATION AND COMPUTATION **88**, 105–155 (1990)

Petri Nets Are Monoids*

JOSÉ MESEGUER[†] AND UGO MONTANARI[‡]

SRI International, Menlo Park, California 94025

Petri nets are widely used to model concurrent systems. However, their composition and abstraction mechanisms are inadequate: we solve this problem in a satisfactory way. We start by remarking that place/transition Petri nets can be viewed as ordinary, directed graphs equipped with two algebraic operations corresponding to parallell and sequential composition of transitions. A distributive law between the two operations captures a basic fact about concurrency. New morphisms are defined, mapping single, atomic transitions into whole computations, thus relating system descriptions at different levels of abstraction. Categories equipped with products and coproducts (corresponding to parallel and nondeterministic compositions) are introduced for Petri nets with and without initial markings. Petri net duality is expressed as a duality functor, and several new invariants are introduced. A tensor product is defined on nets, and their category is proved to be symmetric monoidal closed. This construction is generalized to a large class of algebraic theories on graphs. These results provide a formal basis for expressing the semantics of concurrent languages in terms of Petri nets. They also provide a new understanding of concurrency in terms of algebraic structures over graphs and categories that should apply to other models besides Petri nets and thus contribute to the conceptual unification of concurrency.

1. INTRODUCTION

Petri nets are the first model of concurrent systems which has been developed and, in their various evolutions, the most heavily used in many applications. They have also been the object of many contributions in the literature (for an extensive list of references see Drees *et al.*, 1986). Recently, a renewed interest in Petri nets has been stirred up by the so-called true concurrency approach to the semantics of concurrent systems

* Supported by Office of Naval Research Contracts N00014-86-C-0450 and N00014-88-C-0618, NSF Grant CCR-8707155 and by a grant from the System Development Foundation. A summarized version of this paper appeared in Meseguer and Montanari (1988).

[†] Also Center for the Study of Language and Information, Stanford University, Stanford, California 94305.

[‡] Dipartimento di Informatica, Universitá di Pisa, I-56100 Pisa, Italy; research performed while on leave at SRI International.

0890-5401/90 $3.00

(Boudol and Castellani, 1988; Degano *et al.*, 1988; Degano and Montanari, 1987; Winskel, 1987; Olderog, 1987; van Glabbeek and Vaandrager, 1977).

While the algebraic structure of Petri nets has been extensively investigated, we feel that a key point has been missed: a place/transition Petri net is simply an ordinary, directed graph equipped with two algebraic operations. More precisely, a net provides the generators of the algebraic structure. Several well-known constructions (e.g., the case graph, the firing sequences, the non-sequential processes, etc.) correspond to closure constructions with respect to the algebraic operations.

Consider, for example, the place transition Petri net in Fig. 1. It has a set of places $S=\{a, b, c, d, e, f\}$ and a set of transitions $T=\{t, t'\}$. Numbers on the incoming arrows of a transition specify how many tokens are consumed from each place when the transition fires, and numbers on the outgoing arrows specify how many tokens are generated as a consequence of the transition. The "state" of the net is determined by the number of tokens stored in each place. The left-hand side picture describes a state with two tokens in a, four in b, and three in c. In a Petri net, several transitions can fire *concurrently*. The picture on the right describes the state reached after the concurrent firing of t and t'.

The point is that this Petri net can be understood as an ordinary graph whose set of nodes is the free commutative monoid $S^{\oplus}$ generated by the set S of places (we use additive notation, so a typical element of $S^{\oplus}$ is for example $3a \oplus 2c \oplus 7e$; in the computer science literature such elements are called finite multisets or "bags" and addition is understood as union). The transitions then correspond to arrows of the graph. In this case there are just two arrows,

$$t: a \oplus 2b \rightarrow 3d \oplus 2e$$

$$t': b \oplus 3c \rightarrow e \oplus 4f.$$

Addition can naturally be extended to transitions. For example, we can represent the parallel firing of t and t' by the arrow

$$t \oplus t': a \oplus 3b \oplus 3c \rightarrow 3d \oplus 3e \oplus 4f.$$

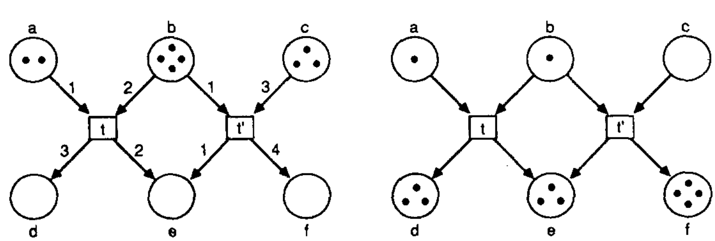

FIG. 1. A place/transition net before and after the concurrent firing of t and t'.

Therefore, our first operation, $\oplus$, together with a zero element, yields a commutative monoid structure on a graph, in the sense that the monoid structure is defined on both nodes and arcs, and that the source and target functions ∂_0 and ∂_1 from arcs to nodes are monoid homomorphisms.

The commutative monoid structure on the nodes is free, having the places of the given net as generators. The commutative monoid structure on the arcs may also be free, and in that case the meaning of $\oplus$ is the parallel, independent composition of transitions. In general, however, the monoid of the arcs need not be free: for example, it may take into account a synchronization algebra defined on the transitions.

It is also convenient to consider *reflexive* graphs, i.e., graphs where every node is the source and the target of an associated identity arc, which is interpreted as an *idle* transition. For example, the identity arc $2a: 2a \to 2a$ is interpreted as the idleness of two tokens in place a. In this way, we can represent the concurrent transition from state $2a \oplus 4b \oplus 3c$ to state $3d \oplus 3e \oplus 4f$ in Fig. 1 by the arc,

$$a \oplus b \oplus t \oplus t' : 2a \oplus 4b \oplus 3c \to a \oplus b \oplus 3d \oplus 3e \oplus 4f.$$

The commutative Petri monoid on a reflexive graph generated from a Petri net by additive closure is a well-know object in Petri net theory: it is called the *case graph.*

The second algebraic operation is even simpler: it is the concatenation of the arcs of the graph and is denoted by a semicolon. Closure with respect to this operation of sequential composition straightforwardly generates new transitions corresponding to computations of the given Petri net. However, the interesting point here is that we close with respect to both sum and sequential composition *at the same time*, thus obtaining a more general notion of computation.

The resulting structures can be seen as small categories, here called *Petri categories*, where the morphisms are computations. This naturally suggests making the sum operation functorial, i.e., making it respect identities and sequential composition. Therefore, a Petri category is a commutative monoid structure on a small category, with the distinguishing feature that the commutative monoid of objects is free. This justifies our title, since Petri nets are monoids both on graphs and, by additional closure, on categories. Indeed, if in the definition of Petri category we relax the freeness requirement, we obtain the more general notion of a commutative monoid structure on a category. Such structures are usually called *strict symmetric monoidal categories*, and the monoid homomorphisms are called *strict monoidal functors* (MacLane, 1971).

The key law in Petri categories is the following *distributive property*. Given $\alpha: u \to v$, $\alpha': u' \to v'$, $\beta: v \to w$, $\beta': v' \to w'$, we have

$$(\alpha; \beta) \oplus (\alpha'; \beta') = (\alpha \oplus \alpha'); (\beta \oplus \beta'). \tag{1}$$

We feel that this law captures a rather basic fact about concurrency: the parallel composition of two given independent computations has the same effect as a computation whose steps are the parallel compositions of the steps of the given computations.

A derived property may make the point clearer. Given $\alpha: u \to v$, $\beta: u' \to v'$, we have

$$(\alpha \oplus u'); (v \oplus \beta) = (u \oplus \beta); (\alpha \oplus v').$$

This is the well-known property that two independent (concurrent) transitions can be executed in any order. The fact that they are concurrent is expressed by the fact that the places involved in one transition are idle while the other transition takes place: the two transitions are not causally related.

Computations of Petri categories are closely related to *nonsequential processes* (Goltz and Reisig, 1983; Reisig, 1985), a well-known, classical concept apt to describe the concurrent behaviors of Petri nets. The two notions coincide (Degano *et al.*, 1989a) in the important case of *safe computations*, i.e., computations where two instances of the same place are never concurrent. When actions are introduced as labels for transitions, computations of Petri categories are also similar to *concurrent histories*, a notion developed earlier in a different context by the second author in joint work with P. Degano (Degano and Montanari, 1987). The above derived property is called commutativity in (Degano and Montanari, 1987), where it plays a pivotal role. Winkowski (1982) introduced two operations of sequential and parallel composition of processes; however, his parallel composition is partial, and the approach is restricted to safe computations.

The earliest use of free monoidal categories in computer science was probably made by Hotz (1965). In (Meseguer and Sols, 1975), linear algebra models of free monoidal categories were used to characterize sequential and parallel compositions of nondeterministic and probabilistic automata and switching networks. As part of their linear algebra approach to nondeterminism and concurrency, Main and Benson (1984) also advocate the use of monoidal categories to formalize sequential and parallel composition.

The formal development we are proposing for the above ideas relies on category theory. In the case of Petri nets, the use of category theory is justified by very concrete motivations. In fact, Petri nets have been often considered inadequate since, at least in their original version, they are not equipped with composition operations and with an abstraction mechanism. The categorical approach due to Winskel (1984, 1987) provides the former, since the categorical constructions of product and coproduct correspond to parallel and nondeterministic composition (respectively) (Winskel, 1984) for languages like CCS (Milner, 1985). A further benefit that category theory provides is very powerful techniques for relating different classes of models (i.e., different categories). This is an important advantage in the case of concurrency, where different models of the same system are often considered for different purposes. For instance, Petri nets, occurrence nets, several versions of event structures, transition systems, synchronization trees, etc. can be given a categorical structure, and their semantic relations can be profitably expressed (typically as coreflections) in the language of categories (Winskel, 1984, 1986).

In this paper, besides directly using categories as a model of computation in Petri categories, we broaden the applicability of Winskel's contribution, and, in addition, we show that the abstraction problem also has a simple and natural solution by providing new very general morphisms corresponding to the notion of implementation.

Our view of a Petri net as a reflexive graph equipped with two operations ($\oplus$ and ;) immediately suggests that morphisms are reflexive graph morphisms (i.e., mappings of arcs and nodes respecting sources, targets, and identity arcs) which furthermore respect the operations of parallel sum and sequential composition.

These morphism are, to our knowledge, new in the context of Petri nets.[1] They are a decisive improvement over the strongly restrictive versions previously proposed in the literature, since they allow simulations where single transitions of the specification correspond to whole computations of the implementation. Relating system descriptions at different levels of abstraction has, admittedly, been one of the main goals of the theory of Petri nets from its very beginning, a goal which has never been fully achieved.

Furthermore, our categorical approach has the advantage of suggesting completely new constructions for Petri nets. As important examples, we express Petri net duality as a duality functor and we make explicit a *symmetric monoidal closed category* structure, where nets are closed under a *function space* construction with an associated (noncartesian) product.

Considering our approach a little more in detail, in Section 2 we naturally define a hierarchy of categories, where the objects have richer and richer algebraic structures: Petri nets, pointed Petri nets (nets with a zero transition), Petri monoids, reflexive nets, Petri categories. All these

[1] See [Hinderer, 1982] for an early attempt to use category theory to obtain a general notion of morphism. Although similar in spirit to the notion that we propose, the basic link between Petri nets and categories was, in hindsight, unsatisfactory.

categories are obviously related by forgetful functors, whose left adjoints provide the closure constructions with respect to the added operations. However, to capture the morphisms we are interested in, it is *not* necessary to make the additional structure explicit. It is sufficient to consider as objects ordinary Petri nets and to equip them with the morphisms defined on their closures. This approach, described in Section 3, generates a hierarchy of categories, where the objects are the same, the ordinary Petri nets, but where more general types of morphisms become available as more and more structure is taken into account. All these categories have products and coproducts.

The graph definition we have followed until now does not require an initial node. In fact, the initial marking of a Petri net is often considered inessential. This makes the formal treatment simpler and nicer. In fact, Winskel (1987) proved that his categories, relying on a Petri net definition which includes an initial marking, do not have coproducts: he restricted his treatment to safe nets and safe morphisms to guarantee the existence of coproducts. Lacking coproducts is a serious drawback, since, as we noticed, they correspond semantically to nondeterministic compositions.

However, an initial state is needed whenever Petri nets are used for defining the operational semantics of concurrent languages (Degano *et al.*, 1988; Degano and Montanari, 1987; Winskel, 1984; Olderog, 1987; van Glabbeek and Vaandrager, 1987). In Section 4, we extend the applicability of our results also to this important case by adding an initial marking, but we require it to be a set (instead of a multiset) of places. No restrictions whatsoever are placed on nets, nor on morphisms, except that they preserve the initial marking. We then show the existence of products and coproducts for all morphisms, includig those allowing a change in the level of abstraction. In practice, our restriction involves no loss in generality: we easily define a functor which adds to a given multiset-marked net a new initial place and a starting transition. This functor lands in a full subcategory, with unreachable initial markings, equipped with products and coproducts, which is our best candidate for language definition applications.

The morphisms proposed by Winskel (1987) called synchronous morphisms, asynchronous morphisms, and homomorphisms, correspond more or less to the first three steps of the hierarchy described in Section 2: Petri nets, pointed Petri nets, and Petri monoids (respectively). However, homomorphisms are hardly used in (Winskel, 1987) and indeed doubts about their usefulness are raised; synchronous and asynchronous morphisms are introduced, as restrictions of the latter, in a somewhat *ad hoc* manner. Indeed, Winskel's treatment of the category of Petri nets and homomorphisms (based on ideas by Reisig: a net is a two-sorted algebra on the multisets $\mathcal{N}^T$ and $\mathcal{N}^S$ with operations ${}^\bullet(\_), (\_)^\bullet : \mathcal{N}^T \to \mathcal{N}^S$

and a constant $M \in \mathcal{N}^S$) recognizes the algebraic nature of nets but does not take full advantage of their *graph* structures.

Besides requiring the initial marking to be a set rather than a multiset, a difference with Winkel's approach is that we consider in most of the paper finite multisets (viewed as elements of a free commutative monoid) rather than arbitrary ones, so that our transitions have finitary preconditions and postconditions; as a consequence, we do not need any restrictions to obtain a category, whereas Winskel has to add conditions on the net to make sure that homomorphisms compose. However, an entirely parallel development of our ideas can be obtained by introducing transitions having in their pre/postconditions both an infinite number of places and places with an infinite multiplicity. We explain in Section 7.4.2 that most of our results hold in this case as well. On the other hand, our Petri monoid category is more general, in that the monoid structure of transitions need not be free. The usefulness of this additional generality may reflect a synchronization algebra and also becomes apparent when we consider Petri categories (where the monoid of transitions is not free for the existence of laws like (1)), which have no counterpart in Winskel's work.

Our approach of viewing Petri nets as ordinary graphs may appear to obscure the well-known and fruitful fact that nets can be dualized by regarding transitions as places and places as transitions. The opposite is the case. In Section 5 we express Petri net duality as a duality functor. We then give a geometrical interpretation of T-invariants and their properties through a very general notion of a *Loop* functor, and we use duality to give a functorial account of S- and T-invariants. Using elementary algebra, we also derive algebraic relations between the groups of S- and T-invariants of a Petri net and associate to a Petri net N two other groups, $S^{\text{ab}}_{\text{mult}}(N)$ and $T^{\text{ab}}_{\text{mult}}(N)$ that apparently are new.

It is well known that a tensor product $A \otimes B$ can be defined in the category ***CMon*** of commutative monoids so that, up to natural isomorphisms, $\otimes$ is associative, commutative, and has $\mathcal{N}$ as an identity. It is also well known that the monoid homomorphisms from A to B form a commutative monoid $[A \to B]$ and that there is a natural isomorphism

$$\boldsymbol{CMon}((A \otimes B), C) \simeq \boldsymbol{CMon}(A, [B \to C]);$$

in other words, the category of commutative monoids is a symmetric monoidal closed category (MacLane, 1971). This is just like a cartesian closed category except that the product $A \otimes B$ is not the categorical product. Since "Petri nets are monoids," this result can be extended to nets: In Section 6 we give tensor product and function space constructions for Petri nets and prove that their category is symmetric monoidal closed. In fact, the definition works even better (without need for a finiteness condition) for the more general case where the monoid of nodes in the graph need not be free. This construction seems to be completely new. Furthermore, we generalize this result to the categories of commutative monoids on graphs, on reflexive graphs, and on categories.

Finally, in Section 7 the basic constructions of Section 6 (and with them most of the results of the paper) are generalized to a large class of algebraic theories on graphs. The required condition is that the algebraic theories of both nodes and arcs be commutative and that the source and target maps be homomorphisms. The development is carried out in the framework of the theory of *commutative monads* (Eilenberg and Moore, 1965; MacLane, 1971; Manes, 1976; Linton, 1966; Kock, 1971). This "meta" result makes our theory applicable to a variety of interesting cases: in Section 7.4 we give examples concerning fuzzy nets, infinitary nets, and probabilistic nets, and in Section 8 we indicate an extension to term rewriting systems.

Although we have for the most part concentrated on the case of Petri nets, the general new concept that emerges from the present work is that of *transition systems as graphs with algebraic structure*. Computations of a transition system then appear as morphism of a path category generated by its graph. This path category will be endowed with an algebraic structure similar to that of the transition system. For Petri nets, the relevant algebraic structure is that of a commutative monoid, and therefore computations have a strict symmetric monoidal category structure, but this is just a particular case. Other algebraic structures besides that of monoid are possible and natural. Considerations of this kind should lead to a general algebraic (meta) model of true concurrency of wide applicability.

Regarding prerequisites, we assume some acquaintance with basic notions of category theory such as category, functor, products, coproducts, etc. However, we give intuitive explanations of adjoints, cartesian and monoidal closed categories, and monads, when each notion is first encountered. An excellent reference is (MacLane, 1971). Section 5 assumes an undergraduate level acquaintance with groups, rings, and modules.

2. Adding Monoid and Category Structure to Petri Nets

2.1. *Petri Nets*

The standard definition of place/transition net (Reisig, 1985; Winskel, 1987) is as follows: A place/transition (P/T) net is a triple $\langle S, T, F \rangle$, where

- S is a set of *places*;
- T is a set of *transitions*;
- $F\colon (S \times T) + (T \times S) \to \mathcal{N}$ is a multiset called the *causal dependency*

relation. (Here $\mathcal{N}$ denotes the natural numbers and $+$ denotes disjoint union of sets.)

In the rest of this paper a Petri net *will always mean the general case of a* place/transition net. Sometimes special requirements (like global finiteness conditions, or limitations on the capacity of the places) are added.

As explained in Section 1, we want to see Petri nets exactly as graphs.

DEFINITION 1. A graph G is a set T of arcs, a set V of nodes and two functions ∂_0 and ∂_1 called *source* and *target*, respectively:

$$\partial_0, \partial_1 : T \to V.$$

A morphism h from G to G' is a pair of functions $\langle f, g \rangle$, $f: T \to T'$ and $g: V \to V'$ such that:

$$g \circ \partial_0 = \partial_0' \circ f$$

$$g \circ \partial_1 = \partial_1' \circ f.$$

This, with the obvious componentwise composition of morphisms, defines the category ***Graph***. We follow the usual notation and write $t: u \to v$ to denote $\partial_0(t) = u$, $\partial_1(t) = v$ for $t \in T$.

DEFINITION 2. A (place/transition) *Petri net* is a graph where the arcs are called *transitions* and where the set of nodes is the free commutative monoid $S^{\oplus}$ over a set of *places* S:

$$\partial_0, \partial_1 : T \to S^{\oplus}.$$

A *Petri net morphism* is a graph morphism $\langle f, g \rangle$, where g is a monoid homomorphism (i.e., leaving 0 fixed and respecting the monoid operation $\oplus$). This defines a category ***Petri***.

The elements of $S^{\oplus}$ will be represented as formal sums $n_1 a_1 \oplus \cdots \oplus n_k a_k$ with the order of the summands being immaterial, with the a_i in S, the n_i in $\mathcal{N}$, addition defined by $(\bigoplus_i n_i a_i) \oplus (\bigoplus_i m_i a_i) = (\bigoplus_i (n_i + m_i) a_i)$ and 0 as the neutral element.

It is easy to see that our definition coincides with the standard definition, if we require there that for each $t \in T$ the set $\{s \mid F(s, t) \neq 0 \text{ or } F(t, s) \neq 0\}$ is finite.

For example, from

$$S = \{a, b, c\}, \qquad T = \{t\}$$

$$F(a, t) = 2, \qquad F(b, t) = 1, \qquad F(t, c) = 2, \qquad F = 0 \text{ elsewhere},$$

we obtain the graph with nodes $\{a, b, c\}^{\oplus}$ and the single arrow $t: 2a \oplus b \to 2c$.

There is an obvious forgetful functor ***Petri*** $\to$ ***Graph*** that forgets about the monoid structure of the nodes. There is an associated free construction sending a graph G to the free Petri net $N(G)$ generated by G. "Freenes" of course means that there is a graph morphism $\eta_G: G \to N(G)$ injecting the generators G into $N(G)$ such that given a Petri net M and a graph morphism $h: G \to M$ there is a unique Petri net morphism $\bar{h}: N(G) \to M$ extending h, i.e., such that $\bar{h} \circ \eta_G = h$. A free construction of this kind always defines a functor "going the other way" and called the *left adjoint* to the given forgetful functor (MacLane, 1971) (dually, the forgetful functor is called the *right adjoint* of its free construction functor). In our case, the left adjoint is a functor N: ***Graph*** $\to$ ***Petri*** associating to a graph $G = (\partial_0, \partial_1: T \to V)$ the Petri net $N(G) = (\partial_0, \partial_1: T \to V^{\oplus})$. In what follows, since free constructions associated to forgetful functors are exactly the same thing as left adjoints, we will just say that there is a left adjoint for a given forgetful functor and indicate the result of the free construction on the objects (the inclusion of generators η tends to be obvious (typically a set-theoretic inclusion) and can safely be left implicit).

The categorical product in the category ***Graph*** of two graphs

$$G = (\partial_0, \partial_1: T \to V) \qquad \text{and} \qquad G' = (\partial'_0, \partial'_1: T' \to V')$$

is the graph

$$G \times G' = (\partial_0 \times \partial'_0, \partial_1 \times \partial'_1: T \times T' \to V \times V').$$

For Petri nets $N = (\partial_0, \partial_1: T \to S^{\oplus})$ and $N' = (\partial'_0, \partial'_1: T' \to S'^{\oplus})$, their product as graphs

$$N \times N' = (\partial_0 \times \partial'_0, \partial_1 \times \partial'_1: T \times T' \to S^{\oplus} \times S'^{\oplus})$$

is also a Petri net, since

$$S^{\oplus} \times S'^{\oplus} \simeq (S + S')^{\oplus} \simeq S^{\oplus} \oplus S'^{\oplus},$$

i.e., finite products and coproducts of free commutative monoids coincide. The Petri net $N \times N'$ is clearly the categorical product in ***Petri*** and is called the *synchronous product* of the nets N and N'. Intuitively, the synchronous product of two Petri nets is the result of a composition operation with synchronization: The places of the result are the union of the places of the factors, while the transitions in the synchronous product are pairs (i.e., synchronizations) of the given transitions.

The category ***Petri*** has also coproducts,

$$N \oplus N' = ([\partial_0, \partial'_0], [\partial_1, \partial'_1]: T + T' \to (S + S')^{\oplus}),$$

where $[\partial_i, \partial'_i]$ denotes the function induced on the coproduct $T+T'$ by functions ∂_i and ∂'_i on the pieces.

Intuitively, the coproduct of two Petri nets is the result of a composition operation without synchronization: the two nets are just laid aside without interaction. We will see that in the case of marked nets (i.e., nets with initial state, discussed in Section 4) the meaning of coproducts is, more suggestively, that of nondeterministic choice composition.

The initial net has no transitions and no places, while the final net has one transition and no places. The construction of the coproduct of two Petri nets generalizes to arbitrary families of nets.

Petri has neither arbitrary limits nor arbitrary colimits. This is due to the fact that the category of free commutative monoids lacks arbitrary limits and colimits. However, dropping the freeness requirement for the monoid of nodes leads to a bigger category ***GralPetri*** that has all limits and colimits.

2.2. *Pointed Petri Nets*

In the category ***Petri***, a map $\langle f, g\rangle : N \to N'$ maps each transition t of N to a transition $f(t)$ of N'. We might, however, want to allow for certain transitions to be *erased* by a mapping. This would correspond to making the map $f: T \to T'$ partial. An approach which is completely equivalent from a semantical point of view, but more convenient technically, is to add a special element 0 to T, making it into a pointed set. Maps between pointed sets are required to leave 0 fixed, and thus directly correspond to partial functions between the original sets. The commutative monoid $S^{\oplus}$ is already a pointed set considering as special element the 0 element of the $\oplus$ operation.

DEFINITION 3. A *pointed Petri net* consists of a Petri net where the set of transitions is a pointed set $(T, 0)$, $0 \in T$, the commutative monoid $S^{\oplus}$ is viewed as a pointed set, and

$$\partial_0, \partial_1 : (T, 0) \to S^{\oplus}$$

are pointed set maps. A *pointed Petri net morphism* is a Petri net morphism $\langle f, g\rangle$, where f is a map of pointed sets. This defines a category $\boldsymbol{Petri_0}$.

There is an obvious forgetful functor $\boldsymbol{Petri_0} \to \boldsymbol{Petri}$ that forgets about the pointed set structure of the transitions. This functor has a left adjoint $(\_)_0 : \boldsymbol{Petri} \to \boldsymbol{Petri_0}$ that associates to a net $N = (\partial_0, \partial_1 : T \to S^{\oplus})$ the pointed net

$$N_0 = (\overline{\partial_0}, \overline{\partial_1} : (T + \{0\}, 0) \to S^{\oplus})$$

with $\overline{\partial_i} = \partial_i$ on T, and $\overline{\partial_i}(0) = 0$.

As in ***Petri***, the product of two pointed Petri nets as graphs has an obvious pointed net structure and yields the categorical product in $\boldsymbol{Petri}_0$.

Coproducts are also easy;

$$N \oplus N' = ([\partial_0, \partial_0'], [\partial_1, \partial_1'] : (T, 0) \oplus (T', 0') \rightarrow (S + S')^{\oplus}),$$

where $(T, 0) \oplus (T', 0')$ is the coproduct of pointed sets, i.e., the disjoint union $T + T'$, except that 0 and $0'$ are identified.

2.3. *Petri Monoids*

DEFINITION 4. A *Petri commutative monoid M* consists of a Petri net where the set of transitions is a commutative monoid $(T, +, 0)$ and where

$$\partial_0, \partial_1 : (T, +, 0) \rightarrow S^{\oplus}.$$

are monoid homomorphisms. A *Petri commutative monoid morphism* is a Petri net morphism $\langle f, g \rangle$, where f is a monoid homomorphism. This defines a category ***CMonPetri***.

There are forgetful functors

$$\boldsymbol{CMonPetri} \rightarrow \boldsymbol{Petri}_0 \rightarrow \boldsymbol{Petri},$$

each with a left adjoint. We shall denote by $(\_)^{\oplus} : \boldsymbol{Petri} \rightarrow \boldsymbol{CMonPetri}$ the left adjoint of their composition. It associates to a Petri net $N = (\partial_0, \partial_1 : T \rightarrow S^{\oplus})$ the Petri commutative monoid $N^{\oplus} = (\overline{\partial_0}, \overline{\partial_1} : T^{\oplus} \rightarrow S^{\oplus})$, where $\overline{\partial_0}$ and $\overline{\partial_1}$ are the unique monoid homomorphisms extending ∂_0 and ∂_1.

In general, however, Petri monoids need not be free. Nonfreeness may reflect a synchronization structure. Assume, for example, a free abelian group $\mathscr{X}\{A\}$ generated by a set A of *basic actions* as in Milner's (1982) approach so that addition of one action with its inverse corresponds to synchronization. Then, given a Petri net $N = (\partial_0, \partial_1 : T \rightarrow S^{\oplus})$ together with a *labeling map* $l: T \rightarrow \mathscr{X}\{A\}$ we can define a Petri monoid N^l with same set of places as N but with a monoid structure that reflects the synchronization information provided by the labeling. N^l has a monoid of transitions the quotient monoid T^l obtained from $T^{\oplus}$ by imposing the relations

$$\alpha = \alpha'$$

for all $\alpha, \alpha' \in T^{\oplus}$ such that $\bar{l}(\alpha) = \bar{l}(\alpha') = 0$, and $\overline{\partial_i}(\alpha) = \overline{\partial_i}(\alpha')$ for $i = 0, 1$. The requirement $\overline{\partial_i}(\alpha) = \overline{\partial_i}(\alpha')$ for $i = 0, 1$ ensures the existence of homomorphisms $\gamma_i: T^l \to S^{\oplus}$, $i=0, 1$ such that $\overline{\partial_i} = \gamma_i \circ q$, where $q: T^{\oplus} \to T^l$ is the quotient homomorphism. We then define $N^l = (\gamma_0, \gamma_1: T^l \to S^{\oplus})$.

As in ***Petri***$_0$ and ***Petri***, the product $M \times M'$ as graphs of two Petri commutative monoids M and M' has an obvious Petri commutative monoid structure and yields the categorical product in ***CMonPetri***.

Coproducts are also straightforward;

$$M \oplus M' = (\partial_0 \oplus \partial'_0, \partial_1 \oplus \partial'_1 : (T, +, 0) \oplus (T', +', 0') \to S^{\oplus} \oplus S'^{\oplus}),$$

where $(T, +, 0) \oplus (T', +', 0')$ is the coproduct of commutative monoids. It is not hard to check that the coproduct of two arbitrary commutative monoids coincides with their product. This implies also the same property in ***CMonPetri***, i.e., $M \times M' = M \oplus M'$.

The Petri commutative monoid,

$$0 = (1_0, 1_0 : 0 \to 0)$$

is the initial and final object in ***CMonPetri***.

It is fruitful to observe that ***CMonPetri*** is a full subcategory of ***CMonGraph***, where objects of ***CMonGraph*** are commutative monoid structures on graphs. In a compact form, a commutative monoid structure on a graph G can be described as a pair of graph morphisms $+: G^2 \to G$ and $0: 1 \to G$ satisfying the commutative monoid equations (expressed as commutative diagrams). The graph 1 is the terminal object of ***Graph*** and has one edge and one node. ***CMonPetri*** is just the full subcategory determined by those monoid structures whose monoid of nodes is free. This justifies our claim in the title that "Petri nets are monoids." This claim will be further supported by adding a sequential composition operator since this will make Petri nets monoids not only on graphs but also on categories. Of course, ***CMonGraph*** has all limits and colimits, so that those limits or colimits that do not exist in ***CMonPetri*** have a meaning in ***CMonGraph***.

2.4. *Reflexive Petri Nets*

A *reflexive graph* G is one in which every node v has a specified arrow $id(v): v \to v$. Reflexive structure is very useful at the Petri net level. It is implicit in the so-called *case graph* of a net N. As we shall see, the case graph is just a free construction that freely adds additional structure to a Petri net. All reflexive Petri net structures live over the category ***RGraph*** of reflexive graphs with objects graphs $G = (\partial_0, \partial_1 : T \to V)$ together with a function $id: V \to T$ such that $\partial_0 \circ id = \partial_1 \circ id = 1_V$. Reflexive graph morphisms $\langle f, g \rangle : (G, id_G) \to (G', id_{G'})$ are graph morphisms satisfying the additional requirement $f \circ id_G = id_{G'} \circ g$.

DEFINITION 5. A *reflexive Petri net* consists of a Petri net N which in addition is a reflexive graph. A *reflexive Petri net morphism* is a Petri net morphism that is also a morphism of reflexive graphs. Similarly, a *reflexive Petri commutative monoid* is a Petri commutative monoid with a reflexive graph structure such that $id: S^{\oplus} \to (T, +, 0)$ is a monoid homomorphism, and a *reflexive Petri commutative monoid morphism* is a Petri commutative monoid morphism that is also a morphism of reflexive graphs. This defines categories ***RPetri*** and ***CMonRPetri*** with obvious forgetful functors,

$$\begin{tikzcd} \boldsymbol{CMonRPetri} \arrow[r] \arrow[d] & \boldsymbol{RPetri} \arrow[r] \arrow[d] & \boldsymbol{RGraph} \arrow[d] \\ \boldsymbol{CMonPetri} \arrow[r] & \boldsymbol{Petri} \arrow[r] & \boldsymbol{Graph}. \end{tikzcd}$$

Note that a reflexive Petri net is naturally endowed with a pointed Petri net structure so that there is no point in defining a category $\boldsymbol{RPetri_0}$, since this coincides with ***RPetri***.

All the above functors have left adjoints. The most interesting of them is the left-adjoint $\mathscr{C}[\_]$ to the forgetful functor $\boldsymbol{CMonRPetri} \to \boldsymbol{Petri}$. $\mathscr{C}[\_]$ associates to each Petri net N its *case graph* $\mathscr{C}[N]$. For $N = (\partial_0, \partial_1 : T \to S^{\oplus})$ we define

$$\mathscr{C}[N] = ([1_{S^{\oplus}}, \overline{\partial_0}], [1_{S^{\oplus}}, \overline{\partial_1}] : S^{\oplus} \oplus T^{\oplus} \to S^{\oplus}, j_1 : S^{\oplus} \to S^{\oplus} \oplus T^{\oplus}),$$

where $id = j_1$ is the coproduct inclusion.

In all these categories, the cartesian product as graphs has a unique structure making it into the categorical product. Coproducts also exist everywhere. For example, for (M, id), (M', id') in ***CMonRPetri*** we have, $(M, id) \oplus (M', id') = (M \oplus M', id \oplus id')$, where $M \oplus M'$ is the coproduct in ***CMonPetri***; also, $(M, id) \oplus (M', id') = (M, id) \times (M', id')$. For (N, id), (N', id') in ***RPetri***, $(N, id) \oplus (N', id')$ has transitions $(T - \mathrm{Im}(id)) + (T' - \mathrm{Im}(id')) + (\mathrm{Im}(id) \times \mathrm{Im}(id'))$ and an identity map given by $id \times id'$.

The category ***CMonRPetri*** is the full subcategory of the category ***CMonRGraph*** of commutative monoid structures on reflexive graphs determined by those structures whose commutative monoid of nodes is free.

2.5. *Petri Categories*

DEFINITION 6. A *Petri category* consists of a reflexive Petri commutative monoid

$$C = (\partial_0, \partial_1 : (T, +, 0) \to S^{\oplus}, id)$$

together with a partial function $\_;\_ : T \times T \to T$ which is defined exactly for those pairs (α, β) such that $\partial_1(\alpha) = \partial_0(\beta)$. In addition, the following axioms are satisfied (whenever the compositions $\alpha; \beta$, etc. are defined):

1. $\partial_0(\alpha; \beta) = \partial_0(\alpha)$ and $\partial_1(\alpha; \beta) = \partial_1(\beta)$
2. $\alpha; id(\partial_1(\alpha)) = \alpha$ and $id(\partial_0(\alpha)); \alpha = \alpha$
3. $(\alpha; \beta); \gamma = \alpha; (\beta; \gamma)$
4. Given $\alpha: u \to v$, $\alpha': u' \to v'$, $\beta: v \to w$, $\beta': v' \to w'$, we have

$$(\alpha + \alpha'); (\beta + \beta') = (\alpha; \beta) + (\alpha'; \beta').$$

Given two Petri categories C and D a Petri category morphism from C to D is a morphism $\langle f, g \rangle: C \to D$ of their underlying reflexive Petri monoids such that $f(\alpha; \beta) = f(\alpha); f(\beta)$. This determines a category ***CatPetri***.

There are forgetful functors

$$\textbf{\textit{CatPetri}} \to \textbf{\textit{CMonRPetri}} \to \textbf{\textit{CMonPetri}} \to \textbf{\textit{Petri}}_0 \to \textbf{\textit{Petri}}$$

and also similar functors to the remaining categories of reflexive Petri nets. All of them have left adjoints. We shall describe the left adjoint $\mathscr{T}[\_]: \textbf{\textit{Petri}} \to \textbf{\textit{CatPetri}}$ for their composition. Given a net $N = (\partial_0, \partial_1: T \to S^{\oplus})$ the Petri category $\mathscr{T}[N]$ is inductively defined by the following rules of inference:

$$\frac{t: u \to v \text{ in } N}{t: u \to v \text{ in } \mathscr{T}[N]} \qquad \frac{u \text{ in } S^{\oplus}}{u: u \to u \text{ in } \mathscr{T}[N]}$$

$$\frac{\alpha: u \to v,\ \beta: v \to w \text{ in } \mathscr{T}[N]}{\alpha; \beta: u \to w \text{ in } \mathscr{T}[N]} \qquad \frac{\alpha: u \to v,\ \alpha': u' \to v' \text{ in } \mathscr{T}[N]}{\alpha \oplus \alpha': u \oplus u' \to v \oplus v' \text{ in } \mathscr{T}[N]}$$

with $\oplus$ and 0 subject to the commutative monoid equations, with identities given by $id(u) = u$, and with $\_;\_$ and id satisfying the equations in parts 2–4 in the definition of Petri category.

Notice that the case graph $\mathscr{C}[N]$ of a Petri net N is a reflexive Petri submonoid $\mathscr{C}[N] \subseteq \mathscr{T}[N]$ with arrows of the form $t_1 \oplus \cdots \oplus t_n \oplus w$: $u_1 \oplus \cdots \oplus u_n \oplus w \to v_1 \oplus \cdots \oplus v_n \oplus w$. Actually, $\mathscr{C}[N]$ generates $\mathscr{T}[N]$ when closed under $\_;\_$ as shown by the following lemma.

LEMMA 7. *Any* $\alpha: u \to v$ *in* $\mathscr{T}[N]$ *can be decomposed as* $\alpha = \alpha_1; ...; \alpha_n$ *with* $\alpha_i \in \mathscr{C}[N]$.

Proof. We can reason by induction on the depth of α as a term and reduce to the case $\alpha = (\beta \oplus \gamma): u \to u'$. By induction hypothesis, $\beta = \beta_1; ...; \beta_n$ and $\gamma = \gamma_1; ...; \gamma_m$ with $\beta_i: v_i \to v_{i+1}$, $\gamma_i: w_i \to w_{i+1} \in \mathscr{C}[N]$. Either $m < n$ or $m \geqslant n$; say $m < n$. We can express $\gamma = \gamma_1; ...; \gamma_m; \overbrace{w_{m+1}; ...; w_{m+1}}^{n-m}$ by the equation of part 2, and by n applications of part 4 we get $\alpha = \beta \oplus \gamma = (\beta_1 \oplus \gamma_1); \ldots; (\beta_m \oplus \gamma_m); (\beta_{m+1} \oplus w_{m+1}); \ldots; (\beta_n \oplus w_{m+1})$, where all the factors on the right-hand side are in $\mathscr{C}[N]$, as desired. ∎

The above decomposition is not unique. In particular, by further applying parts 4 and 2 to the β_i, γ_i, we obtain the following corollary.

COROLLARY 8. *Any* $\alpha : u \to v$ *in* $\mathscr{T}[N]$ *can be decomposed as* $\alpha = (t_1 \oplus u_1); \ldots; (t_k \oplus u_k)$ *with* $t_i \in T$.

The sequence $t_1, \ldots, t_n$ is then called a *firing sequence* for the computation $\alpha \in \mathscr{T}[N]$. However, the advantage of the category $\mathscr{T}[N]$ is that it provides a calculus with simple algebraic laws for parallel and sequential composition of transitions in the net N and permits focusing on and reasoning about entire computations directly, overcoming the need for indirect, sequentialized, descriptions such as those provided by paths in the case graph or by firing sequences.

We have already considered the full subcategory inclusions:

$$\textbf{\textit{CMonPetri}} \subseteq \textbf{\textit{CMonGraph}}$$

$$\textbf{\textit{CMonRPetri}} \subseteq \textbf{\textit{CMonRGraph}}$$

which justify our claim that, after appropriate closure under increasingly general computations, Petri nets are monoids. This claim also holds true for ***CatPetri***. Indeed, if in the definition of Petri category we relax the condition that the commutative monoid of nodes be free, we obtain the notion of a commutative monoid structure on a category, i.e., a category C and functors $+ : C^2 \to C$, $0 : 1 \to C$ (where 1 is the category with one object and one, identity, morphism) satisfying the commutative monoid equations (expressed as commutative diagrams of functors). Such commutative monoid structures on a category are usually called *strict symmetric monoidal categories*, and the monoid homomorphisms are called *strict monoidal functors* (MacLane, 1971). They determine a category that, to be consistent with the rest of our notation, we shall denote ***CMonCat***. Therefore, we have a full subcategory inclusion,

$$\textbf{\textit{CatPetri}} \subseteq \textbf{\textit{CMonCat}},$$

determined by those strict symmetric monoidal categories whose commutative monoid of objects is free. In ***CMonCat***, as in ***CMonGraph*** and ***CMonRGraph***, finite products and finite coproducts coincide,[2] and this

[2] See the arguments in the proof of Proposition 13 below for a justification of this general fact in terms of semiadditive categories.

property is also inherited by ***CatPetri***. We will, however, give a more detailed justification of this property in what follows.

Let

$$C=(\partial_0, \partial_1 : (T, +, 0) \to S^{\oplus}, \_;\_, id)$$
$$C'=(\partial_0', \partial_1' : (T', +, 0) \to S'^{\oplus}, \_;\_, id)$$

be two Petri categories. We already know that the their product $C \times C'$ as graphs is a reflexive Petri commutative monoid. It has also a Petri category structure by operating componentwise, i.e., $(\alpha, \alpha'); (\beta; \beta') = (\alpha; \beta, \alpha'; \beta')$ and is the categorical product of C and C' in ***CatPetri***. It is also the coproduct, i.e., $C \times C' = C \oplus C'$. To see this, note that we already know this for the underlying reflexive Petri commutative monoid structures; therefore, we only have to check that given two Petri category morphisms $\langle f, g\rangle$: $C \to D$, $\langle f', g'\rangle : C' \to D$ the induced reflexive Petri commutative monoid morphism $\langle [f, f'], [g, g'] \rangle : C \times C' \to D$ is actually a Petri category morphism. Indeed, given $(\alpha, \alpha') : (u, u') \to (v, v')$ and $(\beta, \beta') : (v, v') \to (w, w')$, we have

$$\begin{aligned}
[f, f']((\alpha, \alpha'); (\beta; \beta')) &= [f, f'](\alpha; \beta, \alpha'; \beta') \\
&= (f(\alpha); f(\beta)) + (f'(\alpha'); f'(\beta')) \\
&= (f(\alpha) + f'(\alpha')); (f(\beta) + f'(\beta')) \\
&= [f, f'](\alpha, \alpha'); [f, f'](\beta, \beta').
\end{aligned}$$

Since left adjoints preserve coproducts, given Petri nets N and N' we have

$$\mathscr{T}[N \oplus N'] = \mathscr{T}[N] \oplus \mathscr{T}[N'] = \mathscr{T}[N] \times \mathscr{T}[N'].$$

3. Implementation Morphisms

The sequence of categories that we have been considering provides a corresponding sequence of increasingly more general ways of relating two Petri nets. We can view a net N' as an *implementation* of another net N by giving a morphism $N \to N'$. The widening sequence of morphisms between N and N' that we have been considering is:

- (***Petri***) A transition t in N maps to a transition t' in N'.
- (***Petri$_0$***) A transition t in N maps to a transition t' in N' or is erased.
- (***CMonPetri***) A transition t in N maps to a parallel composition $t_1' \oplus \cdots \oplus t_n'$ of transitions in N' (or is erased, $n = 0$).

• (***CMonRPetri***) A transition t in N maps to a parallel composition $t'_1 \oplus \cdots \oplus t'_n \oplus u$ of transitions in N' and idle (identity) transitions.

• (***CatPetri***) A transition t in N maps to an entire computation α in N' with possibly many sequential and parallel steps.

The most general and interesting case is the last one, since it provides a very flexible way of relating system description at different levels of abstraction. This, admittedly, has been one of the main goals of the theory of Petri nets from its very beginning, but the realization of this goal has proved elusive. We claim that our notions of Petri category and a Petri category morphism give a new and very general solution to the abstraction problem for Petri nets. Notice that these morphisms (and a fortiori the less general ones) obviously preserve the dynamic behavior of nets, since the arrows of a Petri category are computations made up of sequential and parallel composition of atomic transitions, and morphisms preserve sequential and parallel compositions.

To achieve this widening in the ways of relating Petri nets it is *not* necessary to make additional structure (monoid, category, etc.) explicit at all, i.e., we can restrict our attention to ordinary (P/T) Petri nets $N, N' \in \boldsymbol{Petri}$, and for them consider the increasingly general morphisms that our approach provides. This is entirely similar to the notion of a matrix in linear algebra, where a linear function $f\colon \mathscr{R}^n \to \mathscr{R}^m$ is equivalent to a matrix, i.e., a function $M\colon [n] \to \mathscr{R}^m$ ($[n] = \{1, ..., n\}$), and M can be viewed as a "generalized function" or morphism $M\colon [n] \to [m]$ between two finite sets. In our case, the role of $[n]$ and $[m]$ is played by ordinary Petri nets N and N', and the role of the matrix is played by Petri net morphisms such as $N \to N'^{\oplus}$, $N \to \mathscr{T}[N']$, etc., that are then viewed as (generalized) morphisms $N \to N'$ of a category having ordinary Petri nets as objects.

DEFINITION 9. Given two Petri nets $N, N' \in \boldsymbol{Petri}$, an *asynchronous morphism* $\langle f, g\rangle\colon N \to N'$ is just a net morphisms $\langle f, g\rangle\colon N \to N'_0$ in ***Petri***. A composition of two asynchronous morphisms $\langle f, g\rangle\colon N \to N'$, $\langle f', g'\rangle\colon N' \to N''$ is the net morphism $\langle f', g'\rangle^{\$} \circ \langle f, g\rangle\colon N \to N''_0$, where $\langle f', g'\rangle^{\$}\colon N'_0 \to N''_0$ in $\boldsymbol{Petri}_0$ is the unique extension of $\langle f', g'\rangle$. Similarly, we define a *linear morphism* $\langle f, g\rangle\colon N \to N'$ to be a net morphism $\langle f, g\rangle\colon N \to N'^{\oplus}$, a *$\mathscr{C}$-morphism* $\langle f, g\rangle\colon N \to N'$ to be a net morphism $\langle f, g\rangle\colon N \to \mathscr{C}[N'']$, and an *implementation morphism* $\langle f, g\rangle\colon N \to N'$ to be a net morphism $\langle f, g\rangle\colon N \to \mathscr{T}[N']$; composition is always defined as a net morphism $\langle f', g'\rangle^{\$} \circ \langle f, g\rangle$, where $\langle f', g'\rangle^{\$}$ is the unique extension of $\langle f', g'\rangle$ to a morphism in ***CMonPetri***, resp., ***CMonRPetri***, resp., ***CatPetri***. It is an easy fact about adjoint functors that this gives categories

$$\boldsymbol{Petri} \subseteq \boldsymbol{AsynchPetri} \subseteq \boldsymbol{LinPetri} \subseteq \mathscr{C}\boldsymbol{Petri} \subseteq \boldsymbol{ImplPetri}. \tag{2}$$

All with the same objects, i.e., (P/T) Petri nets, but with increasingly more general morphisms.

Notice that implementation morphisms (and a fortiori, the less general ones) obviously preserve the dynamic behavior of nets, since the arrows of a Petri category are computations made up of sequential and parallel compositions of atomic transitions, which are preserved by morphisms. The coproduct is the same in all these categories, namely the coproduct $N \oplus N'$ in ***Petri***. Regarding products, in ***Petri*** it is of course the synchronous product that we have already described; in ***LinPetri***, ***$\mathscr{C}$Petri***, and ***ImplPetri***, since products and coproducts coincide in the corresponding categories ***CMonPetri***, ***CMonRPetri***, and ***CatPetri***, the product is just $N \oplus N'$. For ***AsynchPetri***, the product must be a net N'' such that $N''_0 = N_0 \times N'_0$. Thus, $N'' = (N_0 \times N'_0) - \{(0, 0)\}$ obtained from $N_0 \times N'_0$ by removing the transition $(0, 0)$; it is called the *asynchronous product* of N and N'. Each of the inclusions in (2) is a left adjoint with an associated right adjoint. For example, the right adjoint to the inclusion ***Petri*** $\subseteq$ ***ImplPetri*** maps a net N to (the underlying net of) $\mathscr{T}[N]$.

4. Marked Nets

When considering the behavior of a Petri net it is often convenient to specify an *initial marking*, i.e., an element $u \in S^{\oplus}$. We can then consider generalized transitions $\alpha: u \to v$ in $\mathscr{T}[N]$ starting from the "marking" u. If the marking is made part of the structure of the net, then net morphisms should preserve markings. However, this leads to serious problems with the coproduct construction, since coproducts do not exist for asynchronous morphisms (Winskel, 1987) and a fortiori they do not exist for the more general morphisms considered in this paper. The difficulty can be easily explained as a unification problem. Notice that, for nets with an empty set of transitions, net morphisms are just monoid homomorphisms $S^{\oplus} \to S'^{\oplus}$. Consider markings $u \in S^{\oplus}$, $v \in S'^{\oplus}$, with, say $S \cap S' = \varnothing$. The coproduct as marked nets would require giving $w \in S''^{\oplus}$ together with monoid homomorphisms $j_1: S^{\oplus} \to S''^{\oplus}$, $j_2: S'^{\oplus} \to S''^{\oplus}$ such that $j_1(u) = j_2(v) = w$ with j_1, j_2 universal for this property. This is just an algebraic way of requiring the existence of a most general unifier for the equation $u = v$ in the theory of commutative monoids. It is, however, well known that in the theory of commutative monoids there is a finite set of unifiers generating all other unifiers, but in general there is not a single most general unifier (Herold and Siekmann, 1987).

The solution that Winskel (1987) gave to this problem was to consider the restricted category of "safe" nets, such that the image of the map

$F: (S \times T) + (T \times S) \to \mathcal{N}$ is contained in $\{0, 1\}$ and where multiple tokens can never appear as a consequence of transitions from an initial marking without repeated tokens. Asynchronous morphisms were also substantially restricted to so-called "safe morphisms," and for this category a coproduct was shown to exist. We give a solution that is entirely general, in that it applies to ordinary, pointed, commutative monoid, reflexive commutative monoid, and category Petri nets and morphisms, and permits reasoning about marked Petri nets and forming their coproducts at all those levels. Of course, some restriction has to be imposed, since we already know that coproducts do not exist for arbitrary markings. Our restriction is minimal; we just require that the initial marking is of the form $u = a_1 \oplus \cdots \oplus a_n$ with $a_i \neq a_j$ when $i \neq j$, i.e., we rule out multiple tokens per place in the initial marking; we will later justify why this involves no loss of generality in practice.

DEFINITION 10. A *marked Petri net* is a Petri net $N = (\partial_0, \partial_1 : T \to S^{\oplus})$ together with an element $u \in S^{\oplus}$ of the form $u = a_1 \oplus \cdots \oplus a_n$ with $a_i \neq a_j$ when $i \neq j$. A morphism $\langle f, g \rangle : (N, u) \to (N', u')$ is an ordinary net morphism that, in addition, preserves the markings, i.e., $g(u) = u'$. This defines a category ***MPetri***. Similarly, we can define categories $\boldsymbol{MPetri}_0$, ***MCMonPetri***, ***MCMonRPetri***, ***MCatPetri***, ***MAsynchPetri***, ***MLinPetri***, ***MCPetri***, ***MImplPetri***, just be requiring that the markings be preserved.

For $\mathcal{B}$ any of the categories of (unmarked) nets with structure, there is an obvious forgetful functor $M\mathcal{B} \to \mathcal{B}$ forgetting the marking. This forgetful functor always has a left adjoint that just adds a new element a_0 to S and uses it as the marking. All the categories $M\mathcal{B}$ have products, which are the underlying product in $\mathcal{B}$ with marking $u \oplus u'$ if u and u' were the original markings (where we have taken care of making the places of u and u' disjoint via the isomorphisms $S^{\oplus} \times S'^{\oplus} \simeq (S + S')^{\oplus}$ and where we abuse notation by treating injections into the disjoint union as inclusions).

THEOREM 11. *For* $B =$ ***Petri***, $\boldsymbol{Petri}_0$, ***CMonPetri***, ***CMonRPetri***, ***CatPetri***, ***AsynchPetri***, ***LinPetri***, ***CPetri***, ***ImplPetri***, *the category* $M\mathcal{B}$ *has finite coproducts.*

Proof. We give the construction for $\mathcal{B} =$ ***Petri*** and for $\mathcal{B} =$ ***CatPetri*** and leave the other cases as an exercise. For $\mathcal{B} =$ ***Petri***, let $(N, a_1 \oplus \cdots \oplus a_n)$, $(N', b_1 \oplus \cdots \oplus b_m) \in$ ***MPetri*** with, say, $N = (\partial_0, \partial_1 : T \to S^{\oplus})$ and $N' = (\partial'_0, \partial'_1 : T' \to S'^{\oplus})$. Let $A = \{a_1, ..., a_n\}$ and $B = \{b_1, ..., b_m\}$ and consider the monoid homomorphisms $j_A : A^{\oplus} \to (A \times B)^{\oplus}$, $j_B : B^{\oplus} \to (A \times B)^{\oplus}$ given by $j_A(a_i) = (a_i, b_1) \oplus \cdots \oplus (a_i, b_m)$, $j_B(b_j) = (a_1, b_j) \oplus \cdots \oplus (a_n, b_j)$. We then have $j_A(a_1 \oplus \cdots \oplus a_n) = j_B(b_1 \oplus \cdots \oplus b_m) = \sum_{i,j}(a_i, b_j)$. We can now define

$$(N, a_1 \oplus \cdots \oplus a_n) \oplus (N', b_1 \oplus \cdots \oplus b_m)$$
$$= \left(\partial_0'', \partial_1'' : T + T' \to ((S - A) + (S' - B) + (A \times B))^{\oplus}, \sum_{i,j}(a_i, b_j)\right)$$

with $\partial_i''(t) = h_1(\partial_i(t))$, $\partial_i''(t') = h_2(\partial_i'(t'))$ for $t \in T$, $t' \in T'$, where h_1 is the map

$$(S-A)^{\oplus} \oplus A^{\oplus} \xrightarrow{k^{\oplus} \oplus j_A} ((S-A) + (S'-B))^{\oplus} \oplus (A \times B)^{\oplus},$$

with $k^{\oplus}$ the unique monoid homomorphism extending the injection k of $(S-A)$ into the disjoint union $(S-A)+(S'-B)$, and where h_2 is defined similarly. To check the universal property, the key idea is to remark that ***FinSet***$^{\mathrm{op}}$, the dual of the category of finite sets, is isomorphic to a subcategory of the category of free commutative monoids obtained by sending a function $f: \{b_1, ..., b_m\} \to \{a_1, ..., a_n\}$ to the monoid homomorphism $f^{-1}: (\{a_1, ..., a_n\})^{\oplus} \to (\{b_1, ..., b_m\})^{\oplus}$ defined by $f^{-1}(a_i) = b_{i_1} \oplus \cdots \oplus b_{i_k}$ whenever (set theoreticaly) $f^{-1}(a_i) = \{b_{i_1}, ..., b_{i_k}\}$. By definition, this homomorphism satisfies $f^{-1}(a_1 \oplus \cdots \oplus a_n) = b_1 \oplus \cdots \oplus b_m$. Note that in ***FinSet***$^{\mathrm{op}}$ the coproduct of A and B is $A \times B$.

For $\mathscr{B} =$ ***CatPetri***, let $(C, a_1 \oplus \cdots \oplus a_n), (C', b_1 \oplus \cdots \oplus b_m) \in$ ***MCatPetri***, with, say, $C = (\partial_0, \partial_1 : (T, +, 0) \to S^{\oplus}, \_\circ\_, id))$ and $C' = (\partial_0', \partial_1' : (T', +', 0) \to S'^{\oplus}, \_\circ'\_, id'))$. Their coproduct is constructed as follows: let $(E, \sum_{i,j}(a_i, b_j))$ denote the coproduct in ***MPetri*** of the underlying marked Petri nets, whose construction we have just described. Then $(C, a_1 \oplus \cdots \oplus a_n) \oplus (C', b_1 \oplus \cdots \oplus b_m)$ is obtained as a quotient of $(\mathscr{T}[E], \sum_{i,j}(a_i, b_j))$ by imposing the following relations on $\mathscr{T}[E]$:

1. $\alpha \oplus \beta = \alpha + \beta$, $\alpha, \beta \in T$
2. $\alpha' \oplus \beta' = \alpha' +' \beta'$, $\alpha', \beta' \in T'$
3. $\alpha; \beta = \alpha \circ \beta$, $\alpha, \beta \in T$
4. $\alpha'; \beta' = \alpha' \circ' \beta'$, $\alpha', \beta' \in T'$
5. $id(u) = h_1(u)$, $u \in S^{\oplus}$
6. $id(u') = h_2(u')$, $u' \in S'^{\oplus}$. ∎

In practice, our requirement that the initial marking should have no multiplicities involves no loss in generality. Consider a more general type of marked net (N, u) with an arbitrary marking $u = n_1 a_1 \oplus \cdots \oplus n_k a_k$. We can easily transform such a net into a net $(N^{\bullet}, a_0) \in$ ***MPetri***. If $N = (\partial_0, \partial_1 : T \to S^{\oplus})$ then $N^{\bullet} = (\partial_0^{\bullet}, \partial_1^{\bullet} : T + \{start\} \to (S + \{a_0\})^{\oplus})$ with $\partial_0^{\bullet}, \partial_1^{\bullet}$ identical to ∂_0, ∂_1 on T, with $\partial_0^{\bullet}(start) = a_0$ and $\partial_1^{\bullet}(start) = u$. Except for the initial transition, *start*, the behaviors of (N, u) and $(N^{\cdot}, a_0)$ are identical. This construction is indeed a functor ***GMPetri* → *MPetri*** from the category ***GMPetri*** of marked Petri nets with arbitrary markings to our category ***MPetri***. Similar functors $\boldsymbol{GM\mathscr{B}} \to \boldsymbol{M\mathscr{B}}$ exist for the remaining $\mathscr{B}$. A very nice property of this construction is that nets of the form $(N^{\cdot}, a_0)$ have an initial marking that can never be reached again, i.e., our functor lands inside a full subcategory ***UMPetri*** ⊆ ***MPetri*** of marked nets with unreachable initial markings (in general, $\boldsymbol{UM\mathscr{B}} \subseteq \boldsymbol{M\mathscr{B}}$) for which the coproduct exactly corresponds to the nondeterministic choice operator of languages such as CCS (Milner, 1985).

All the free constructions of the unmarked case carry over to the marked case without a change, i.e., if $\mathscr{B} \to \mathscr{B}'$ is one of the forgetful functors and has, say F, as its left adjoint, the corresponding forgetful functor at the marked level $\boldsymbol{M\mathscr{B}} \to \boldsymbol{M\mathscr{B}'}$ has a left adjoint mapping (X, u) to (FX, u). For example, the left adjoint to the forgetful functor ***MCMonPetri* → *MPetri*** sends (N, u) to $(N^{\oplus}, u)$.

5. Duality and Invariants

It is well known that Petri nets can be dualized by regarding transitions as places, and places as transitions (Petri, 1973). Such duality has many fruitful applications. In this section, we express Petri net duality as a duality functor. We then give a geometrical interpretation of T-invariants and their properties through a very general notion of a *Loop* functor, and we use duality to give a functorial account of S- and T-invariants. Using elementary algebra, we also derive algebraic relations between the groups of S- and T-invariants of a Petri net, and associate to a Petri net N two other groups, $S^{\mathrm{ab}}_{\mathrm{mult}}(N)$ and $T^{\mathrm{ab}}_{\mathrm{mult}}(N)$ that seem to be new.

5.1. *Duality*

Given vector space $\mathscr{R}^3$, its dual space $(\mathscr{R}^3)^*$ is the vector space of all linear functions $f: \mathscr{R}^3 \to \mathscr{R}$ (usually called *linear forms*). As is well known, $(\mathscr{R}^3)^*$ is also a three-dimensional vector space with canonical basis the three projections $x, y, z: \mathscr{R}^3 \to \mathscr{R}$, i.e., $x(a, b, c) = a$, $y(a, b, c) = b$ and $z(a, b, c) = c$. However, if we consider an infinite-dimensional vector space V, its dual space V^* is of strictly greater dimension than V. Given a linear function $h: \mathscr{R}^n \to \mathscr{R}^m$, say with matrix M, h determines a map *in the other direction* for the dual spaces, $h^*: (\mathscr{R}^m)^* \to (\mathscr{R}^n)^*$ mapping each linear form $f: \mathscr{R}^m \to \mathscr{R}$ to the linear form $f \circ h: \mathscr{R}^n \to \mathscr{R}$. If we express $(\mathscr{R}^m)^*$ and $(\mathscr{R}^n)^*$ in terms of their canonical bases of coordinate projections, h^* has a very simple matrix form, namely M^t, the *transpose* of M. Duality therefore means that we can "run the linear function h backwards." This is entirely similar to the case of a binary relation $R: B \to A$ which can also be viewed as relation $R^*: A \to B$.

The notion of a vector space on a field generalizes to the notion of a *module* on a (commutative) ring. For instance, for $\mathscr{Z}$ the ring of integers, a $\mathscr{Z}$-module is just an abelian group, and a $\mathscr{Z}$-linear homomorphism is a group homomorphism. This notion can be further generalized to the notion of a *semimodule*[3] on a semiring, by requiring only that the "vectors" form a commutative monoid and that the $+$ of the coefficients is a commutative monoid. For instance, for $\mathscr{N}$ the semiring of natural numbers, $\mathscr{N}$-semimodules are just commutative monoids, and $\mathscr{N}$-linear homomorphisms are monoid homomorphisms. This permits viewing duality of vector spaces and duality of (finitary) relations as common instances of the general phenomenon of duality for semimodules. The process is always the same: for R the semiring of coefficients, there is a functor $(\_)^*: \boldsymbol{SMod}_{R^{\mathrm{op}}} \to \boldsymbol{SMod}_R$ defined by $V^* = [V \to R]$, where $[V \to R]$ is the R-semimodule of R-linear functions from V to R.

In particular, we have a duality functor $(\_)^*: \boldsymbol{CMon}^{\mathrm{op}} \to \boldsymbol{CMon}$ mapping each commutative monoid M to the commutative monoid $[M \to \mathscr{N}]$ of monoid homomorphisms from M to $\mathscr{N}$. If S and S' are finite, then a monoid homomorphism $f: S^{\oplus} \to S'^{\oplus}$ can be described by an $\mathscr{N}$-matrix M, and the dual $f^*: (S^{\oplus})^* \to (S'^{\oplus})^*$ has M^{t} as its associated matrix, since for S finite we have $S^{\oplus} \simeq (S^{\oplus})^*$.

Consider now the category $\boldsymbol{CMonGraph}$. The underlying category $\boldsymbol{Graph}$ can be viewed as a functor category $\boldsymbol{Set}^J$, where J is the category with two objects 1 and 2, with two identities, and with two arrows $\partial_0, \partial_1: 1 \to 2$. Similarly, we can view $\boldsymbol{CMonGraph}$ as the functor category $\boldsymbol{CMon}^J$. The category J has the remarkable property of being isomorphic to its dual J^{op}. One such isomorphic can be obtained by permuting 1 and 2 and also permuting the ∂_i. This isomorphism combines nicely with the duality of $\boldsymbol{CMon}$ to give a duality functor

$$(\_)^*: \boldsymbol{CMonGraph}^{\mathrm{op}} \to \boldsymbol{CMonGraph},$$

sending $M = (\partial_0, \partial_1: M_1 \to M_2)$ to $M^* = (\partial_1^*, \partial_0^*: M_2^* \to M_1^*)$ and sending $\langle f, g\rangle: M \to M'$ to $\langle f^*, g^*\rangle: M'^* \to M^*$. This duality functor restricts to a functor

$$(\_)^*: \boldsymbol{Petri}_{\mathrm{fin}}^{\oplus} \to \boldsymbol{Petri}_{\mathrm{fin}}^{\oplus},$$

where $\boldsymbol{Petri}_{\mathrm{fin}}^{\oplus}$ is the full subcategory determined by those $M \in \boldsymbol{CMonPetri}$ of the form $M = N^{\oplus}$, with N having finite sets of places and arrows.

[3] In (Meseguer and Sols, 1975), categories of semimodules were proposed as a way of unifying nondeterministic, probabilistic, and "fuzzy" computations. More recently, Main and Benson (1984) have used them in an algebraic treatment of nondeterminism and concurrency.

Given a net N, if we define N^* as the Petri net uniquely determined by the equation $(N^*)^{\oplus} = (N^{\oplus})^*$, it is not hard to see that N^* is obtained from N by exchanging places and transitions and that its source function has as matrix the transposed of ∂_1's matrix and its target function has as matrix the transposed of ∂_0's matrix. For example, in the nets in Fig. 2, the matrices for ∂_0 and ∂_1 of N are

$$M_0 = \begin{bmatrix} 4 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}, \qquad M_1 = \begin{bmatrix} 4 & 0 \\ 1 & 2 \\ 0 & 3 \end{bmatrix},$$

namely, e.g., $t' : a \oplus 2b \to a \oplus 2b$, while the matrices for ∂_0 and ∂_1 of N^* are

$$M_0^* = M_1^t = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 2 & 3 \end{bmatrix}, \qquad M_1^* = M_0^t = \begin{bmatrix} 4 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}.$$

For any $M \in \boldsymbol{Petri}_{\text{fin}}^{\oplus}$ there is a natural isomorphism $M \simeq M^{**}$. For $M \in \boldsymbol{CMonPetri} - \boldsymbol{Petri}_{\text{fin}}^{\oplus}$, we only have $M^{**} \in \boldsymbol{CMonGraph}$, but there is still a natural homomorphism $\varphi_M : M \to M^{**}$.

So far, we have only considered commutative monoid structures on Petri nets. We can take a further step and consider abelian group structures. For example, we can consider the category $\boldsymbol{Ab}^J$ where $\boldsymbol{Ab}$ is the category of abelian groups. We then have an entirely similar duality functor:

$$(\_)^* : \boldsymbol{AbGraph}^{\text{op}} \to \boldsymbol{AbGraph}.$$

The forgetful functor $\boldsymbol{AbGraph} \to \boldsymbol{CMonGraph}$ has a left adjoint

$$(\_)^{\text{ab}} : \boldsymbol{CMonGraph} \to \boldsymbol{AbGraph}.$$

In particular, a commutative monoid Petri net $M = (\partial_0, \partial_1 : M_1 \to S^{\oplus})$ is sent to $M^{\text{ab}} = (\partial_0^{\text{ab}}, \partial_1^{\text{ab}} : M_1^{\text{ab}} \to (S^{\oplus})^{\text{ab}})$ for $(\_)^{\text{ab}} : \boldsymbol{CMon} \to \boldsymbol{Ab}$, the left adjoint of the forgetful functor $\boldsymbol{Ab} \to \boldsymbol{CMon}$. The group $(S^{\oplus})^{\text{ab}}$ is just $\mathscr{Z}\{S\}$, the free abelian group on generators S. We extend this notation to

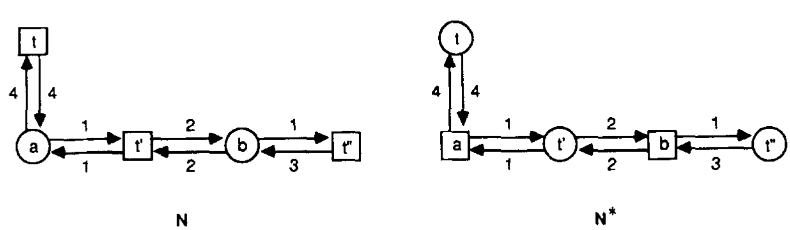

FIG. 2. A net N and its dual N^*.

Petri nets and write $\mathscr{Z}\{N\} = (N^{\oplus})^{ab}$ for $N \in$ ***Petri***. If N has a finite number of transitions and a finite number of places, we again have $\mathscr{Z}\{N\} \simeq \mathscr{Z}\{N\}^{**}$. The advantage of considering abelian groups instead of commutative monoids is that they are easier to work with and have nicer structural properties. For example, a subgroup of a finitely generated group is finitely generated, and free if the original group is so. In general, neither finite generation nor freeness are inherited by the submonoids of a commutative monoid. From the conceptual point of view, however, considering only groups would be unnecessarily restrictive, and indeed there is no need to require that the algebraic structures considered always are groups.

5.2. *Invariants*

The group of T-invariants of a Petri net N is an abelian group naturally associated to the structure $\mathscr{Z}\{N\}$ that we have just introduced. However, there are several possible variations of this notion that can be considered replacing $\mathscr{Z}\{N\}$ by $N^{\oplus}$ (the so-called *positive* T-invariants) or even by $\mathscr{C}[N]$ or $\mathscr{T}[N]$. As we shall see, $\mathscr{T}[N]$ is the most natural choice from an intuitive point of view, but its monoid of invariants has not been explicitly considered before. In all cases, a T-invariant corresponds to a (possible generalized) computation $\alpha: u \to u$ that begins and ends in the same state. Rather than just associating some group or monoid of invariants to a Petri net N, we can take a more intrinsic and geometric viewpoint and define *Loop* functors. For $G = (\partial_0, \partial_1 : T \to V)$ a graph, the subgraph $Loop(G) \subseteq G$ has the same nodes as G and arrows those $t \in T$ such that $\partial_0(t) = \partial_1(t)$, i.e., the arrows of $Loop(G)$ are the equalizer[4] of the pair (∂_0, ∂_1). It follows easily from the equalizer property that *Loop* is indeed an endofunctor Loop: ***Graph*** $\to$ ***Graph***. In fact, *Loop* can be defined as an endofunctor Loop: $\mathscr{B} \to \mathscr{B}$ for any of the categories that we have already considered, i.e., for $\mathscr{B} =$ ***Petri***, ***CMonPetri***, ***CMonRPetri***, ***CatPetri***, and more generally for $\mathscr{B} =$ ***CMonGraph***, ***AbGraph***, ***CMonRGraph***, ***CMon-Cat***.

All such categories $\mathscr{B}$ come equipped with a functor *Arrow* that forgets about the nodes and keeps only the arrows, with whatever structure they had. For example, we have Arrow: ***Petri*** $\to$ ***Set*** mapping $(\partial_0, \partial_1 : T \to S^{\oplus})$ to T, Arrow: ***CMonPetri*** $\to$ ***CMon*** mapping $(\partial_0, \partial_1 : (M, +, 0) \to S^{\oplus})$ to $(M, +, 0)$, and Arrow: ***AbGraph*** $\to$ ***Ab*** mapping $(\partial_0, \partial_1 : (A, +, 0) \to (B, +, 0))$ to $(A, +, 0)$. By definition, for any of the categories $\mathscr{B}$ just listed, a set, monoid, or group of invariants for $X \in \mathscr{B}$ is the object

[4] In any category, given morphisms $f, g: A \to B$ a morphism $j: E \to A$ is called their **equalizer** if $f \circ j = g \circ j$ and for any morphism $h: X \to A$ such that $f \circ h = g \circ h$ there is a unique morphism $\bar{h}: X \to E$ such that $h = j \circ \bar{h}$. In the category of sets, the equalizer of f and g is the set of $x \in A$ such that $f(x) = g(x)$. The dual notion of **coequalizer** is obtained by reversing all the arrows in the above definition.

$T^{\mathscr{B}}_{\text{inv}}(X) = \text{Arrow}(\text{Loop}(X))$. For N a Petri net, the usual *group* of its T-invariants is just $T^{\text{ab}}_{\text{inv}}(N) = \text{Arrow}(\text{Loop}(\mathscr{Z}\{N\}))$. Notice that in the definition of $T^{\text{ab}}_{\text{inv}}(N)$ (differently from $T^{\mathscr{B}}_{\text{inv}}(X)$) the application of the functor $\mathscr{Z}\{N\}$ is included. We shall denote this group $T^{\text{ab}}_{\text{inv}}(N)$, to distinguish it from the monoids $T^{\text{mon}}_{\text{inv}}(N) = \text{Arrow}(\text{Loop}(N^{\oplus}))$, the monoid of *positive* T-invariants, and $T^{\text{cat}}_{\text{inv}}(N) = \text{Arrow}(\text{Loop}(\mathscr{T}[N]))$. $T^{\text{cat}}_{\text{inv}}(N)$ is the most natural of them all, since it consists of all computations $\alpha: u \to u$ of N that begin and end in the same state. The following theorem is related to Theorem 6.7(g) in (Reisig, 1985) and expresses the intuition that if there is a nonidentity computation of N ending in its initial state, then there must also exist a parallel composition of atomic transitions with the same property.

THEOREM 12. *For N a Petri net, $T^{\text{mon}}_{\text{inv}}(N) \neq 0$ iff $T^{\text{cat}}_{\text{inv}}(N) \neq S^{\oplus}$.*

Proof. Of course, since $N^{\oplus} \subseteq \mathscr{T}[N]$, we have $T^{\text{mon}}_{\text{inv}}(N) \subseteq T^{\text{cat}}_{\text{inv}}(N) - S^{\oplus}$. We have to show that if there is an $\alpha: u \to u$ in $T^{\text{cat}}_{\text{inv}}(N)$ with α not in $S^{\oplus}$, then there is an $\alpha' : u' \to u'$ in $T^{\text{mon}}_{\text{inv}}(N)$ with $\alpha' \neq 0$.

By Corollary 8, such an α is of the form $\alpha = (t_1 \oplus u_1); ...; (t_n \oplus u_n)$ with $n \geqslant 1$, say $t_i \oplus u_i: v_i \oplus u_i \to w_i \oplus u_i$ with $w_i \oplus u_i = v_{i+1} \oplus u_{i+1}$. Therefore, $t_1 \oplus \cdots \oplus t_n: v_1 \oplus \cdots \oplus v_n \to w_1 \oplus \cdots \oplus w_n$. If we show $v_1 \oplus \cdots \oplus v_n = w_1 \oplus \cdots \oplus w_n$, the theorem is proved. Indeed, we have $u = v_1 \oplus u_1 = w_n \oplus u_n$ and therefore $(t_1 \oplus u_1) \oplus \cdots \oplus (t_1 \oplus u_1): u \oplus (v_2 \oplus u_2) \oplus \cdots \oplus (v_n \oplus u_n) \to (w_1 \oplus u_1) \oplus \cdots \oplus (w_{n-1} \oplus u_{n-1}) \oplus u$. Since $w_i \oplus u_i = v_{i+1} \oplus u_{i+1}$, the source and the target of this arrow are identical, and since $S^{\oplus}$ is a cancellative monoid,[5] this shows $v_1 \oplus \cdots \oplus v_n = w_1 \oplus \cdots \oplus w_n$, as desired. ∎

PROPOSITION 13. *For $\mathscr{B}$ any of the categories listed above except $\mathscr{B} =$ **Petri**, the functor $T^{\mathscr{B}}_{\text{inv}}$ preserves finite products and coproducts.*[6]

Proof. Notice that for all such $\mathscr{B}$, $\mathscr{B}$-morphisms are of the form $\langle f, g\rangle : X \to Y$, with f, g monoid homomorphisms, and that $\mathscr{B}(X, Y)$ has a commutative monoid structure by defining $\langle f, g\rangle + \langle f', g'\rangle = \langle f+f', g+g'\rangle$. Also, given $\langle h, i\rangle : X' \to X$ and $\langle j, k\rangle : Y' \to Y$ in $\mathscr{B}$, we have

$$\begin{aligned}\langle j, k\rangle \circ \langle f+f', g+g'\rangle &= \langle j \circ (f+f'), k \circ (g+g')\rangle \\ &= (\langle j, k\rangle \circ \langle f, g\rangle) + (\langle j, k\rangle \circ \langle f', g'\rangle)\end{aligned}$$

[5] A commutative monoid is called **cancellative** if $x+y=x+z$ implies $y=z$.

[6] For $S^{\text{ab}}_{\text{inv}}(N)$ (a dual concept to be defined below), this fact was also observed by Winskel (1987).

and, similarly, $\langle f+f', g+g'\rangle \circ \langle h, i\rangle = (\langle f, g\rangle \circ \langle h, i\rangle) + (\langle f' + g'\rangle \circ \langle h, i\rangle)$. Finally, $\langle 0, 0\rangle \circ \langle f, g\rangle = \langle 0, 0\rangle$, and $\langle f, g\rangle \circ \langle 0, 0\rangle = \langle 0, 0\rangle$. This makes all such $\mathscr{B}$ *semiadditive categories* in the sense of Definition 40.1 in (Herrlich and Strecker, 1973). In any semiadditive category, finite products and finite coproducts coincide (Herrlich and Strecker, 1973, Proposition 40.9). Moreover, a functor $F: \mathscr{B} \to \mathscr{B}'$ between two semiadditive categories preserves finite products and coproducts if and only if it is *additive* in the sense that $F_{X,Y}: \mathscr{B}(X, Y) \to \mathscr{B}'(FX, FY)$ is a monoid homomorphism for all $X, Y \in \mathscr{B}$ (Herrlich and Strecker, 1973, Theorem 40.16). It is trivial to check that the functors *Loop* and *Arrow* are additive for all such $\mathscr{B}$, and therefore so is $T^{\mathscr{B}}_{\text{inv}}$ as desired. ∎

As an application of Petri net duality, we consider S-invariants. For $\mathscr{B} =$ ***CMonGraph***, ***AbGraph***, we have duality functors. We then define $S^{\mathscr{B}}_{\text{inv}}(X) = T^{\mathscr{B}}_{\text{inv}}(X^*)$.

In any category $\mathscr{C}$ having equalizers and coequalizers, given a pair of maps $f, g: X \to Y$, its *left exact sequence* is the diagram

$$\mathrm{eq}(f, g) \longrightarrow X \underset{g}{\overset{f}{\rightrightarrows}} Y$$

where $\mathrm{eq}(f, g) \to X$ is the equalizer map. Similarly, its *right exact sequence* is the diagram

$$X \underset{g}{\overset{f}{\rightrightarrows}} Y \longrightarrow \mathrm{coeq}(f, g)$$

for $Y \to \mathrm{coeq}(f, g)$ the coequalizer map. The *exact sequence* of f, g is the diagram

$$\mathrm{eq}(f, g) \longrightarrow X \underset{g}{\overset{f}{\rightrightarrows}} Y \longrightarrow \mathrm{coeq}(f, g).$$

For categories of modules over a commutative ring, or, more generally, for abelian categories, this usually is represented in terms of the difference $f - g$ and leads to the notion of an exact sequence

$$0 \longrightarrow \ker(f-g) \longrightarrow X \xrightarrow{f-g} Y \longrightarrow \mathrm{coker}(f-g) \longrightarrow 0,$$

where $\mathrm{eq}(f, g) = \ker(f-g)$ is the subobject mapped to 0 by $(f-g)$ and $\mathrm{coeq}(f, g)$ is the quotient object $\mathrm{coker}(f-g) = Y/\mathrm{Im}(f-g)$.

However, if R is a semiring but not a ring, as happens for $R = \mathscr{N}$, this latter representation is not possible. In order to relate S-invariants with T-invariants, we shall use a lemma about symmetric monoidal closed categories, a concept that we explain below.

The simplest example of symmetric monoidal closed categories is given by *cartesian closed* categories, i.e., categories $\mathscr{C}$ with finite products and a final object 1 such that for any object X the functor $X \times \_ : \mathscr{C} \to \mathscr{C}$ has a right adjoint $[X \to \_] : \mathscr{C} \to \mathscr{C}$. In other words, there is a natural "lambda abstraction" isomorphism

$$\mathscr{C}(X \times Y, Z) \simeq \mathscr{C}(X, [Y \to Z]).$$

It is not hard to see that letting X vary defines a functor $[\_ \to \_]: \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \to \mathscr{C}$ called the *internal hom* functor, which is related to the ordinary "external" hom by the formula

$$\mathscr{C}(X, Y) \simeq \mathscr{C}(1, [X \to Y]).$$

The simplest cartesian closed category is the category of sets, where the internal and the external homs coincide. More interesting examples are provided by the category of Scott domains used in denotational semantics and by the category ***Cat*** of small categories; we shall see later that ***Graph*** and ***RGraph*** are also cartesian closed. The notion of cartesian closed category can be generalized by dropping the condition that the product is a categorical product. In this way we obtain the notion of a (*symmetric*) *monoidal closed category* (MacLane, 1971), consisting of a category $\mathscr{C}$ together with a product functor $\_ \otimes \_ : \mathscr{C}^2 \to \mathscr{C}$ and a unit object $I \in \mathscr{C}$, together with "unit," "associativity," and "commutativity" natural isomorphisms, making $\mathscr{B}$ into a *symmetric monoidal category*[7] that in addition is *closed* in the sense that for each $X \in \mathscr{C}$ the functor $X \otimes \_$ has a right adjoint $[X \to \_]$. Again, letting X vary we have an internal hom functor and natural isomorphisms

$$\mathscr{C}(X \otimes Y, Z) \simeq \mathscr{C}(X, [Y \to Z])$$

$$\mathscr{C}(X, Y) \simeq \mathscr{C}(I, [X \to Y]).$$

For any commutative semiring R, the category of R-semimodules is closed symmetric monoidal. The internal hom$[A \to B]$ is just the R-semimodule of R-linear functions, and $A \otimes B$ is the *R-tensor product* of the two semimodules (for a detailed and very accessible treatment of tensor products of modules see MacLane and Birkhoff (1967); the case of semimodules is just a slight generalization). Tensor products can be characterized by a universal R-bilinear map $\mu: A \times B \to A \otimes B$ such that for each R-bilinear $f: A \times B \to C$ there is a unique R-linear homomorphism $\bar{f}: A \otimes B \to C$ such

[7] This just generalizes the strict symmetric monoidal categories (i.e., commutative monoid structures in a category) that we have already encountered in our study of Petri nets by relaxing the commutative monoid axioms to hold "up to isomorphism," e.g., commutativity now means $X \otimes Y \simeq Y \otimes X$, etc.

that $f = \bar{f} \circ \mu$. For free semimodules the tensor product has a very easy description, since it is also free and generated by the cartesian product of the generators for the factors. In particular, for $R = \mathcal{N}, \mathcal{Z}$ we have

$$S^{\oplus} \otimes S'^{\oplus} = (S \times S')^{\oplus}$$

$$\mathcal{Z}\{S\} \otimes \mathcal{Z}\{S'\} = \mathcal{Z}\{S \times S'\}.$$

We shall see later how this generalizes to Petri nets. For the moment we just need the following

LEMMA 14. *For $\mathscr{C}$ a symmetric monoidal closed category with internal hom functor $[\_ \to \_]: \mathscr{C}^{\mathrm{op}} \times \mathscr{C} \to \mathscr{C}$ and $B \in \mathscr{C}$ any object, the contravariant functor $[\_ \to B]: \mathscr{C}^{\mathrm{op}} \to \mathscr{C}$ maps any colimit cone in $\mathscr{C}$ to a limit cone in $\mathscr{C}$.*

Proof. Since $\mathscr{C}$ is symmetric monoidal closed, we have $\mathscr{C}(X, [Y \to B]) \simeq \mathscr{C}(X \otimes Y, B) \simeq \mathscr{C}(Y, [X \to B])$ with $X \otimes \_$ left adjoint to $[X \to \_]$. We have to prove that for any $X \in \mathscr{C}$, $\mathscr{C}(X, [\mathrm{colim}\, Y_i \to B]) \simeq \lim \mathscr{C}(X, [Y_i \to B])$. Since left adjoints preserve colimits (MacLane, 1971), we have $\mathscr{C}(X, [\mathrm{colim}\, Y_i \to B]) \simeq \mathscr{C}(X \otimes (\mathrm{colim}\, Y_i), B) \simeq \mathscr{C}(\mathrm{colim}(X \otimes Y_i), B) \simeq \lim \mathscr{C}(X \otimes Y_i, B) \simeq \lim \mathscr{C}(X, [Y_i \to B])$. ∎

COROLLARY 15. *For R any commutative semiring and $X \rightrightarrows Y \to Z$ a right exact sequence of R-semimodules, the dual sequence $Z^* \to Y^* \rightrightarrows Y^*$ obtained by applying the functor $(\_)^* = [\_ \to R]$ is left exact.*

COROLLARY 16. *Let $\mathscr{B} = \boldsymbol{CMonGraph}$ (resp. $\boldsymbol{AbGraph}$), let $X = (\partial_0, \partial_1: X_1 \to X_2)$ be an object in $\mathscr{B}$ and consider the exact sequence in $\boldsymbol{CMon}$ (resp. $\boldsymbol{Ab}$):*

$$T^{\mathscr{B}}_{\mathrm{inv}}(X) \longrightarrow X_1 \underset{\partial_1}{\overset{\partial_0}{\rightrightarrows}} X_2 \longrightarrow \mathrm{coeq}(\partial_0, \partial_1).$$

Then $S^{\mathscr{B}}_{\mathrm{inv}}(X) = \mathrm{coeq}(\partial_0, \partial_1)^*$.

COROLLARY 17. *For any Petri net N with finite sets of places and transitions, there are isomorphisms $T^{\mathrm{mon}}_{\mathrm{inv}}(N) \simeq S^{\mathrm{mon}}_{\mathrm{inv}}(N^*)$ and $T^{\mathrm{ab}}_{\mathrm{inv}}(N) \simeq S^{\mathrm{ab}}_{\mathrm{inv}}(N^*)$.*

Notice that for $\mathscr{B} = \boldsymbol{AbGraph}$, the above sequence yields the exact sequence

$$0 \longrightarrow T^{\mathscr{B}}_{\mathrm{inv}}(X) \longrightarrow X_1 \xrightarrow{\partial = \partial_1 - \partial_0} X_2 \longrightarrow (X_2)/\mathrm{Im}\, \partial \longrightarrow 0 \qquad (3)$$

so that we get the formula $S^{\mathscr{B}}_{\mathrm{inv}}(X) = ((X_2)/\mathrm{Im}\, \partial)^*$. In particular, for $X = \mathcal{Z}\{N\}$ coming form a Petri net $N = (\partial_0, \partial_1: T \to S^{\oplus})$ with a finite set

S of places, the group $\mathscr{Z}\{S\}/\mathrm{Im}\,\partial$ is a finitely generated abelian group and, as it is well known (e.g., MacLane and Birkhoff, 1967) it can be decomposed as $(\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)_{\mathrm{free}} \oplus (\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)_{\mathrm{torsion}}$, where $(\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)_{\mathrm{free}} = \mathscr{Z}\{U\}$ is a free abelian group with a finite set U of generators, and $(\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)_{\mathrm{torsion}} = \mathscr{Z}_{n_1} \oplus \cdots \oplus \mathscr{Z}_{n_k}$ is a direct sum of finite cyclic groups.

COROLLARY 18. *For* $N = (\partial_0, \partial_1 \colon T \to S^{\oplus})$ *a Petri net with S finite,* $S^{\mathrm{ab}}_{\mathrm{inv}}(N) \simeq (\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)_{\mathrm{free}}$.

Proof. $(\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)^* = [\mathscr{Z}\{S\}/\mathrm{Im}\,\partial \to \mathscr{Z}] = [\mathscr{Z}\{U\} \to \mathscr{Z}] \oplus [\mathscr{Z}_{n_1} \to \mathscr{Z}] \oplus \cdots \oplus [\mathscr{Z}_{n_k} \to \mathscr{Z}] = [\mathscr{Z}\{U\} \to \mathscr{Z}] \simeq \mathscr{Z}\{U\}$, since $\mathscr{Z}\{U\}$ is finitely generated and therefore $\mathscr{Z}\{U\}^* \simeq \mathscr{Z}\{U\}$, and since for any $n \in \mathcal{N}$ the only homomorphisms $\mathscr{Z}_n \to \mathscr{Z}$ are the zero ones, i.e., $[\mathscr{Z}_n \to \mathscr{Z}] = 0$.

Consider for example the net in Fig. 2(a). The maps $\partial_0, \partial_1 \colon \mathscr{Z}\{T\} \to \mathscr{Z}\{S\}$ are given by the matrices

$$M_0 = \begin{bmatrix} 4 & 0 \\ 1 & 2 \\ 0 & 1 \end{bmatrix}, \qquad M_1 = \begin{bmatrix} 4 & 0 \\ 1 & 2 \\ 0 & 3 \end{bmatrix}$$

and therefore the map $\partial \colon \mathscr{Z}\{T\} \to \mathscr{Z}\{S\}$ has matrix

$$M = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 2 \end{bmatrix}$$

so that the exact sequence (3) becomes

$$\begin{aligned} 0 &\longrightarrow T^{\mathrm{ab}}_{\mathrm{inv}}(N) \\ &= \mathscr{Z}\{t, t'\} \longrightarrow \mathscr{Z}\{t, t', t''\} \xrightarrow{\partial} \mathscr{Z}\{a, b\} \longrightarrow \mathscr{Z}\{a\} \oplus \mathscr{Z}_2 \longrightarrow 0. \end{aligned}$$

Corollary 16 stated that $S^{\mathrm{ab}}_{\mathrm{inv}}(N) \simeq \mathscr{Z}\{a\}$ and this is clear in the example, since ∂^* has matrix $-M^t$ so that for N^* we have a corresponding exact sequence

$$\begin{aligned} 0 &\longrightarrow S^{\mathrm{ab}}_{\mathrm{inv}}(N) \\ &= \mathscr{Z}\{a\}^* \longrightarrow \mathscr{Z}\{a, b\}^* \xrightarrow{\partial^*} \mathscr{Z}\{t, t', t''\}^* \longrightarrow \mathscr{Z}\{t, t'\}^* \oplus \mathscr{Z}_2 \longrightarrow 0. \end{aligned}$$

The group $(\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)_{\mathrm{torsion}}$ contains additional information about the net N; it measures the multiplicity with which tokens grow due to transitions. In our example, Z_2 indicates that two extra tokens are generated in place

b each time transition t'' fires. We define $T^{\mathrm{ab}}_{\mathrm{mult}}(N) = (\mathscr{Z}\{S\}/\mathrm{Im}\,\partial)_{\mathrm{torsion}}$. This defines an additive functor $T^{\mathrm{ab}}_{\mathrm{mult}}$: $\boldsymbol{Petri}^{\oplus}_{\mathrm{fin}} \to \boldsymbol{Ab}$ which preserves finite products and coproducts. Similarly, for $N^{\oplus} \in \boldsymbol{Petri}^{\oplus}_{\mathrm{fin}}$ we define $S^{\mathrm{ab}}_{\mathrm{mult}}(N) = (\mathscr{Z}\{T\}^*/\mathrm{Im}\,\partial^*)_{\mathrm{torsion}}$. This again gives an additive functor $S^{\mathrm{ab}}_{\mathrm{mult}}$: $(\boldsymbol{Petri}^{\oplus}_{\mathrm{fin}})^{\mathrm{op}} \to \boldsymbol{Ab}$ preserving finite products and coproducts. By duality, we have the following corollary.

COROLLARY 19. *For any Petri net N with finite sets of places and transitions, there are isomorphisms $S^{\mathrm{ab}}_{\mathrm{mult}}(N) \simeq T^{\mathrm{ab}}_{\mathrm{mult}}(N^*)$ and $T^{\mathrm{ab}}_{\mathrm{mult}}(N) \simeq S^{\mathrm{ab}}_{\mathrm{mult}}(N^*)$.*

The monoid $S^{\mathrm{mon}}_{\mathrm{inv}}(N)$ and the groups $S^{\mathrm{ab}}_{\mathrm{inv}}(N)$ and $T^{\mathrm{ab}}_{\mathrm{mult}}(N)$ can be expressed in terms of a construction dual to *Loop*. Notice that, if we define the category of multisets ***Multiset*** with objects functions $\mu: X \to Y$ and morphisms pairs of functions $(f, g): (\mu: X \to Y) \to (\mu': X' \to Y')$ such that $\mu' \circ f = g \circ \mu$, we have an obvious inclusion ***Multiset*** $\hookrightarrow$ ***Graph*** mapping $(\mu: X \to Y)$ to the graph $(\mu, \mu: X \to Y)$. The functor *Loop* is just the right adjoint for that inclusion, but there is also a left adjoint Loop*: ***Graph*** $\to$ ***Multiset*** mapping a graph $(\partial_0, \partial_1: T \to V)$ to the multiset $(q \circ \partial_0 = q \circ \partial_1: T \to N \xrightarrow{q} \mathrm{coeq}(\partial_0, \partial_1))$, obtained by imposing on the nodes N the equivalence relation generated by the pairs $(\partial_0(t), \partial_1(t))$, $t \in T$. *Loop** is similarly defined for ***CMonGraph*** and ***AbGraph*** as a coequalizer construction. For N a Petri net, we can define the monoid of positive S-invariants of N as

$$S^{\mathrm{mon}}_{\mathrm{inv}}(N) = \mathrm{Node}(\mathrm{Loop}^*(N^{\oplus}))^* = \mathrm{Arrow}(\mathrm{Loop}((N^{\oplus})^*))$$

and the group of S-invariants of N as

$$S^{\mathrm{ab}}_{\mathrm{inv}}(N) = \mathrm{Node}(\mathrm{Loop}^*(\mathscr{Z}\{N\}))^* = \mathrm{Arrow}(\mathrm{Loop}(\mathscr{Z}\{N\}^*)).$$

For N a finite Petri net, we can define

$$T^{\mathrm{ab}}_{\mathrm{mult}}(N) = \mathrm{Node}(\mathrm{Loop}^*(\mathscr{Z}\{N\}))_{\mathrm{torsion}},$$

i.e., as the torsion subgroup of the group Node(Loop*($\mathscr{Z}\{N\}$)). For yet another description of invariants, see the footnote in Section 6.3.

6. TENSOR PRODUCTS AND FUNCTION SPACES

Commutative monoids can be viewed as semimodules on the semiring of natural numbers $\mathscr{N}$, just as abelian groups can be viewed as $\mathscr{Z}$-modules on the ring of integers $\mathscr{Z}$. In this way, they provide the most basic instance of linear and multilinear algebra. We have already seen that the categories

$$\begin{array}{ccc} Fa & \xrightarrow{Ff} & Fa' \\ \varphi_a \downarrow & & \downarrow \varphi_b \\ Ga & \xrightarrow{Gf} & Ga' \end{array}$$

FIG. 3. The commutative diagram for natural transformations.

CMon and ***Ab*** (or, more generally, any category of semimodules on a commutative semiring) are closed symmetric monoidal. In this section we shall see that this generalizes to ***CMonGraph***, ***CMonRGraph***, and ***CMonCat***, basically because ***Graph***, ***RGraph***, and ***Cat*** are cartesian closed, and in addition ***Graph*** and ***RGraph*** are topoi (Lawvere, 1971). These properties are, to a good extent, inherited by the subcategories ***CMonPetri***, ***CMonRPetri***, and ***CatPetri***, although the internal hom objects may at times be outside such subcategories.

The categories ***Petri*** and $\boldsymbol{Petri}_0$ also have an associated symmetric monoidal closed structure that we describe in detail below.

6.1. *Cartesian Closed Structure of Graphs and Categories*

This subsection recalls the well-known fact that graphs, reflexive graphs and categories form cartesian closed categories. This will be important in understanding the monoidal closed structure of Petri nets, Petri commutative monoids (reflexive or not) and Petri categories.

The fact that (small) categories are cartesian closed is familiar to anybody acquainted with natural transformations. Given (small) categories A and B, the category B^A has objects functors $F: A \to B$. Morphisms $\varphi: A \to B$ between two such functors are *natural transformations*, i.e., families $\{\varphi_a: Fa \to Ga \mid a \in |C|\}$ such that for each $f: a \to a' \in A$ the diagram in Fig. 3 commutes. We then have an isomorphism

$$C^{A \times B} \simeq C^{B^A}$$

natural in $A, B, C \in \boldsymbol{Cat}$, i.e., ***Cat*** is cartesian closed (MacLane, 1971).

For any small category J, the category $\boldsymbol{Set}^J$ is a topos (Lawvere, 1971) and therefore cartesian closed. In particular, the categories ***Graph*** and ***RGraph*** are topoi,[8] since $\boldsymbol{Graph} = \boldsymbol{Set}^J$, where J is the category with two objects 1, 2, their identities $1_1, 1_2$ and two morphisms $\partial_0, \partial_1: 1 \to 2$. Similarly, $\boldsymbol{RGraph} = \boldsymbol{Set}^K$, for K the category obtained by adding to J a morphism $\mathrm{id}: 2 \to 1$ and the equation $\partial_0 \circ id = \partial_1 \circ id = 1_2$. We have already discussed the straightforward construction of products in ***Graph*** and

[8] For a beautiful treatment of the topos structure of graphs see the recent work by Lawvere (1989).

RGraph. Given any small category C, and objects $X, Y \in \boldsymbol{Set}^C$, the internal hom object $[X \to Y] \in \boldsymbol{Set}^C$ is always given by the Yoneda formula

$$[X \to Y](c) \simeq \mathrm{nat}(C(c, \_), [X \to Y]) \simeq \mathrm{nat}(C(c, \_) \times X, Y)$$

which in our case can be specialized for $C = J, K$. We shall presently explain the meaning of this formula in ***Graph*** and ***RGraph***. Given graphs $G = (\partial_0, \partial_1 : T \to V)$ and $G' = (\partial'_0, \partial'_1 : T' \to V')$, the graph $[G \to G']$ has as set of arrows the set

$$\{(h: T \to T', g: V \to V', g': V \to V') \mid \partial'_0 \circ h = g \circ \partial_0, \partial'_1 \circ h = g' \circ \partial_1\}$$

and as set of nodes the set V'^V of functions from V to V'. The source and target maps are the second and third projections, i.e., $\partial_0(h, g, g') = g$, $\partial_1(h, g, g') = g'$. We can illustrate this with an example. Let $G = (\lambda x.(2x), \lambda x.(2x+1): \mathcal{N} \to \mathcal{N})$, i.e., the graph with set of nodes the natural numbers and with exactly one arrow $n: 2n \to 2n+1$ for each $n \in \mathcal{N}$. Let G' be an arbitrary graph. Then the graph $[G \to G']$ has as nodes sequences $g: \mathcal{N} \to V'$ of nodes in G'. An arrow $g \to g'$ between two such sequences is a sequence $h: \mathcal{N} \to T'$ of arrows $h_n: g_{2n} \to g'_{2n+1}$ (see Fig. 4).

The external hom ***Graph***(G, G') is obtained by considering the graph homomorphisms $1 \to [G \to G']$, where $1 = (1_{[1]}, 1_{[1]}: [1] \to [1])$, for $[1]$ the one point set $\{1\}$, is the final object of ***Graph***, i.e., we have

$$\boldsymbol{Graph}(G, G') \simeq \boldsymbol{Graph}(1, [G \to G']).$$

Since we will be considering internal homs in many different categories, we will adopt a uniform convention of qualifying the functor $[\_ \to \_]$ with a subscript suggesting its category of definition. Thus we will sometimes write $[A \to B]_C$ for the internal hom B^A in ***Cat***, $[G \to G']_G$ for the internal hom in ***Graph*** and $[G \to G']_{RG}$ for the internal hom in ***RGraph*** that we shall describe below.

$$\begin{array}{c} g_0 \xrightarrow{h_0} g'_1 \\ g_2 \xrightarrow{h_1} g'_3 \\ g_4 \xrightarrow{h_2} g'_5 \\ g_6 \xrightarrow{h_3} g'_7 \\ \vdots \end{array}$$

FIG. 4. An arrow of the graph $[G \to G']$, where $G = (\lambda x.(2x), \lambda x.(2x+1): \mathcal{N} \to \mathcal{N})$ and G' is an arbitrary graph.

Given reflexive graphs $G = (\partial_0, \partial_1 : T \to V, id)$, $G' = (\partial'_0, \partial'_1 : T' \to V', id')$, the internal hom $[G \to G']_{RG}$ has as set of arrows the set of tuples

$$\{(h : T \to T', f : T \to T', f' : T \to T', g : V \to V', g' : V \to V') \mid \langle f, g\rangle, \langle f', g'\rangle : G \to G' \text{ are reflexive graph homomorphisms and } \partial'_0 \circ h = g \circ \partial_0,\ \partial'_1 \circ h = g' \circ \partial_1\},$$

its set of nodes is the set $\boldsymbol{RGraph}(G, G')$ of reflexive graph homomorphisms, and we define

$$\partial_0(h, f, f', g, g') = \langle f, g\rangle$$
$$\partial_1(h, f, f', g, g') = \langle f', g'\rangle$$
$$id\langle f, g\rangle = (f, f, f, g, g).$$

Therefore, nodes in $[G \to G']_{RG}$ are reflexive graph homomorphisms and an arrow $\langle f, g\rangle \to \langle f', g'\rangle$ is a way of systematically relating them by a function $h : T \to T'$ as shown in the diagram in Fig. 5. Again, we have $\boldsymbol{RGraph}(G, G') \simeq \boldsymbol{RGraph}(1, [G \to G']_{RG})$, for 1 the final object $(1_{[1]}, 1_{[1]} : [1] \to [1], id = 1_{[1]})$.

Consider the sequence of forgetful functors

$$\boldsymbol{Cat} \to \boldsymbol{RGraph} \to \boldsymbol{Graph}$$

which allows us to regard a category as a reflexive graph, or just as a graph, and to regard a reflexive graph as a graph. In particular, for A and B categories, we have the following internal homs: $[A \to B]_C$, $[A \to B]_{RG}$ and $[A \to B]_G$, and for G and G' reflexive graphs we have the internal

$$a \xrightarrow{t} b$$

G

$$\begin{tikzcd} ga \arrow[r, "ft"] \arrow[dr, "ht"] & gb \\ g'a \arrow[r, "f't"] & g'b \end{tikzcd}$$

G'

FIG. 5. Evaluation at $t \in T$ of an arrow $\langle f, g\rangle \to \langle f', g'\rangle$ of graph $[G \to G']_{RG}$.

homs $[G \to G']_{RG}$ and $[G \to G']_G$. How are all these homs related? In the case of external homs, we have

$$\mathbf{Cat}(A, B) \subseteq \mathbf{RGraph}(A, B) \subseteq \mathbf{Graph}(A, B) \tag{4}$$

$$\mathbf{RGraph}(G, G') \subseteq \mathbf{Graph}(G, G'). \tag{5}$$

But for internal homs the situation is more subtle. Basically, the internal homs have richer structure and contain more information as we move up from graphs to reflexive graphs and to categories. In order to relate these different homs, we have to "throw away" the extra information of the richer structure. This takes the form of natural transformations called *comparison maps*:[9]

$$\begin{aligned} \rho &: [A \to B]_C \to [A \to B]_{RG} \\ \rho' &: [G \to G']_{RG} \to [G \to G']_G \\ \rho'' &: [A \to B]_C \to [A \to B]_G, \end{aligned}$$

where ρ'' is obtained by composing ρ and ρ'. We presently describe ρ and ρ'. ρ maps a natural transformation $\varphi: \langle f, g \rangle \to \langle f', g' \rangle$ to the arrow

$$(\lambda\alpha \in \text{Arrows}(A) \cdot (\varphi_{\partial_1(\alpha)} \circ f\alpha): |A| \to |B|, f, f', g, g'): \langle f, g \rangle \to \langle f', g' \rangle$$

in $[A \to B]_{RG}$; i.e., we extract from the natural transformation φ the diagonals $\varphi_{a'} \circ f\alpha = f'\alpha \circ \varphi_a$, for each $\alpha: a \to a'$ in A. The map ρ' maps an arrow $(h, f, f', g, g'): \langle f, g \rangle \to \langle f', g' \rangle$ in $[G \to G']_{RG}$ to the arrow $(h, g, g'): g \to g'$ in $[G \to G']_G$. Denoting by $[A \to B]_C^{\rho}$, $[G \to G']_{RG}^{\rho'}$, and $[A \to B]_C^{\rho''}$ the images of the comparison maps ρ, ρ', and ρ'', we then have the internal versions

$$\begin{aligned} &[A \to B]_C^{\rho} \subseteq [A \to B]_{RG} \\ &[A \to B]_C^{\rho''} \subseteq [A \to B]_{RG}^{\rho'} \subseteq [A \to B]_G \\ &[G \to G']_{RG}^{\rho'} \subseteq [G \to G']_G \end{aligned}$$

of the external homset inclusions (4) and (5).

6.2. *Monoidal Closed Structure of Petri and Petri$_0$*

We can slightly generalize the categories ***Petri*** and ***Petri$_0$*** by dropping the requirement that the nodes are a free commutative monoid and just requiring that the nodes have a commutative monoid structure $M =$

[9] In general, for $F: \mathscr{C} \to \mathscr{C}'$ a product-preserving functor between two Cartesian closed categories, a *comparison map* $\rho: F([A \to B]_{\mathscr{C}}) \to [FA \to FB]_{\mathscr{C}'}$ can always be obtained by "currying" the map $F(\varepsilon: [A \to B]_{\mathscr{C}} \times A \to B)$, where ε is the evaluation map in $\mathscr{C}$.

$(M, +, 0)$. In this way we obtain categories ***GralPetri*** and ***GralPetri***$_0$ that contain ***Petri*** and ***Petri***$_0$ as full subcategories. We shall see below that ***GralPetri*** and ***GralPetri***$_0$ are closed symmetric monoidal. In particular, there is a tensor product that restricts to ordinary Petri nets, and (after imposing a finiteness condition) given two Petri nets there is a third Petri net that is their internal hom. All these constructions seem to be new, as well as the monoidal closed structure of ***Petri*** that seems not to have been recognized before.

THEOREM 20. *GralPetri and GralPetri$_0$ are (symmetric) monoidal closed categories.*

Proof. We consider first the case of the category ***GralPetri***. Let $N = (\partial_0, \partial_1 : T \to M)$ and $N' = (\partial_0', \partial_1' : T' \to M')$ be generalized Petri nets (with M and M' commutative monoids). We define their tensor product $N \otimes N'$ as the generalized Petri net $N \otimes N' = (\partial_0'', \partial_1'' : T \times T' \to M \otimes M')$, where $M \otimes M'$ is the tensor product of the monoids M, M' and ∂_i'' is the composition

$$T \times T' \xrightarrow{\partial_i \times \partial_i'} M \times M' \xrightarrow{\alpha} M \otimes M',$$

where $\alpha : M \times M' \to M \otimes M'$ is the universal bilinear map for $M \otimes M'$.

Since the tensor product of two free commutative monoids $S^{\oplus}$ and $S'^{\oplus}$ is the commutative monoid $(S \times S')^{\oplus}$, the tensor product for generalized Petri nets restricts to one for ordinary Petri nets, $\_ \otimes \_ : \boldsymbol{Petri}^2 \to \boldsymbol{Petri}$. The tensor product $N \otimes N'$ of two Petri nets N and N' has as transitions the cartesian product of their transitions, as places the cartesian product of their places and as multiplicities the product of their multiplicities (see the example in Fig. 6). The unit object I is the Petri net $(\partial_0, \partial_1 : [1] \to \mathcal{N})$, with $\partial_0 = \partial_1$ the inclusion of $[1]$ in $\mathcal{N}$.

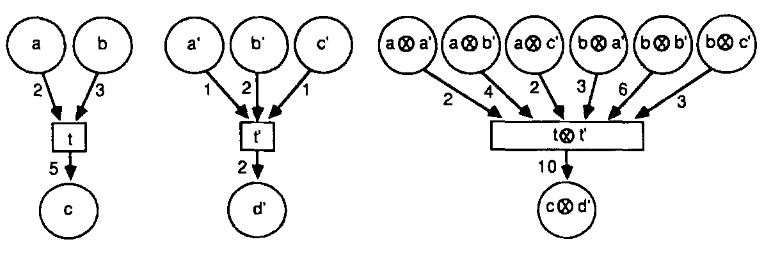

FIG. 6. Two nets N and N' and their tensor product $N \otimes N'$.

The internal hom$[N \to N']_P$ of our two generalized Petri nets is a subgraph of the graph$[N \to N']_G$ and has as arrows the set of triples

$$\{(h: T \to T', g: M \to M', g': M \to M') \mid g, g' \text{ are monoid homomorphisms and } \partial'_0 \circ h = g \circ \partial_0 \text{ and } \partial'_1 \circ h = g' \circ \partial_1\}$$

and as monoid of nodes the monoid $[M \to M']$ of monoid homomorphisms from M to M', with ∂_0 and ∂_1 the second and third projections. We then have

$$\textbf{\textit{GralPetri}}(N, N') \simeq \textbf{\textit{GralPetri}}(I, [N \to N']_P)$$

and we leave for the reader to check the natural isomorphism:

$$\textbf{\textit{GralPetri}}(N \otimes N', N'') \simeq \textbf{\textit{GralPetri}}(N, [N' \to N'']_P).$$

The symmetric monoidal closed structure of $\textbf{\textit{GralPetri}}_0$ can be easily described by remarking that every generalized pointed Petri net is isomorphic to one of the form N_0, for N a generalized Petri net, where $(\_)_0: \textbf{\textit{GralPetri}} \to \textbf{\textit{GralPetri}}_0$ is the left adjoint to the forgetful functor $\textbf{\textit{GralPetri}}_0 \to \textbf{\textit{GralPetri}}$ that adds a transition $0: 0 \to 0$ to the net $\mathbf{N}$. We can then define the tensor product in $\textbf{\textit{GralPetri}}_0$ by

$$N_0 \otimes N'_0 = (N \otimes N')_0,$$

where $\otimes$ in the right-hand side is performed in $\textbf{\textit{GralPetri}}$. This restricts to a functor $\_ \otimes \_: \textbf{\textit{Petri}}_0^2 \to \textbf{\textit{Petri}}_0$. The unit of the tensor product is the pointed Petri net I_0. We finally define the internal hom by

$$[N_0 \to N'_0]_{P_0} = [N \to N'_0]_P$$

which is pointed with point $(0: T \to T' + \{0\}, 0: M \to M', 0: M \to M')$. This works, since $\textbf{\textit{GralPetri}}_0(N_0 \otimes N'_0, N''_0) \simeq \textbf{\textit{GralPetri}}(N \otimes N', N''_0) \simeq \textbf{\textit{GralPetri}}(N, [N' \to N''_0]_P) \simeq \textbf{\textit{GralPetri}}(N, [N'_0 \to N''_0]_{P_0}) \simeq \textbf{\textit{GralPetri}}_0(N_0, [N'_0 \to N''_0]_{P_0})$. ∎

Notice that, whenever $S = \{a_1, ..., a_n\}$ is finite, we have

$$[S^{\oplus} \to S'^{\oplus}] \simeq \overbrace{S'^{\oplus} \times \cdots \times S'^{\oplus}}^{n} \simeq \overbrace{S'^{\oplus} \oplus \cdots \oplus S'^{\oplus}}^{n} \simeq \overbrace{(S' + \cdots + S')}^{n}{}^{\oplus}.$$

Therefore we have the following corollary.

COROLLARY 21. *For $N, N' \in$ **Petri** (resp. $N, N' \in \textbf{\textit{Petri}}_0$) and N with a finite set of places, $[N \to N']_P \in$ **Petri** (resp. $[N \to N']_{P_0} \in \textbf{\textit{Petri}}_0$).*

Furthermore, defining $\boldsymbol{Petri}_{S\text{-fin}}$ and $\boldsymbol{Petri}_{0S\text{-fin}}$ as the categories of Petri nets and pointed Petri nets with finite sets of places, we have also the following corollary.

COROLLARY 22. *$\boldsymbol{Petri}_{S\text{-fin}}$ and $\boldsymbol{Petri}_{0S\text{-fin}}$ are symmetric monoidal closed categories.*

6.3. *Monoidal Closed Structure of **CMonPetri**, **CMonRPetri**, **CatPetri***

The categories we should concentrate on are ***CMonGraph***, ***CMonRGraph***, and ***CMonCat***. Since topoi are categories of generalized sets, most standard mathematical constructions carry over to a topos with a natural numbers object. In particular, the constructions establishing that ***CMon*** is symmetric monoidal closed could be carried over for commutative monoids over a topos $\mathcal{S}$ with a natural numbers object and specialized for $\mathcal{S} = \boldsymbol{Graph}$ and $\mathcal{S} = \boldsymbol{RGraph}$ to ***CMonGraph*** and ***CMonRGraph***. However, it is not difficult to give a more direct description of their symmetric monoidal closed structure. Tensor products are constructed pointwise, i.e., for $M = (\partial_0, \partial_1 : M_1 \to M_2)$ and $M' = (\partial'_0, \partial'_1 : M'_1 \to M'_2)$ in ***CMonGraph***, we define $M \otimes M' = (\partial_0 \otimes \partial'_0, \partial_1 \otimes \partial'_1 : M_1 \otimes M'_1 \to M_2 \otimes M'_2)$, and for $M = (\partial_0, \partial_1 : M_1 \to M_2, id)$ and $M' = (\partial'_0, \partial'_1 : M'_1 \to M'_2, id')$ in ***CMonRGraph***, we define $M \otimes M' = (\partial_0 \otimes \partial'_0, \partial_1 \otimes \partial'_1 : M_1 \otimes M'_1 \to M_2 \otimes M'_2, id \otimes id')$. The unit[10] is $I = (1_{\mathcal{N}}, 1_{\mathcal{N}} : \mathcal{N} \to \mathcal{N})$ in ***CMonGraph*** and $I = (1_{\mathcal{N}}, 1_{\mathcal{N}} : \mathcal{N} \to \mathcal{N}, id = 1_{\mathcal{N}})$ in ***CMonRGraph***. For ***CMonGraph***, the internal hom $[M \to M']_{CMG}$ is a subgraph of $[M \to M']_G$ with arrows those $(h, g, g') : g \to g'$ in $[M \to M']_G$ such that h, g, and g' are monoid homomorphisms; such arrows form a commutative monoid by componentwise addition. The monoid of nodes is the monoid $[M_1 \to M_2]$, and ∂_0 and ∂_1 are second and third projection. Similarly, for M, $M' \in \boldsymbol{CMonRGraph}$, the internal hom $[M \to M']_{CMR}$ is a reflexive subgraph of $[M \to M']_{RG}$ with monoid of arrows given by those $(h, f, f', g, g') : \langle f, g \rangle \to \langle f', g' \rangle$ in $[M \to M']_{RG}$ such that h, f, f', g and g' are all monoid homomorphisms; and monoid of nodes the external homset $\boldsymbol{CMonRGraph}(M, M')$ which is a commutative monoid by componentwise addition.

We remark that the tensor products restrict to the full subcategories $\boldsymbol{CMonPetri} \subseteq \boldsymbol{CMonGraph}$ and $\boldsymbol{CMonRPetri} \subseteq \boldsymbol{CMonRGraph}$ and that the unit objects belong to them. In addition, we have the following lemma and corollary.

[10] Notice that the external hom of ***CMonGraph*** has a commutative monoid structure, so that for each $X \in \boldsymbol{CMonGraph}$ we have a functor $\boldsymbol{CMonGraph}(X, \_) : \boldsymbol{CMonGraph} \to \boldsymbol{CMon}$. The monoid of invariants $T^{\text{mon}}_{\text{inv}}(N)$ of a Petri net N then has a very simple description in terms of the unit $I \in \boldsymbol{CMonGraph}$, namely $T^{\text{mon}}_{\text{inv}}(N) = \boldsymbol{CMonGraph}(I, N^{\oplus})$.

LEMMA 23. *For $M, M' \in$ **CMonPetri** and M such that its set S of places is finite, $[M \to M']_{CMG} \in$ **CMonPetri**.*

COROLLARY 24. *The full subcategory **CMonPetri**$_{S\text{-fin}}$ determined by those Petri commutative monoids with finite sets of places, is symmetric monoidal closed.*

For $M, M' \in$ ***CMonRPetri***, the internal hom $[M \to M']_{CMR}$ need not be in ***CMonRPetri***, even if M has a finite set of places; however, there is an inclusion

$$[M \to M']^{\rho'}_{CMR} \subseteq [M \to M']_{CMG}.$$

If M has a finite set of places, $[M \to M']_{CMG} \in$ ***CMonPetri***, but $[M \to M']^{\rho'}_{CMR}$ need not belong to ***CMonPetri***, since a submonoid of a free commutative monoid need not be free.

We must discuss the category ***CMonCat*** of strict monoidal categories. Given $C, D \in$ ***CMonCat***, we define $[C \to D]_{CMC}$ as the category with objects strict monoidal functors and morphisms natural transformations. As mentioned before, the category ***CatPetri*** is the full subcategory determined by those $C \in$ ***CMonCat*** whose commutative monoid of objects is free. In this context, for consistency with the rest of the paper, we will use additive notation for strict monoidal products for a $C \in$ ***CMonCat***.

LEMMA 25. *$[C \to D]_{CMC}$ can be made into a strict symmetric monoidal category.*

Proof. Addition on the objects is defined by $\langle f, g\rangle + \langle f', g'\rangle = \langle f + f', g + g'\rangle$ and determines a commutative monoid structure. Addition on natural transformations $\varphi: \langle f, g\rangle \to \langle f', g'\rangle$ and φ': $\langle h, i\rangle \to \langle h', i'\rangle$ is the natural transformation $\varphi + \varphi': \langle f + h, g + i\rangle \to \langle f' + h', g' + i'\rangle$ determined by the family $\{\varphi_x + \varphi'_x: g(x) + i(x) \to g'(x) + i'(x) \mid x \in |C|\}$. Such a family is natural, since for $\alpha: x \to y$ in C we have, writing things in diagrammatic order,

$$\begin{aligned}(f+h)(\alpha); (\varphi_y + \varphi'_y) &= (f\alpha + h\alpha); (\varphi_y + \varphi'_y) = (f\alpha; \varphi_y) + (h\alpha; \varphi'_y)\\ &= (\varphi_x; f'\alpha) + (\varphi'_x; h'\alpha) = (f' + h')(\alpha); (\varphi_x + \varphi'_x).\end{aligned}$$

It follows from our definition of $\varphi + \varphi'$ that ∂_0, ∂_1, and *id* are monoid homomorphisms. The only condition left to check is the equation $(\varphi + \varphi'); (\psi + \psi') = (\varphi; \psi) + (\varphi'; \psi')$ that we leave as an exercise. ∎

The tensor product is determined by a universal property of bilinearity. Given $C, D \in$ ***CMonCat***, we have to exhibit a category $C \otimes D \in$ ***CMonCat*** and a bilinear functor $\langle f, g \rangle : C \times D \to C \otimes D$, i.e., a functor $\langle f, g \rangle$ such that for any $\alpha, \alpha' : x \to y$ in C and $\beta, \beta' : z \to w$ in D we have

1. $f(\alpha, 0) = f(0, \beta) = 0$
2. $f(\alpha + \alpha', \beta) = f(\alpha, \beta) + f(\alpha', \beta)$
3. $f(\alpha, \beta + \beta') = f(\alpha, \beta) + f(\alpha, \beta')$.

Notice that the forgetful functor ***CMonCat*** $\to$ ***CMonRGraph*** has a left adjoint P: ***CMonRGraph*** $\to$ ***CMonCat*** which is essentially a path category construction. We can easily construct $C \otimes D$ in terms of the tensor product of C and D in ***CMonRGraph*** that we shall denote by $C \otimes_R D$. We have a bilinear morphism $C \times D \to C \otimes_R D$ for the underlying commutative monoid on a reflexive graph structure. We can then compose with the unit map $C \otimes_R D \to P(C \otimes_R D)$ to get another bilinear morphism for the ***CMonRGraph*** structure. Denote the composite blinear map

$$C \times D \to C \otimes_R D \to P(C \otimes_R D)$$

by $\langle f, g \rangle$. To make it into a functor, we have to further impose on $P(C \otimes_R D)$ relations Γ of the form

$$f(\alpha; \alpha', \beta; \beta') = f(\alpha, \beta); f(\alpha', \beta')$$

and then we define $C \otimes D \in$ ***CMonCat*** as the quotient $P(C \otimes_R D)/\Gamma$. In this way we get a bilinear functor as a composition

$$\langle \bar{f}, \bar{g} \rangle : C \times D \to C \otimes_R D \to P(C \otimes_R D) \to C \otimes D.$$

Such a functor has the desired universal property, since, given a bilinear functor $\langle h, i \rangle : C \times D \to E$, it induces a unique $(h^{\bullet}, i^{\bullet}) : C \otimes_R D \to E$ in ***CMonRGraph***, which in turn induces a unique $(h^{\dagger}, i^{\dagger}) : P(C \otimes_R D) \to E$ in ***CMonCat***. We then have $h^{\dagger}(f(\alpha; \alpha', \beta; \beta')) = h(\alpha; \alpha', \beta; \beta') = h(\alpha; \beta)$; $h(\alpha', \beta') = h^{\dagger}(f(\alpha, \beta))$; $h^{\dagger}(f(\alpha', \beta')) = h^{\dagger}(f(\alpha, \beta); f(\alpha', \beta'))$, i.e. $(h^{\dagger}, i^{\dagger})$ induces a unique $(h^{\ddagger}, i^{\ddagger}) : C \otimes D \to E$ in ***CMonCat***, as desired. The unit for the tensor product is the category with objects $\mathcal{N}$ and just one identity morphism $n : n \to n$ for each n, with monoidal product $n + m : n + m \to n + m$. This ends our discussion of the symmetric monoidal closed structure for ***CMonCat***.

We finish the discussion of the different internal homs in this section by listing, for $C, D \in$ ***CMonCat***, $M_1, M_2 \in$ ***CMonRGraph***, and $M, M' \in$ ***CMonGraph***, the following inclusions:

$$[C \to D]_{CMC} \subseteq [C \to D]_C$$

$$[C \to D]^{\rho}_{CMC} \subseteq [C \to D]_{CMR} \subseteq [C \to D]_{RG}$$

$$[C \to D]^{\rho''}_{CMC} \subseteq [C \to D]^{\rho'}_{CMR} \subseteq [C \to D]_G$$

$$[M_1 \to M_2]_{CMR} \subseteq [M_1 \to M_2]_{RG}$$

$$[M_1 \to M_2]^{\rho'}_{CMR} \subseteq [M_1 \to M_2]_{CMG} \subseteq [M_1 \to M_2]_G$$

$$[M \to M']_{CMG} \subseteq [M \to M']_P \subseteq [M \to M']_G.$$

7. Generalizations

One of the advantages of adopting a categorical point of view when investigating a problem is that often new connections are naturally discovered and results can be easily transferred by relying on the common categorical properties. In this section, we take a more abstract view of the developments in previous sections. A common pattern emerges that naturally suggests a wide variety of ways in which the ideas we have presented can be generalized. In what follows, we will find very useful to present our ideas using the concept of a *monad*, as explained below.

7.1. *Monads*

Let $T = (\Sigma, \Gamma)$ be a presentation by operations Σ and equations Γ of a (one sorted) algebraic theory, such as commutative monoids, rings, etc. Any set X generates a *free* T-algebra $T(X) = T_{\Sigma,\Gamma}(X)$ and there is a natural map $\eta_X : X \to T(X)$ interpreting the generators inside the algebra. This natural map is a natural transformation $\eta : 1_{\mathbf{Set}} \to T$ between two endofunctors. There is also a natural transformation $\mu : T^2 \to T$. For $T = (\Sigma, \varnothing)$, $\mu : T_\Sigma(T_\Sigma(X)) \to T_\Sigma(X)$ maps a term $t[t_1, ..., t_n]$ (where the $t_1, ..., t_n \in T_\Sigma(X)$ are viewed as variables without any internal structure) to the substitution term $t(t_1, ..., t_n) \in T_\Sigma(X)$, and for $T = (\Sigma, \Gamma)$ μ acts just the same on equivalence classes of terms. The triple (T, μ, id) is a *monad* (MacLane, 1971), i.e., a monoid for the (monoidal) product given by functor composition $T \circ T = T^2$, with associativity and identity expressed by the expected commutative diagrams of natural transformation (MacLane, 1971). Then, a *monad morphism* $\alpha : (T, \mu, \eta) \to (T', \mu', \eta')$ is a natural transformation $\alpha : T \to T'$ such that it is a monoid homomorphism. Intuitively, a monad morphism $\alpha : T \to T'$ is an interpretation of equational theories that maps operations of T into (possibly derived) operations of T'.

Conversely, any monad T in the category $\mathbf{Set}$ is generated by an algebraic theory $T = (\Sigma, \Gamma)$, although the operations may be infinitary and range over all cardinals (Manes, 1976). The category of T-algebras can be recovered from the monad (T, μ, id) itself, since it is isomorphic to a category $\boldsymbol{Set}_T$ having as objects pairs $(X, q\colon T(X) \to X)$ such that q satisfies the condition of being an *action* of the monoid T (just as in automata theory; again see MacLane, 1971, for the two commutative diagrams). Morphisms $f\colon (X, q) \to (Y, q')$ are given by functions $f\colon X \to Y$ such that $q' \circ Tf = f \circ q$. For example, for T the theory of commutative monoids, the monoid structure of a given commutative monoid $(M, +, 0)$ can be recovered from the unique homomorphism $q\colon M^{\oplus} \to M$ induced by the identity function $1_M\colon M \to M$, and the map $(M, +, 0) \mapsto (M, q)$ is an isomorphism of categories $\boldsymbol{CMon} \simeq \boldsymbol{Set}_{(\ )^{\oplus}}$.

The notion of a monad was originally defined in (Eilenberg and Moore, 1965), where it was called *triple*. This notion is extremely valuable since it permits generalizing universal algebra over the category of sets to universal algebra over arbitrary categories, for which more subtle kinds of algebraic structures, not expressible in classical terms, may exist.

In this paper we have made crucial use of the fact that the category ***CMon*** is a symmetric monoidal closed category, and we have pointed out that this is a property common to the different versions of linear algebra provided by the choice of different commutative semirings. The symmetric monoidal closed structure is intimately connected with the fact that the semimodule operations "commute" with each other. For example, given $\lambda \in R$, a coefficient, and $+$ vector addition, we have

$$(\lambda x) + (\lambda y) = \lambda(x + y).$$

Linton (1966) proved an important characterization theorem showing that, indeed, for all *commutative* algebraic theories $T = (\Sigma, \Gamma)$ the category of algebras is symmetric monoidal closed, where a theory $T = (\Sigma, \Gamma)$ is called *commutative* iff for any two operations $\sigma\colon n \to 1$, $\tau\colon m \to 1$, the equation

$$\sigma(\tau(\mathbf{x}_{1j}), ..., \tau(\mathbf{x}_{nj})) = \tau(\sigma(\mathbf{x}_{i1}), ..., \sigma(\mathbf{x}_{in}))$$

with variables $\{x_{ij} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n\}$ and vectors $\mathbf{x}_{kj}$ and $\mathbf{x}_{ik}$ corresponding to the kth row and kth column, respectively, of the matrix of variables. This is equivalent to saying that for any operation $\sigma\colon n \to 1$ in Σ, and any T-algebra A, the operation $A_\sigma\colon A^n \to A$ is a Σ-homomorphism. A monad (T, μ, η) on ***Set*** is commutative iff it is the monad of a commutative theory (the definition extends naturally to infinitary operations). This can be extended to *strong* monads (T, μ, η) over an arbitrary symmetric monoidal closed category $\mathscr{V}$, i.e., monads such that T is a strong functor (also called a $\mathscr{V}$-functor) in the sense that T maps not only the external homs but also the internal homs, $T_{AB}\colon [A \to B] \to [TA \to TB]$ and μ and η are $\mathscr{V}$-natural transformations. This was done by Kock (1971), who gave a diagrammatic definition of commutative monad and showed that $\mathscr{V}_T$ is itself a closed category. As we shall see below, the notion of a commutative monad is extremely useful for generalizing the closed symmetric monoidal structure of Petri nets to more general notions of transition system.

7.2. *Some Monads*

Any right adjoint $U: \mathscr{A} \to \mathscr{B}$ with left adjoint $F: \mathscr{B} \to \mathscr{A}$ has associated natural transformations $\eta: 1_{\mathscr{B}} \to U \circ F$ ("insertion of generators") and $\varepsilon: F \circ U \to 1_{\mathscr{A}}$ ("evaluation") and generates a monad $T = (U \circ F, U\varepsilon F, \eta)$ (MacLane, 1971).

Therefore, the left adjoints N: ***Graph*** $\to$ ***GralPetri***, N_0: ***Graph*** $\to$ ***GralPetri***$_0$, and CM: ***Graph*** $\to$ ***CMonGraph*** to the forgetful functors ***GralPetri*** $\to$ ***Graph***, ***GralPetri***$_0$ $\to$ ***Graph***, and ***CMonGraph*** $\to$ ***Graph*** (in terms of functors already described, $N_0(G) = N(G)_0$ and $CM(G) = N(G)^{\oplus}$) yield monads N, N_0, CM: ***Graph*** $\to$ ***Graph***. It is not hard to check that these monads are strong for the cartesian closed structure of ***Graph***. We claim that the following isomorphisms exist:

$$\textbf{\textit{GralPetri}} \simeq \textbf{\textit{Graph}}_N$$

$$\textbf{\textit{GralPetri}}_0 \simeq \textbf{\textit{Graph}}_{N_0}$$

$$\textbf{\textit{CMonGraph}} \simeq \textbf{\textit{Graph}}_{CM}.$$

Similarly, N: ***RGraph*** $\to$ ***GralRPetri*** and CM: ***RGraph*** $\to$ ***CMonRGraph***, left adjoints to the obvious forgetful functors, yield strong monads on ***RGraph*** and isomorphisms

$$\textbf{\textit{GralRPetri}} \simeq \textbf{\textit{RGraph}}_N$$

$$\textbf{\textit{CMonRGraph}} \simeq \textbf{\textit{RGraph}}_{CM}$$

and the left adjoint CM: ***Cat*** $\to$ ***CMonCat*** to the forgetful functor ***CMonCat*** $\to$ ***Cat*** yields a strong monad CM on ***Cat*** with an isomorphism

$$\textbf{\textit{CMonCat}} \simeq \textbf{\textit{Cat}}_{CM}.$$

All this is just an abstract way of saying that all the above categories are categories of *algebras* but instead of being algebras over the category of sets, they are algebras over the categories ***Graph***, ***RGraph***, or ***Cat***. Of course, the notion of "being an algebra" is now more subtle, since, say, the algebraic structure on the nodes need not be the same as that of the arrows. This subtlety is what the notion of a monad captures.

7.3. *A Common Pattern*

We have already seen that the categories ***GralPetri***, ***GralPetri***$_0$, and ***CMonGraph*** are symmetric monoidal closed. In each case, the left adjoint operates by taking a graph $G = (\partial_0, \partial_1 : A \to V)$, generating some (possibly trivial) algebraic structure $T(A)$ on the arrows, generating a (possibly richer) algebraic structure $T'(V)$ on the nodes and then "lifting" the original ∂_0, ∂_1 in a natural way. In our examples, $T'(V)$ is always $V^{\oplus}$, whereas for ***GralPetri*** we have $T(A) = A$, for ***GralPetri***$_0$ $T(A) = A_0$ and for ***CMonGraph*** $T(A) = A^{\oplus}$. Of course, $1_{\boldsymbol{Set}}$, $(\_)_0$ and $(\_)^{\oplus}$ are monads, indeed *commutative* monads, associated to the theories of sets, pointed sets and commutative monoids, and the natural inclusions $\eta : 1_{\boldsymbol{Set}} \to (\_)^{\oplus}$, $(\_)_0 \to (\_)^{\oplus}$ are monad morphisms. A monad morphism $\alpha : T \to T'$ maps each operation $\sigma : n \to 1$ of T to a (derived) operation $\alpha(\sigma) : n \to 1$ in T'. Given a T-algebra X and a T'-algebra Y, a map $f: X \to Y$ is called an α-homomorphism iff for each operation $\sigma : n \to 1$ in T, $f(\sigma(x_1, ..., x_n)) = \alpha(\sigma)(f(x_1), ..., f(x_n))$. Note that for any T-homomorphism $g: X_0 \to Y$ and any T'-homomorphism $h: Y \to Y_1$, the compositions $f \circ g$ and $h \circ f$ are α-homomorphisms. This suggests the following theorem.

THEOREM 26. *Let T, T' be commutative monads on **Set** and $\alpha : T \to T'$ a monad morphism. Then the category **Graph**$_\alpha$ with objects $(\partial_0, \partial_1 : X \to Y)$ such that X is a T-algebra, Y is a T'-algebra and ∂_0, ∂_1 are α-homomorphisms is isomorphic to the category of algebras **Graph**$_{T_\alpha}$ for a monad $T_\alpha = (T_\alpha, \mu_\alpha, \eta_\alpha)$ on **Graph** that sends each $G = (\partial_0, \partial_1 : A \to V)$ to the graph $T_\alpha(G) = (TA \to^{\alpha A} T'A \rightrightarrows^{T'\partial_0}_{T'\partial_1} T'V)$, with $\mu_\alpha = \langle \mu_T, \mu_{T'} \rangle$ and $\eta_\alpha = \langle \eta_T, \eta_{T'} \rangle$. Besides, the category **Graph**$_\alpha$ is symmetric monoidal closed.*

Proof. The isomorphism ***Graph***$_\alpha \simeq$ ***Graph***$_{T_\alpha}$ follows easily from the isomorphism ***Alg***$_T \simeq$ ***Set***$_T$ and ***Alg***$_{T'} \simeq$ ***Set***$_{T'}$ at the ***Set*** level. The proof that ***Graph***$_\alpha$ is symmetric monoidal closed is a straightforward generalization of our proof for ***GralPetri***. Indeed, given $A = (\partial_0, \partial_1 : X \to Y)$ and $B = (\partial'_0, \partial'_1 : X' \to Y')$ in ***Graph***$_\alpha$, we define the internal hom $[A \to B]_\alpha$ as the following subgraph of $[A \to B]_G$. The nodes are those functions $g: Y \to Y'$ that are T'-homomorphisms. Therefore, since T' is commutative, they form a T'-algebra $[Y \to Y']_{T'}$. The arrows $(f, g, g'): g \to g'$ are those arrows of $[A \to B]_G$ such that f is a T-homomorphism and g, g' are T'-homomorphisms. They have a natural T-algebra structure defined as follows: if $\sigma : n \to 1$ is a T-operation, then we define $\sigma((f_1, g_1, g'_1), ..., (f_n, g_n, g'_n)) = (\sigma(f_1, ..., f_n),\ \alpha(\sigma)(g_1, ..., g_n),\ \alpha(\sigma)(g'_1, ..., g'_n))$, where the expressions in the right-hand side are well-defined functions in $[X \to X']_T$ and in $[Y \to Y']_{T'}$. Using the fact that ∂_0, ∂_1, ∂'_0, and ∂'_1 are α-homomorphisms, it is not hard to check that this indeed gives an arrow $\alpha(\sigma)(g_1, ..., g_n) \to \alpha(\sigma)(g'_1, ..., g'_n)$ in $[A \to B]_G$. The second and third projections are α-homomorphisms by the very definition of the T-structure on the arrows.

The tensor product $A \otimes_\alpha B$ is of the form $(\partial''_0, \partial''_1 : X \otimes_T X' \to Y \otimes_T Y')$, where ∂_i'' is the unique α-homomorphism induced by the following α-bilinear[11] map

$$A \times A' \xrightarrow{\eta_{T,A\times A'}} T(A \times A') \xrightarrow{T(\partial_i \times \partial_i')} T(B \times B') \xrightarrow{q} B \otimes_{T'} B',$$

where q is the unique T-homomorphism induced by the universal T'-bilinear map $B \times B' \to B \otimes_{T'} B'$. ∎

In particular, we have $\boldsymbol{GralPetri} = \boldsymbol{Graph}_{\eta_\oplus}$, $\boldsymbol{GralPetri}_0 = \boldsymbol{Graph}_{\alpha_2}$, and $\boldsymbol{CMonGraph} = \boldsymbol{Graph}_{\alpha_3}$ for $\eta_\oplus : 1_{Set} \to (\_)^\oplus$ the unit of $(\_)^\oplus$, $\alpha_2 = (\_)_0 \hookrightarrow (\_)^\oplus$ and $\alpha_3 = 1_\oplus : (\_)^\oplus \to (\_)^\oplus$.

7.4. *Generalizations*

This suggests a very general notion of *transition system* as given by a category of algebras over graphs, possibly of the form $\boldsymbol{Graph}_\alpha$ for $\alpha : T \to T'$ a monad map. Our motto "Petri nets are monoids" leads to the more general slogan "transition systems are algebras" (over $\boldsymbol{Graph}$). The possibilities are many. We sketch several of them below under the assumption that T and T' are commutative, and briefly discuss the noncommutative case in the conclusions.

7.4.1. *R-Petri nets.* Replace $\mathcal{N}$ by a commutative semiring R and define $\boldsymbol{GralPetri}_R = \boldsymbol{Graph}_{\eta_R}$ for η_R the unit of the monad of R-semimodules. For $R = \mathcal{Z}$ we obtain the *abelian* Petri nets that we already encountered in our discussion of duality and invariants. For R a distributive lattice L with top and bottom, we obtain a notion of *L-fuzzy* Petri nets.[12] All these monads are commutative and most of the results in the paper generalize to this setting, including the results on coincidence of finite products and coproducts for the categories where this held.

7.4.2. *Infinitary Petri nets.* These are Petri nets where a transition may have an infinite number of places as preconditions and as postconditions. Let $\mathcal{N}_\infty = \mathcal{N} \cup \{\infty\}$. For $\{n_i \mid i \in I, n_i \in \mathcal{N}_\infty\}$ and indexed family of arbitrary cardinality, we can define $\sum_{i\in I} n_i \in \mathcal{N}_\infty$ to be the usual sum in $\mathcal{N}$ if only finitely many of the n_i are nonzero and all $n_i \neq \infty$, or ∞ otherwise. There is a commutative monad $(\_)^{\oplus_\infty}$, sending each set S to the set $S^{\oplus_\infty}$ of all functions $f : S \to \mathcal{N}_\infty$. Such functions can be expressed as sums $f = \sum_{s\in S} f(s)s$. The unit $\eta_{\oplus_\infty}$ maps an element $s \in S$ to the function mapping

[11] Given T-algebras X, X', and X'', a map $f : X \times X' \to X''$ is *T-bilinear* iff each of the maps $\lambda x' \in X'.f(x, x')$ and $\lambda x \in X.f(x, x')$ is a T-homomorphism. Given a monad map $\alpha : T \to T'$, T-algebras X and X' and T'-algebra Y, a map $f : X \times X' \to Y$ is *α-bilinear* iff each of the maps $\lambda x' \in X'.f(x, x')$ and $\lambda x \in X.f(x, x')$ is an α-homomorphism.

[12] Actually, the most satisfactory notion would assume a complete distributive lattice and infinitary operations similar to those in Section 7.4.2 below.

s to 1 and everything else to 0; the monad multiplication $\mu_{\oplus_\infty}$ maps an element $\sum_{f \in S^{\oplus_\infty}} \lambda_f f$ to the function

$$\lambda s \in S \cdot \sum_{f \in S^{\oplus_\infty}} \lambda_f f(s).$$

$\mathcal{N}_\infty$ is a semiring, and any $(\_)^{\oplus_\infty}$-algebra has an underlying $\mathcal{N}_\infty$-semimodule structure. Also, $(\_)^{\oplus_\infty}$-homomorphisms are $\mathcal{N}_\infty$-linear maps and are closed under addition and zeros; i.e., $\boldsymbol{Set}_{(\_)^{\oplus_\infty}}$ is a semiadditive category where finite products coincide with finite sums, so that this carries over to categories of Petri nets whenever the transitions have the $(\_)^{\oplus_\infty}$-structure in a way entirely similar to the finitary case treated in this paper. The basic category of (generalized) infinitary Petri nets is $\boldsymbol{Graph}_{\eta_{\oplus_\infty}}$ and, by commutativity, it is a symmetric monoidal closed category.

7.4.3. *Probabilistic Petri nets.* Consider the "simplex" monad $\Delta : \boldsymbol{Set} \to \boldsymbol{Set}$ with

$$\Delta(X) = \left\{ \sum_i \lambda_i x_i \mid \lambda_i \in \mathcal{R}, \lambda_i \geqslant 0, x_i \in X, \sum_i \lambda_i = 1 \right\}$$

where the sums $\sum_i \lambda_i x_i$ are assumed to be finite, and with multiplication $\mu_\Delta : \Delta^2 \to \Delta$ defined by

$$\mu_{\Delta, X}\left(\sum_i \lambda_i \left[\sum_i \mu_{ij} x_j \right] \right) = \sum_j \left(\sum_i \lambda_i \mu_{ij} \right) x_j$$

and unit $\eta_\Delta : 1_{\boldsymbol{Set}} \to \Delta$ given by $\eta_{\Delta, X}(x) = 1x$. Then $\boldsymbol{Graph}_{\eta_\Delta}$ is a category of *probabilistic* (generalized) Petri nets where transitions are of the form $t : \lambda_1 a_1 + \cdots + \lambda_n a_n \to \mu_1 b_1 + \cdots + \mu_m b_m$. If transition t fired, then exactly one token was consumed (and $\lambda_1, ..., \lambda_n$ are the probabilities that the token was consumed from place $a_1, ..., a_n$) and one token was produced (and $\mu_1, ..., \mu_m$ are the probabilities that the token was produced in place $b_1, ..., b_m$). It is easy to check that the monad Δ is commutative, so that $\boldsymbol{Graph}_{\eta_\Delta}$ is symmetric monoidal closed by our general theorem. This category seems very well suited for applications, where transitions have a probabilistic nature, and should provide a fruitful and interesting link with the well-developed notion of Markov process in probability theory.

8. Conclusions

We have given a new definition of place/transition Petri nets as graphs equipped with the operations of parallel and sequential composition on the transitions. Known concepts, like case graphs and invariants, have been derived in a natural way. More importantly, new morphisms, relating system descriptions at rather different levels of abstraction, and new constructions, like a function space for Petri nets, have been defined.

It has been mentioned in Section 1 that transitions of Petri categories coincide, in the case of safe computations, with Petri nonsequential processes (Goltz and Reisig, 1983; Reisig, 1985). In the general case, however, the situation is more complex, and a full treatment can be found in (Degano *et al.*, 1989, 1989a), of which we give here a short account.

Best and Devillers (1987) observed that for general place/transition Petri nets, while one might expect processes to be more abstract than firing sequences and thus many firing sequences to correspond to the same process, the two notions are in fact incomparable. Thus they looked for a new notion of computation, more abstract than both firing sequences and processes. In a somewhat *ad hoc* manner, they defined a *swapping* operation on processes: when two concurrent instances of the same place can be found, their causal consequences can be exchanged. Equivalence classes with respect to swapping, which we may call *commutative processes*, are recognized as the least abstract model which is more abstract than both firing sequences and processes and is suggested as the correct observation level for nets.

In (Degano *et al.*, 1989, 1989a), commutative processes are proved isomorphic to the morphisms of $\mathcal{T}[N]$, thus providing an operational counterpart to the algebraic definition presented in this paper. Furthermore, another small category $\mathcal{P}[N]$ is proposed for modelling the classical notion of processes associated to a net N. In $\mathcal{P}[N]$, the same axioms hold as in $\mathcal{T}[N]$, except for the commutativity of parallel composition $\oplus$ of transitions. Instead, $\mathcal{P}[N]$ contains a subcategory of *symmetries* expressing the fact that in a marking the tokens on the same place can be permuted. A *coherence axiom* holds, which equates any parallel composition α of processes with another parallel composition α' of the same processes, where the different order between α and α' is compensated by composing suitable symmetries in sequence before α and after α'.

The main result of (Degano *et al.*, 1989, 1989a) is showing that the morphisms of $\mathcal{P}[N]$ are just a slight refinement, which we call *concatenable processes*, of classical processes. The refinement consists of imposing a total ordering among those minimal places (or "heads") of a process that are instances of the same place and a similar ordering for the maximal places (or "tails"). This makes possible to define a new general notion of *sequential composition* of processes, which of course corresponds to morphism composition in $\mathcal{P}[N]$.

Besides $\mathcal{P}[N]$, in (Degano *et al.*, 1989, 1989a) a category $\mathcal{S}[N]$ is introduced containing the classical firing sequences. Finally, a fourth category $\mathcal{K}[N]$ is added, providing a most concrete extremum for both $\mathcal{P}[N]$ and $\mathcal{S}[N]$. The axiom expressing the functoriality of parallel composition of transitions maps $\mathscr{K}[N]$ to $\mathscr{P}[N]$ and $\mathscr{S}[N]$ to $\mathscr{T}[N]$, while commutativity of parallel composition maps $\mathscr{K}[N]$ to $\mathscr{S}[N]$ and, as we saw, $\mathscr{P}[N]$ to $\mathscr{T}[N]$ (see Fig. 7). Thus the pushout diagram of the four categories gives a full account in algebraic terms of the relationship between interleaving and partial ordering observations of P/T net computations. It is easy to see that the morphism from $\mathscr{P}[N]$ to $\mathscr{T}[N]$ is bijective when restricted to safe computations.

Our development of algebraic theories on graphs can be extended by dropping the commutativity requirement altogether and consider arbitrary theories $T_\Sigma = (\Sigma, \varnothing)$ or $T_{\Sigma, E} = (\Sigma, E)$. This is intimately connected with the notion of *concurrent term rewriting* developed by the first author in joint work with J. A. Goguen and C. Kirchner (Goguen *et al.*, 1987) using more elementary methods. The study of this case will be the subject of a separate investigation (Meseguer, 1990).

Although we have for the most concentrated on the case of Petri nets, the general new concept that emerges from the present work is that of transition systems as graphs with algebraic structure. Computations of a transition system then appear as morphism of a path category generated by its graph. This path category will be endowed with an algebraic structure similar to that of the transition system. For Petri nets, the relevant algebraic structure is that of a commutative monoid, and therefore computations have a strict symmetric monoidal category structure, but this is just a particular case. For example, for Σ-term rewriting the category of computations has a Σ-algebra structure, and for infinitary Petri nets there

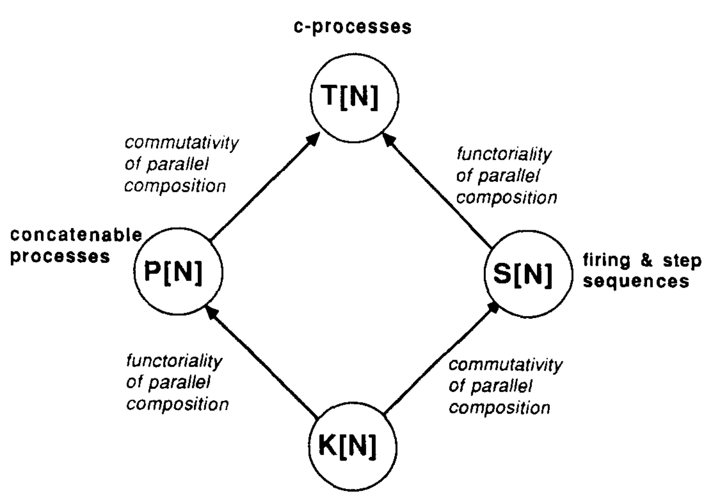

FIG. 7. The categories $\mathscr{K}[N]$, $\mathscr{P}[N]$, $\mathscr{S}[N]$, and $\mathscr{T}[N]$ and their semantic relationship.

is an infinitary parallel composition of computations. In each case, there will be a "distributive law" relating sequential and parallel composition of computations. Considerations of this kind should lead to a general algebraic (meta) model of true concurrency of wide applicability.

The categorical approach we have outlined here should provide the framework necessary to develop a hierarchy of models where the necessary structure is introduced only at the proper level, as advocated for instance in (Degano and Montanari, 1985). The structure to be added includes, for instance, actions, invisible actions, a synchronization mechanism, a term structure with variables and substitutions, and a spatial structure on the places.

Acknowledgments

We thank Joseph Goguen, for his extensive comments to a previous draft and for his very valuable technical suggestions, and F. William Lawvere, for his comments and his many enlightening conversations with the first author on categorical matters. We also thank Friedrich von Henke, Gordon Plotkin, and Wolfgang Reisig who made very helpful suggestions that led to improvements in the paper.

Received February 3, 1988; final manuscript accepted August 29, 1989

References

Benson, D. B. (1975), The basic algebraic structures in categories of derivations, *Inform. and Comput.* **28**, 1–29.

Best, E., and Devillers, R. (1987), Sequential and concurrent behaviour in Petri net theory, *Theoret. Comput. Sci.* **55**, 87–136.

Boudol, G., and Castellani, I. (1988), Concurrency and atomicity, *Theoret. Comput. Sci.* **59**, 25–84.

Degano, P., De Nicola, R., and Montanari, U. (1988), A distributed operational semantics for CCS based on condition/event systems, *Acta Inform.* **26**, 59–91.

Degano, P., Meseguer, J., and Montanari, U. (1989), Axiomatizing net computations and processes, *in* "Proceedings, 4th Symposium on Logic in Computer Science, IEEE, June 1989, Asilomar, CA," pp. 175–185.

Degano, P., Meseguer, J., and Montanari, U. (1989a), "Axiomatizing the Algebra of Net Computations and Processes," Technical Report SRI Int., Menlo Park and Dipartimento di Informatica, Pisa.

Degano, P., and Montanari, U. (1985), Specification languages for distributed systems, *in* "TAPSOFT" (H. Ehrig *et al.*, Eds.), pp. 29–51, Lect. Notes in Comput. Sci., Vol. 185, Springer-Verlag, New York/Berlin.

Degano, P., and Montanari, U. (1987), Concurrent histories: A basis for observing distributed systems, *J. Comput. System Sci.* **34**, Nos. 2/3, 422–461.

Drees, S., Gomm, D., Plünnecke, H., Reisig, W., and Walter, R. (1986), "Bibliography of Net Theory," Arbeitspapiere der GMD 212, Gesellschaft für Mathematik und Datenverarbeitung MBH.

Eilenberg, S., and Moore, J. C. (1965), Adjoint functors and triples, *Illinois J. Math.* **9**, 381–398.

GOGUEN, J., KIRCHNER, C., AND MESEGUER, J. (1987), Concurrent term rewriting as a model of computation, *in* "Proceedings, Workshop on graph reduction, Santa Fe, New Mexico" (R. Keller and J. Fasel, Eds.), pp. 53–93, Lect. Notes in Comput. Sci., Vol. 279, Springer-Verlag, New York/Berlin.

GOLTZ, U., AND REISIG, W. (1983), The nonsequential behavior of Petri nets, *Inform. and Comput.* **57**, 125–147.

HEROLD A., AND SIEKMANN, J. (1987), Unification in abelian semigroups, *J. Automated Reasoning* **3**, 247–283.

HERRLICH, H., AND STRECKER, G. E. (1973), "Category Theory," Allyn & Bacon, Rockleigh, NJ.

HINDERER, W. (1982), Transfer of graph constructs in goguen's paper to net constructs, *in* "Application and Theory of Petri Nets" (C. Girault and W. Reisig, Eds.), pp. 142–150, Informatik-Fachberichte, Vol. 52, Springer-Verlag, New York/Berlin.

HOTZ, G. (1965), Eine algebraisierung des syntheseproblemen von schaltkreisen, i and ii, *EIK* **1**, 185–206, 209–231.

KOCK, A. (1971), Closed categories generated by commutative monads, *J. Austral. Math. Soc.* **12**, 405–424.

LAWVERE, F. W. (1971), Quantifiers and sheaves, *in* "Actes Congrès Intl. Math., Nice 1970," Vol. 1, pp. 329–334, Gauthier-Villars, Paris.

LAWVERE, F. W. (1989), Qualitative distinctions between some toposes of generalized graphs, *in* "Proceedings, AMS Summer Research Conference on Categories in Computer Science and Logic, Boulder, Colorado, June 1987" (J. Gray and A. Scedrov, Eds.), Amer. Math. Soc., Providence, RI.

LINTON, F. E. J. (1966), Autonomous equational categories, *J. Math. Mech.* **15**, 637–642.

MACLANE, S. (1971), "Categories for the Working Mathematician," Springer-Verlag, New York/Berlin.

MACLANE, S., AND BIRKHOFF, G. (1967), "Algebra," Macmillan, New York.

MAIN, M. G., AND BENSON, D. B. (1984), Functional behaviour of nondeterministic and concurrent programs, *Inform. and Comput.* **62**, 144–189.

MANES, E. (1976), "Algebraic Theories," Graduate Texts in Mathematics, Vol. 26, Springer-Verlag, New York/Berlin.

MESEGUER, J. (1990), Rewriting as a unified model of concurrency, Technical Report SRI-CSL-90-02, SRI International, Computer Science Laboratory, February 1990.

MESEGUER, J., AND MONTANARI, U. (1988), Petri nets are monoids: A new algebraic foundation for net theory, *in* "Proceedings, 3rd Symposium on Logic in Computer Science," pp. 155–164, IEEE, New York.

MESEGUER, J., AND SOLS, I. (1975), Automata in semimodule categories, *in* "Category Theory Applied to Computation and Control" (E. G. Manes, Ed.), pp. 193–198, Lect. Notes in Comput. Sci., Vol. 25, Springer-Verlag, New York/Berlin.

MILNER, R. (1985), Lectures on a calculus for communicating systems, *in* "Control Flow and Data Flow: Concepts of Distributed Programming" (M. Broy, Ed.), pp. 205–228, NATO ASI Ser. F, Vol. 14, Springer-Verlag, New York/Berlin.

MILNER, R. (1982), "Calculi for Synchrony and Asynchrony," Technical Report, CSR-104–82, Comput. Sci. Dept., University of Edinburgh, February.

OLDEROG, E.-R. (1987), Operational Petri net semantics for ccsp, *in* "Advances in Petri Nets 1987" (G. Rozenberg, Ed.), pp. 196–223, Springer-Verlag, New York/Berlin.

PETRI, C. A. (1973), Concepts of net theory, *in* "Mathematical Foundations of Computer Science," pp. 137–146, Math. Inst. of the Slovak Academy of Sciences.

REISIG, W. (1985), "Petri Nets," Springer-Verlag, New York/Berlin.

VAN GLABBEEK, R., AND VAANDRAGER, F. (1987), Petri net model for algebraic theories of concurrency, *in* "PARLE Conference" (J. W. de Bakker *et al.*, Eds.), Lect. Notes in Comput. Sci., Vol. 259, Springer-Verlag, New York/Berlin.

WINKOWSKI, J. (1982), An algebraic description of system behaviours, *Theoret. Comput. Sci.* **21**, 315–340.

WINSKEL, G. (1984), Categories of models for concurrency, *in* "Workshop on the Semantics of Concurrency" (S. Brooks, Ed.).

WINSKEL, G. (1986), "Event Structures," Technical Report 95, University of Cambridge Computer Laboratory, July.

WINSKEL, G. (1987), Petri nets, algebras, morphisms, and compositionality, *Inform. and Comput.* **72**, 197–238.