Dialectica Petri Nets

Elena Di Lavore*
Department of Software Science
Tallinn University of Technology
Tallinn, Estonia

Department of Computer Science
University of Pisa
Pisa, Italy

Wilmer Leal†
Department of Computer Science
University of Florida
Gainesville, Florida, USA
Max Planck Institute for Mathematics in the Sciences
Leipzig, Germany
Bioinformatics Group, Department of Computer Science
Universität Leipzig
Leipzig, Germany

Valeria de Paiva
Topos Institute
Berkeley, California, USA
Abstract. The categorical modelling of Petri nets has received much attention recently. The Dialectica construction has also had its fair share of attention. We revisit the use of the Dialectica construction as a categorical model for Petri nets generalising the original application to suggest that Petri nets with different kinds of transitions can be modelled in the same categorical framework. Transitions representing truth-values, probabilities, rates or multiplicities, evaluated in different algebraic structures called lineales are useful and are modelled here in the same category. We investigate (categorical instances of) this generalised model and its connections to more recent models of categorical nets.

Keywords: Petri nets, Dialectica categories, categorical models of Petri nets, linear logic of Petri nets, symmetric monoidal closed category, lineale, chemical reaction networks

1. Introduction

Petri nets exert endless fascination over category theorists. Maybe category theorists see Petri nets as a gauntlet thrown at them, because the definition of a morphism of Petri nets is not obvious and different definitions lead to different categories. Maybe the bipartite graphs that usually depict Petri nets look too similar to automata ones, and these are the initial sources of good categorical examples in computing. In any case, many different categorical models of Petri nets do exist and some are fundamentally different from others. One fundamental difference is whether one concentrates on the token game and the behaviour of a given Petri net or on the graphs underlying different nets. Another difference is which possible operations combining different Petri nets one considers. Alternatively, the difference may lie in the type of relationships (labels) that the Petri net can model.

In this work, we present a categorical model of Petri nets with linear connectives. We explore the model originally introduced by Winskel [Win87, Win88], but use it with morphisms, as in the work of Brown [BG90] and others, that relate Petri nets to constructors in Linear Logic [Gir87]. These connectives enable the assembly of small networks into larger ones in a principled manner. The model is flexible enough to capture a broad range of relationships, provided that the set of labels encoding these relationships can be transformed into a lineale, a poset version of a symmetric monoidal closed category.

Petri nets. A Petri net is simply a directed bipartite graph that has two types of elements, conditions and events (also called places and transitions). These are usually depicted as circles and rectangles, respectively (Figure 1, right). This work is not concerned with the dynamic behaviour of Petri nets while it focuses on their structure. However, keeping in mind the intended semantics of a net may help with intuitions. When describing the dynamic behaviour of a Petri net, one starts with an initial marking. This consists of a number of tokens, depicted as black circles inside places, for every place in the net as shown in Figure 2. Over this fixed structure of possible events and conditions, a causal

---

1Elena Di Lavore was supported by the European Social Fund Estonian IT Academy research measure (project 2014-2020.4.05.19-0001).

1Wilmer Leal was supported by the German Academic Exchange Service (DAAD): Forschungsstipendien-Promotionen in Deutschland, 2017/2018 (Bewerbung 57299294).
\[
aA + bB \rightarrow cC + dD
\]
\[
eE + c'C \rightarrow fF
\]
\[
c''C + d'D \rightarrow gG
\]

Figure 1: Chemical reaction data represented as a list (left) and as a Petri net (right). A, B, \ldots, H are substances and \(a, b, \ldots, h\) are stoichiometric coefficients that indicate the proportion in which they combine.

Figure 2: A Petri net with an initial marking.

dependency (or flow) relation between sets of events and conditions is described via pre- and post-relations, and it is this structure which determines the possible dynamic behaviour of the net. A transition in this causal dependency relation is enabled if all places connected to it as inputs contain at least the number of tokens indicated by the arc. The transition \(r\) in Figure 2 is enabled as its preconditions contain enough tokens, while transition \(s\) needs \(r\) to fire first as \(r\) would produce enough tokens for \(s\) to fire. The dynamic behaviour of a Petri net is often referred to as “token game”.

**Petri nets via Dialectica Categories.** Petri nets were described categorically in several works [BG90, MM90, MOM05] and are still been discussed [RSS14, BM20, Mas19]. Models need to capture the practitioner’s imagination and make themselves useful, both for calculations and for insights. Categorical models can be useful for both insights and calculations, but we have not seen categorical models that encompass different kinds of transitions in a single net.

Petri nets were modelled using Dialectica categories [dP91b] previously, but the original Brown and Gurr model [BG95] worked only for *elementary nets*, that is nets whose transitions are marked with \(\{0,1\}\) for presence or absence of a relationship. An extension of this modelling to deal with integers \(\mathbb{N}\) was planned [dP91a, BGdP91], but never published. In this work we put together different kinds of transitions, in a single categorical framework. This way the categorical modelling applies to the many kinds of newer applications [EDLR16] that already use different kinds of labels on the
transitions (see Section 2.1 for a brief account of Petri net transitions often used in applications).

The original dialectica construction [dP91b] was given in two different styles called the categories DC [dP89a] and the categories GC [dP89b]. For both constructions C is a cartesian closed category with some other structure. The first style is connected to Gödel’s Dialectica Interpretation hence the ‘D’ in DC for Dialectica. The second style called GC ([dP89b]) is based on a suggestion of Girard’s (hence the ‘G’) on how to simplify the first construction, if one wants a model of Linear Logic. These two constructions are connected, via monoidal comonads as described in [dP91b]. Here we are mostly interested in the construction called GC, whose morphisms are simpler. When the category C is Set, this construction can be presented in two ways. This is because a relation in Set between U and X can be thought of as either a subset of the product, \( \alpha \subseteq U \times X \), or as map into 2, \( \alpha: U \times X \to 2 \).

The work here uses only the second presentation, defining general relation maps into algebraic structures called lineales. This is because changing the lineale where our relations take ‘values’, gives us the possibility of modelling several different kinds of processes. The original dialectica construction deals only with the Heyting algebra-like lineale 2. Here we discuss several other lineales and dialectica categories built over these different lineales.

We will use some intuitions from the game semantics for linear logic as described by Blass [Bla92], who explicitly compares it with de Paiva’s Dialectica interpretation [dP91b].

This paper is about models of Petri nets in Dialectica categories. Petri nets have several different definitions in the literature and multifaceted applications as well. This work emphasises the underlying graph of a Petri net, which is a kind of labelled, directed L-weighted hypergraph, and focuses on a static semantics, simply in terms of ways of putting together combinations of underlying graphs. This is a first step in understanding these ubiquitous modelling systems, their executions and their dynamic semantics. While we will suggest semantic interpretations of the different logics for Petri nets, we focus on their compositions via linear logic connectives. In this, we follow a line of research that explores how to combine Petri nets with linear logic connectives [BG95, EW90].

Related work. Our work fits in the vast landscape of categorical approaches to Petri nets, building on [BGdP91, dP91a]. Meseguer and Montanari’s seminal work [MM90] focused on reachability properties of Petri nets, defining a category of all possible executions of a net. This work adopted the collective tokens philosophy. Its ideas were extended to the individual tokens philosophy in [BMMS01]. Marti-Oliet and Meseguer explore the connections between linear logic and Petri nets [MOM05]. Other categorical models of Petri nets focus on obtaining nets by composing smaller nets along some boundaries. One of the first compositional models doing this was [KSW97] where nets are composed along common places. In [RSS14], nets are composed along common transitions and compositionality is used to study reachability properties of Petri nets. The work of [BG90, dP91a] and [BGdP91] concentrates on combining Petri nets via different monoidal products that give to the category of Petri nets a linear logic structure. More recently, there has been numerous works building on the ideas of [MM90] and adopting the formalism of [KSW97]. In [BM20] and [Mas20], the authors focus on studying the categorical properties of reachability. In [Koc20] a more fine-grained categorical model is proposed, that allows Kock to encompass the individual and collective token philosophies in the same framework. Finally, [BGMS21] constructs a unifying framework for [MM90, BMMS01] and [Koc20] extending [Mas20].
Our work extends the approach of [BG90] to allow different kinds of arcs, e.g. inhibitor, probabilistic, partially defined, natural/integer numbers valued, and the coexistence of them in the same net.

Outline. In Section 2, we will first provide a brief overview of the types of labels commonly used in Petri net applications. We will use these findings to motivate the need for a categorical model of Petri nets capable of encoding this broad spectrum of relations. We will conclude this section by transforming sets of labels into structures called lineales, whose configuration will later be leveraged to build linear logic connectives. We show that lineales form a cartesian category.

The construction of our categorical model of Petri nets will take place in two steps. First, in Section 3, we will build an intermediate category called $M_L \text{Set}$, that will be used to encode pre- and post-conditions. Its name is a mnemonic for multisets, where the multitude of elements is specified by values in the lineale $L$ instead of by just natural numbers. Within this category, we will define linear connectives and prove that it forms a symmetric monoidal closed category. Subsequently, in Section 4, we will take two instances of the category $M_L \text{Set}$, one to encode preconditions and the other to encode postconditions, and glue them together (by taking a pullback in the category $\text{Cat}$) to build our category $\text{Net}_L$ of Petri nets. Additionally, in this section, we will develop logical connectives for the category $\text{Net}_L$ and, in particular, demonstrate that it inherits the symmetric monoidal closed structure from $M_L \text{Set}$.

Finally, Section 5 presents examples of Petri nets spanning a wide spectrum of linear structures, all motivated by various applications, and concludes by showing functoriality of the constructions $M_L \text{Set}$ and $\text{Net}_L$.

2. Lineales: a transition structure for each application of Petri nets

In order to define our category of Petri nets, we will take a look at applications to motivate the kind of structure, that of a lineale, that will be used to encode a general class of pre- and post-condition relations. After exposing the motivating application in Section 2.1, this section presents the definition of lineale (Section 2.2) and gives examples of them (Section 2.3).

2.1. Chemical and metabolic networks: an inspiration

Networked systems are determined by their connections [EFL+20]. Perhaps the most basic type of relationship in any network is one that only allows us to express either presence or absence, that is, where the relationship connecting nodes uses the set $\{0,1\}$ as a ruler or label set. In real-world applications this is, though, not sufficient. In this section we explore frequent and rich applications of Petri nets, from chemical reaction networks to metabolic networks, searching for the kind of labels used on pre- and post- conditions.

Chemical reaction networks. Chemical combination is compositional in nature. Although data on substance reactivity are typically annotated as a list of chemical equations (Figure 1, left), chemists reason on the network structure (Figure 1, right) that emerges when the reactions are connected to make their concurrency explicit [Sch98]. Synthesis planning is a prominent example: suppose we
want to synthesise substance F, but we cannot carry out reaction $r_2$, because we have no substance C on our lab’s shelf. In such a case, equation $r_1$ provides another synthetic route, since it is possible to obtain F from A and B (and E). In other words, the synthesis of F results from composing reactions $r_1$ and $r_2$.

Directed hypergraphs and their enhancements, such as Petri nets, are used to model chemical reaction networks for they are models of concurrency of directed relations. These models provide a rich semantic basis on which to interpret questions that arise in chemistry, such as what substances can be synthesised from a given set of starting materials? [SS18] Given a target substance, which synthetic routes are known and which starting materials are needed to reach the target? [Hof09] Do chemical reactions turn targets into key precursors? [EFL+20] How many synthetic routes pass through a given reaction? These questions can be answered by probing the topology and geometry of the wiring of the network. The first two questions are answered by defining suitable closure operators [SS18], and the last two questions by computing the curvature of the edges of the network [LRSJ21], taking into consideration the proportions in which substances combine (stoichiometric coefficients).

At the level of abstraction described above, transitions of chemical reaction networks are discrete in nature, and pre- and post- conditions correspond to presence/absence of substances or to stoichiometric coefficients, which can be modelled by the rulers $\{0, 1\}$ and $\mathbb{N}$, respectively.

**Metabolic networks.** These networks comprise the metabolic pathways (network of chemical reactions) and the gene interactions that regulate them. A key aspect of the former is the kinetic modelling. There, Petri nets model reaction rates. For elementary reactions, which take place in a single step, the Law of Mass Action states that reaction rates are proportional to the concentration of reactants. Both quantities, rate of reactions and concentration of reactants, are usually taken as positive real numbers; therefore, in this application, Petri nets are challenged to handle continuous tokens and transitions, which requires the ruler $\mathbb{R}^+$. On the other hand, gene interactions are handled by implementing genetic switches that are modelled by discrete transitions. A Petri net model for a metabolic network therefore needs two different rulers on the same net.

When applied to concrete metabolisms, a Petri net model will usually need to incorporate more than two rulers at the same time. For instance, [SNMM11] shows a hybrid Petri net representation of the gene regulatory network of *C. elegans* that is labelled with discrete and continuous transitions, but also with negative integers, real numbers, strings, and products of them.

Summarising, applications may need rulers such as $\{0, 1\}$, $\{-1, 0, -1\}$ (for data uncertainty, which is common in complex network systems [NM20]), $\mathbb{N}$, $\mathbb{R}^+$, $\mathbb{Z}$, $\mathbb{R}$, strings, and their finite products. The ability of choosing from a vast pool of rulers to label pre- and post- conditions is one of the strengths of the categorical construction presented in this paper.

### 2.2. Lineales: the codomain of general relations

Classical relations are functions of type $U \times X \rightarrow 2$. If a relation assigns 1 to a pair $(u, x)$, then the two elements are related, otherwise they are not. In some applications, one would like more information about the relationship between the pair $(u, x)$. For example, we could record the intensity of the relation by assigning to each pair $(u, x)$ a natural number. In general, we can define poset-valued relations. For our purposes, posets will have additional structure: a multiplication and an internal-
The multiplication makes the poset also a monoid. Section 3 uses this structure to define a monoidal closed structure on lineale-valued relations (Theorem 3.1).

**Definition 2.1. (Partially ordered monoid)**

A partially ordered commutative monoid \((L, \sqsubseteq, *, e)\) is a commutative monoid \((L, *, e)\) equipped with a partial order \(\sqsubseteq\) that is compatible with the monoidal operation, i.e. if \(a \sqsubseteq b\) and \(a' \sqsubseteq b'\) then \(a * a' \sqsubseteq b * b'\).

The internal-hom is the best approximation to an inverse operation for the multiplication of the monoid.

**Definition 2.2. (Internal-hom in a monoid)**

Let \((L, \sqsubseteq, *, e)\) be a partially ordered commutative monoid. A binary operation \(\rightarrow: L^{op} \times L \rightarrow L\) is said to be an internal-hom when it is right adjoint to the monoidal product \(*\), i.e. \(\forall a, b, c \in L, b * c \sqsupseteq a \iff b \sqsubseteq c \rightarrow a\). The internal-hom is also required to respect the ordering, contravariantly in the first coordinate and covariantly in the second, i.e. if \(b \sqsupseteq a\) and \(a' \sqsupseteq b'\) then \(a \rightarrow a' \sqsupseteq b \rightarrow b'\).

A lineale \([dP02]\) is a monoidal closed poset. This means that a lineale has a commutative monoid structure with a right adjoint, the internal-hom, that are both compatible with the order. This structure is also known as a residuated ordered commutative monoid \([H¨oh95]\).

**Definition 2.3.** A lineale is a tuple \((L, \sqsubseteq, *, e, \rightarrow)\) such that \((L, \sqsubseteq, *, e)\) is a partially ordered monoid and \(\rightarrow\) is an internal-hom for \((L, \sqsubseteq, *, e)\).

Notice that in any lineale, \(b = e \rightarrow b\) for any \(b \in L\). Since \(b * e \sqsupseteq b\) holds in any monoid \((L, *, e)\), it follows by the definition of \(\rightarrow\) that \(b \sqsupseteq e \rightarrow b\) for all \(b \in L\). Moreover, choosing \(e = e \rightarrow b\) in \(c * e \sqsupseteq b \iff c \sqsupseteq e \rightarrow b\) we obtain \(e \rightarrow b = (e \rightarrow b) * e \sqsupseteq b \iff e \rightarrow b \sqsupseteq e \rightarrow b\). Since the right side of the equivalence is trivially true, we must have \(e \rightarrow b \sqsupseteq b\).

**Definition 2.4.** A morphism of lineales \(h: (L, \sqsubseteq, *, e, \rightarrow) \rightarrow (L', \sqsubseteq', *, e', \rightarrow')\) is a monotone function \(h: (L, \sqsubseteq) \rightarrow (L', \sqsubseteq')\) that laxly preserves the monoid structure, i.e. \(e' \sqsupseteq' h(e)\) and \(h(a) \rightarrow h(b) \sqsupseteq' h(a * b)\).

In the same way that lineales are posetal monoidal closed categories, morphisms of lineales are lax monoidal functors between them. Lineales and their morphisms form a category \(\text{Lin}\). Note that the preservation of the internal-hom follows from the adjointness property: since \(h(a \rightarrow b) \sqsupseteq' h(a) \rightarrow h(b)\), then \(h(a \rightarrow b) \sqsupseteq' h(a) \rightarrow h(b)\).

The next lemma recalls the relationship between minima and internal-homs. It is a poset instance of the well-known adjoint functor theorem, see, for instance, Section V.6 in \([Mac71]\).

**Lemma 2.1.** A partially ordered monoid \((M, \sqsubseteq, *, e)\) in which infima always exist defines a lineale with internal-hom \(a \rightarrow b = \inf\{x \in M \mid x * a \sqsupseteq b\}\).

For partially ordered groups, the internal-hom is exactly the inverse operation to the multiplication. In fact, any partially ordered group is a lineale with \(a \rightarrow b = b * a^{-1}\) (see also \([ST14, \ Example \ 4.4.1]\)). Some of our examples will fall into this case.
Definition 2.5. (Partially ordered group)
A partially ordered group \((G, \sqsupseteq, *, e, (-)^{-1})\) is a partially ordered monoid \((G, \sqsupseteq, *, e)\) together with an inverse operation \((-)^{-1}\) that makes \((G, *, e, (-)^{-1})\) a group and respects the ordering contravariantly, i.e. if \(a \sqsupseteq b\) then \(b^{-1} \sqsupseteq a^{-1}\).

Lemma 2.2. A partially ordered group \((G, \sqsupseteq, *, e, (-)^{-1})\) can be endowed with the structure of a lineale with \(a \nrightarrow b := b * a^{-1}\).

Proof:
A group is a monoid, thus we only need to check that \(a \nrightarrow b\) actually defines an internal-hom and that it respects the ordering.

\[
\begin{align*}
  b * c \sqsupseteq a & \iff b * c * c^{-1} \sqsupseteq a * c^{-1} & b \sqsupseteq a \land a' \sqsupseteq b' \\
  \Rightarrow b \sqsupseteq c \nrightarrow a & \Rightarrow a^{-1} \sqsupseteq b^{-1} \land a' \sqsupseteq b' \\
  \Leftrightarrow b \sqsupseteq c \nrightarrow a & \Rightarrow a' * a^{-1} \sqsupseteq b' * b^{-1} \\
  & \Rightarrow a \nrightarrow a' \sqsupseteq b \nrightarrow b'
\end{align*}
\]

2.3. Examples of lineales
While the lineale 2 is associated with Boolean and Heyting algebras, which are traditional algebraic models for classical and intuitionistic propositional logic, other lineales are associated with different non-classical systems. We describe some lineales and the non-classical logic associated to them. Finally, we consider the coexistence of several lineales in a single net by taking products of them.

Example 2.1. (Classical lineale)
The original work on the categorical version of the Dialectica interpretation has concentrated on relations that take values into 2, considered as a lineale. The set \(2 = \{0, 1\}\) has a lineale structure \((2, \geq, \lor, 0, \nrightarrow)\), where \(1 \geq 0\), the operation \(\lor\) is the logical disjunction if 0 is interpreted as false and 1 as true, and the internal-hom is defined as \(a \nrightarrow b = \min\{x \in 2 \mid x \lor a \geq b\}\). This is a lineale structure by Lemma 2.1. More explicitly, \(a \nrightarrow b = 0\) if \(a \geq b\) and \(a \nrightarrow b = b\) otherwise. This operation is the adjoint of the disjunction and differs from the logical implication, which is adjoint to the conjunction. In fact, \(a \nrightarrow b = \neg (b \rightarrow a)\).

Example 2.2. (Kleene lineale)
We describe a 3-valued propositional logic where the undefined truth-value, the “unknown” state, can be thought of as neither true nor false. We interpret 0 and 1 as false and true respectively. The additional truth value \(u\) represents undefined. The three elements set \(3 = \{0, u, 1\}\) has a lineale structure \((3, \geq, \max, 0, \nrightarrow)\), where \(1 \geq u \geq 0\) and the internal-hom is defined as \(a \nrightarrow b = \min\{x \in 3 \mid \max\{x, a\} \geq b\}\). More explicitly, \(a \nrightarrow b = 0\) if \(a \geq b\) and \(a \nrightarrow b = b\) otherwise. This is indeed a lineale by Lemma 2.1.
We should also mention the lineale 4, associated with Belnap-Dunn’s four-valued logic [BJ77]. These four values also correspond to the algebraic identities for the two conjunctions and two disjunctions of Linear Logic.

**Example 2.3. (Multirelations lineale)**

We consider a lineale defined on the natural numbers, \((\mathbb{N}, \geq, +, 0, \rightarrow)\), where the order is the usual order on natural numbers, \(m \geq n\) iff \(m \geq n\), and the monoid structure is that of addition. While we can think of the classical truth values as indicating whether or not two elements are related, we can think of the natural numbered truth values as indicating how many times two elements are related. Note that this lineale is not a quantale as suprema need not exist. The internal-hom is defined as in Lemma 2.1 and gives a lineale structure. In the case of natural numbers, this internal-hom, \(a \rightarrow b = \min\{n \in \mathbb{N} \mid n + a \geq b\}\), becomes \(b - a\), when \(b \geq a\), and 0, otherwise. In other words, \(a \rightarrow b = \max\{0, b - a\}\).

**Example 2.4. (Integers lineale)**

Similarly to the multirelations lineale, we consider a lineale structure on the integers, \((\mathbb{Z}, \geq, +, 0, \rightarrow)\). As the monoid of integers with addition is actually a group, we can apply Lemma 2.2 to choose the internal-hom structure as subtraction: \(a \rightarrow b = b - a\).

**Example 2.5. (Probabilistic lineale)**

Next we consider the reals, in the form of the closed interval \([0, 1]\). These have long been considered for fuzzy sets, as the real number associated with a pair \((u, x)\) can be thought of as the probability of the association between \(u\) and \(x\). We show that the closed interval \([0, 1]\) admits a lineale structure.

- The monoid structure is given by the product of real numbers, \(a \cdot b := a \cdot b\), whose unit is 1.
- The partial order is given by the usual ordering on the reals.
- The internal-hom is given by a ‘truncated division’ \(a \rightarrow b := \begin{cases} \frac{b}{a} & a > b \\ 1 & \text{otherwise} \end{cases} \). This structure defines a lineale by Lemma 2.1: if \(a > b\), then \(\inf\{x \in [0, 1] \mid x \cdot a \geq b\} = \frac{b}{a}\), while if \(a \leq b\), then \(\inf\emptyset = 1\).

**Example 2.6. (Morphism of lineales)**

Consider the Multirelations lineale from Example 2.3 and the Classical lineale from Example 2.1. The function \(h : \mathbb{N} \rightarrow 2\), defined by \(h(0) = 0\) and \(h(n) = 1\) for all \(n \geq 1\), is a morphism of lineales. In fact, it is monotone, if \(m \leq n\) then \(h(m) \leq h(n)\), and it preserves the unit, \(h(0) = 0\), and the multiplication, \(h(m + n) = h(m) \lor h(n)\). If we interpret the logical values in \(\mathbb{N}\) as recording ‘how many times’ a proposition is true, this morphism forgets this number and only records whether or not the proposition is true.
**Product of lineales.** We have produced a pool of lineales, each of them suitable for transitions taking values in certain data types \(\{0, 1\}, \{-1, 0, -1\}, \mathbb{N}, \mathbb{Z}\) or \(\mathbb{R}^+\). As discussed in Section 2, in empirical data analysis, a transition often carries data on more than one variable simultaneously. In this section, we show that any finite combination of lineales can be endowed with the structure of a lineale by taking finite products of them.

As lineales are just the poset-version of symmetric monoidal closed categories, the next proposition is the posetal version of the analogous result for symmetric monoidal closed categories. Symmetric monoidal closed categories form a category \(\text{SymClosedCat}\) whose objects are symmetric monoidal closed categories, and morphisms are functors that preserve the adjunction. It is folklore that this category is cartesian. For completeness, we directly show the particular case of our interest: the component-wise product of two lineales is a lineale.

**Proposition 2.1.** If \((L_1, \leq_1, *_1, e_1, -o_1)\) and \((L_2, \leq_2, *_2, e_2, -o_2)\) are lineales, then \(L_1 \times L_2\) has a lineale structure where, for elements \(l = (l_1, l_2), l' = (l'_1, l'_2) \in L_1 \times L_2\),

- the product is defined component-wise, \(l * l' = (l_1 * l'_1, l_2 * l'_2)\);
- the unit is the pair of units, \(e = (e_1, e_2)\);
- the order \(l \leq l'\) holds if and only if \(l_1 \leq_1 l'_1\) and \(l_2 \leq_2 l'_2\); and
- the internal-hom is defined component-wise, \(l \rightarrow l' = (l_1 \rightarrow_1 l'_1, l_2 \rightarrow_2 l'_2)\).

This operation is the product in the category of lineales, \(\text{Lin}\).

**Proof:**

\((L_1 \times L_2, *, e)\) is the cartesian product of two monoids and therefore it is a monoid. \((L_1 \times L_2, \leq)\) is a partial ordered set with the ordering defined above. Since \(l_i \leq l'_i\) implies both \(l_i * i k_i \leq l'_i * i k_i\) and \(k_i * i l_i \leq k_i * i l'_i\) for each \(l_i, k_i \in L_i\) and \(i = 1, 2\); then \(l \leq l'\) implies \(l * k \leq l' * k\) and \(k * l \leq k * l'\) for every \(k = (k_1, k_2) \in L_1 \times L_2\). This proves that \((L_1 \times L_2, \leq, *, e, -o)\) is a partially ordered monoid. We need to prove that the internal-hom defined above is right adjoint to the monoidal product.

\[
\begin{align*}
b * c & \leq a \\
\iff (b_1 * c_1, b_2 * c_2) & \leq (a_1, a_2) \\
\iff b_1 * c_1 & \leq_1 a_1 \land b_2 * c_2 \leq_2 a_2 \\
\iff b_1 & \leq_1 c_1 \rightarrow_1 a_1 \land b_2 \leq_2 c_2 \rightarrow_2 a_2 \\
\iff (b_1, b_2) & \leq (c_1 \rightarrow_1 a_1, c_2 \rightarrow_2 a_2) \\
\iff b & \leq c \rightarrow a
\end{align*}
\]

This proves that the product of lineales is again a lineale.

Finally, we check the universal property. The projections of the underlying sets are also strict morphisms of lineales, as the structure is defined component-wise. For two morphisms of lineales \(g: (L, \sqsupseteq, *, e, -o) \rightarrow (L_1, \sqsupseteq_1, *_1, e_1, -o_1)\) and \(h: (L, \sqsupseteq, *, e, -o) \rightarrow (L_2, \sqsupseteq_2, *_2, e_2, -o_2)\), there is a unique function \(\langle g, h \rangle: L \rightarrow L_1 \times L_2\) commuting with the projections by the universal property of the product in \(\text{Set}\). This function also preserves the order and the monoidal structure as it does so component-wise. \(\square\)
3. The category $\mathcal{M}_L\text{Set}$

Having defined a lineale, we proceed to construct the intermediate category $\mathcal{M}_L\text{Set}$ over which our category of Petri nets $\text{Net}_L$ is built.

Its objects $(U, X, \alpha)$ represent $L$-valued relations $\alpha$ between the sets $U$ and $X$. The game semantics interpretation of $\alpha$ is given by the game played on the proposition $\bigvee_{u \in U} \bigwedge_{x \in X} \alpha(u, x)$: Proponent tries to prove $\alpha$ and chooses an element $u \in U$, then Opponent tries to refute it and chooses an element $x \in X$; the result of the game is determined by the value of $\alpha(u, x)$. In the classical case, $L = \{0, 1\}$, Proponent wins if $\alpha$ is true on the chosen pair $(u, x)$, $\alpha(u, x) = 1$. In the other cases, we may imagine that the possible outcomes of the game are not just “win” and “loose”, but they carry the extra information given by the lineale.

Morphisms need to carry the information about both the players moves [Bla92]. Thus, morphisms $(U, X, \alpha) \to (V, Y, \beta)$ are pairs of functions, the first one $f: U \to V$ that maps every choice of Proponent in $\alpha$ to one in $\beta$ and the second one $F: Y \to X$ that maps every choice of Opponent in $\beta$ to one in $\alpha$. Suppose that Proponent has a winning strategy in $\alpha$, i.e. there is $u \in U$ such that, for all $x \in X$, $\alpha(u, x)$ is true. Then, Proponent can choose $f(u)$ in $\beta$ and we know that each choice $y \in Y$ of Opponent will lead to an outcome $\beta(f(u), y)$ that is bounded by $\alpha(u, F(y))$.

These morphisms make $\mathcal{M}_L\text{Set}$ a category (Proposition 3.1).

**Definition 3.1. (Category $\mathcal{M}_L\text{Set}$)**

Given a lineale $(L, \sqsupseteq, *, e, \triangleright)$, the category $\mathcal{M}_L\text{Set}$ is defined by the following data.

- An object is a triple $A = (U, X, \alpha)$, denoted by $U \xleftarrow{\alpha} X$, where $U$, $X$ are sets and $\alpha: U \times X \to L$ is a function in $\text{Set}$.

- A morphism $(f, F): (U \xleftarrow{\alpha} X) \to (V \xleftarrow{\beta} Y)$ is a pair of morphisms, $f: U \to V$ and $F: Y \to X$ in $\text{Set}$, such that $\forall u \in U \forall y \in Y \alpha(u, Fy) \sqsupseteq \beta(fu, y)$.

$$
\begin{array}{ccc}
U \times Y & \xrightarrow{f \times 1_Y} & V \times Y \\
\downarrow 1_{U \times F} & & \downarrow \beta \\
U \times X & \xrightarrow{\alpha} & L 
\end{array}
$$

The category $\mathcal{M}_L\text{Set}$ allows us to have $L$-valued relations, including multirelations ($L = \mathbb{N}$) and any other label set that can be seen as a lineale. The objects of $\mathcal{M}_L\text{Set}$ can be seen as $L$-enriched profunctors [Bor94, Section 7.7], also known as distributors, where the lineale $L$ is seen as a one-object monoidal closed category. Their morphisms are, usually, natural transformations, while here we consider lax ones.
Example 3.1. For three elements $a, a', b \in L$ of a lineale $(L, \sqsubseteq, *, e, \circ)$ and sets $U = \{u, u'\}$, $X = \{x, x'\}$, $V = \{v\}$ and $Y = \{y\}$, we can define two objects $(U \leftarrow X)$ and $(V \leftarrow Y)$ of $M_L\text{Set}$.

\[
\begin{align*}
\alpha(u, x) &= a \\
\alpha(u, x') &= a' \\
\beta(v, y) &= b
\end{align*}
\]

These can be represented as weighted bipartite graphs where the elements of $U$ and $X$ are the vertices and are connected by an edge with weight $a \in L$ whenever the value of $\alpha$ on them is $a$ (Figure 3).

Figure 3: Two morphisms in $M_L\text{Set}$, if $a, a' \sqsubseteq b$ (left) and if $b \sqsubseteq a, e$ (right).

(f, F): $(U \leftarrow X) \to (V \leftarrow Y)$ and, if $b \sqsubseteq a, e$, there is a morphism $(g, G): (V \leftarrow Y) \to (U \leftarrow X)$.

\[
\begin{align*}
f(u) &= f(u') = v \\
g(v) &= u \\
F(y) &= x \\
G(x) &= G(x') = y
\end{align*}
\]

Proposition 3.1. $M_L\text{Set}$ is a category.

Proof:
The identity arrow of an object $U \leftarrow X$ in $M_L\text{Set}$ is given by the pair $(1_U, 1_X)$ of identities in $\text{Set}$. Moreover, given objects $A = (U \leftarrow X)$, $B = (V \leftarrow Y)$, and $C = (W \leftarrow Z)$, and morphisms $(f, F): A \to B$ and $(g, G): B \to C$, their composition is computed componentwise as $(g, G) \circ (f, F) = (g \circ f, F \circ G): A \to C$. Notice that $(g, G) \circ (f, F)$ is a morphism in $M_L\text{Set}$: given $u \in U$ and $z \in Z$, we have $\alpha(u, FGz) \sqsubseteq \beta(fu, Gz) \sqsubseteq \gamma(gfu, z)$. Associativity and unitality come from associativity and unitality in $\text{Set}$.

We now proceed to define products and coproducts in $M_L\text{Set}$ and to equip it with a symmetric monoidal closed structure. To achieve this, we lift the symmetric monoidal closed structure of the lineale: the product and coproduct rely only on the set structure, the tensor product is defined using the monoidal structure of the lineale, and the internal-hom in $M_L\text{Set}$ is derived from the internal-hom of the lineale.
The intuitions from the game semantics for linear logic justify the types of the product and coproduct in the Dialectica construction [Bla92]. For Proponent to win a game \( A \& B \), it needs to have a winning strategy both in \( A \) and in \( B \), while Opponent only needs to choose one between \( A \) and \( B \), and have a winning strategy there. Thus, in the Proponent side, the product of \( A \) and \( B \) is the categorical product \( U \times V \), while in the Opponent side, it is the coproduct \( X + Y \). Similarly, for Proponent to win a game \( A \oplus B \), it needs to choose one between \( A \) and \( B \), and have a winning strategy there, while Opponent needs to have a winning strategy both in \( A \) and in \( B \). This is reflected in the types of the coproduct, \( U + V \) for the Proponent side and \( X \times Y \) for the Opponent side.

**Definition 3.2. (Product and coproduct in \( M_{LSet} \))**

Given two objects \( A = (U \xrightarrow{\alpha} X) \) and \( B = (V \xleftarrow{\beta} Y) \) in \( M_{LSet} \), we define their cartesian product \( A \& B \) and their coproduct \( A \oplus B \) as the following objects.

\[
A \& B = (U \times V \xrightarrow{\alpha \& \beta} X + Y) \quad \quad A \oplus B = (U + V \xleftarrow{\alpha \oplus \beta} X \times Y)
\]

The function \( \alpha \& \beta \) is \( U \times V \times (X + Y) \xrightarrow{[\alpha \& \epsilon_U, \beta \& \epsilon_U]} L \), where \( \epsilon_U : U \times V \times Y \rightarrow V \times Y \) is the function that discards \( U \) and \( \epsilon_V : U \times V \times X \rightarrow U \times X \) the function that discards \( V \) in \( \text{Set} \). Similarly, the function \( \alpha \oplus \beta \) is \( (U + V) \times X \times Y \xrightarrow{[\alpha \oplus \epsilon_Y, \beta \oplus \epsilon_X]} L \), where \( \epsilon_Y : U \times X \times Y \rightarrow U \times X \) is the function that discards \( Y \) and \( \epsilon_X : V \times X \times Y \rightarrow V \times Y \) the function that discards \( X \) in \( \text{Set} \).

**Proposition 3.2.** The operations in Definition 3.2 give products and coproducts in \( M_{LSet} \).

**Proof:**

The projections \( \pi_A : A \& B \rightarrow A \) and \( \pi_B : A \& B \rightarrow B \) are defined by projections and coprojections in \( \text{Set} \): \( \pi_A = (\pi_U, \iota_X) \) and \( \pi_B = (\pi_V, \iota_Y) \).

Let \( (f, F) : C \rightarrow A \) and \( (g, G) : C \rightarrow B \) be two morphisms in \( M_{LSet} \). We check that the pair \( ((f, g), [F, G]) \) is a morphism of type \( C \rightarrow A \& B \), using the properties of products and coproducts in \( \text{Set} \) and the definition of morphism in \( M_{LSet} \).

\[
(\alpha \& \beta) \circ ((f, g) \times \mathbb{1}_{X+Y}) \\
= [\alpha, \beta] \circ (((f, \epsilon_V \circ g) \times \mathbb{1}_X) + ((\epsilon_U \circ f, g) \times \mathbb{1}_Y)) \\
= [\alpha, \beta] \circ ((f \times \mathbb{1}_X) + (g \times \mathbb{1}_Y)) \\
= [\alpha \circ (f \times \mathbb{1}_X), \beta \circ (g \times \mathbb{1}_Y)] \\
\subseteq [\gamma \circ (\mathbb{1}_W \times F), \gamma \circ (\mathbb{1}_W \times G)] \\
= \gamma \circ (\mathbb{1}_W \times [F, G])
\]

The projections composed with \( ((f, g), [F, G]) \) give back \( (f, F) \) and \( (g, G) \).

\[
\pi_A \circ ((f, g), [F, G]) = (\pi_U \circ (f, g), [F, G] \circ \iota_X) = (f, F) \\
\pi_B \circ ((f, g), [F, G]) = (\pi_V \circ (f, g), [F, G] \circ \iota_Y) = (g, G)
\]
This is the unique morphism doing so: any morphism \((h, H): C \to A \& B\) that commutes with the projections needs to be \(((f, g), [F, G])\) by the universal property of the product and coproduct in \(\text{Set}\). This shows that \(\&\) is the categorical product in \(M_L\text{Set}\).

Dually, we show that \(\oplus\) is the categorical coproduct. The projections \(\iota_A: A \to A \oplus B\) and \(\iota_B: B \to A \oplus B\) are defined by projections and coprojections in \(\text{Set}\): \(\iota_A = (\iota_U, \pi_X)\) and \(\iota_B = (\iota_V, \pi_Y)\).

Let \((f, F): A \to C\) and \((g, G): B \to C\) be two morphisms in \(M_L\text{Set}\). We check that the pair \(((f, g), (F, G))\) is a morphism of type \(A \oplus B \to C\), using the properties of products and coproducts in \(\text{Set}\) and the definition of morphism in \(M_L\text{Set}\).

\[
\begin{align*}
(\alpha \oplus \beta) \circ (1_U + V \times (F, G)) &= [\alpha, \beta] \circ ((1_U \times (F, \epsilon_Y \circ G)) + (1_V \times (\epsilon_X \circ F, G))) \\
&= [\alpha, \beta] \circ ((1_U \times F) + (1_V \times G)) \\
&= [\alpha \circ (1_U \times F), \beta \circ (1_V \times G)] \\
&\cong [\gamma \circ (f \times 1_Z), \gamma \circ (g \times 1_Z)] \\
&= \gamma \circ ([f, g] \times 1_Z)
\end{align*}
\]

The coprojections composed with \(((f, g), (F, G))\) give back \((f, F)\) and \((g, G)\).

\[
\begin{align*}
([f, g], (F, G)) \circ \iota_A &= (f, F) \\
([f, g], (F, G)) \circ \iota_B &= (g, G)
\end{align*}
\]

This is the unique morphism doing so: any morphism \((h, H): A \oplus B \to C\) that commutes with the coprojections needs to be \(((f, g), (F, G))\) by the universal property of the product and coproduct in \(\text{Set}\). \(\Box\)

**Example 3.2.** The product and coproduct of the objects \(A = (U \overset{\alpha}{\leftarrow} X)\) and \(B = (V \overset{\beta}{\leftarrow} Y)\) defined in Example 3.1 are below and in Figure 4.

\[
\begin{align*}
\alpha \& \beta(u, v, x) &= a \\
\alpha \& \beta(u', v, x) &= a' \\
\alpha \& \beta(u, v, x') &= e \\
\alpha \& \beta(u', v, x') &= e \\
\alpha \& \beta(u, v, y) &= b \\
\alpha \& \beta(u', v, y) &= b \\
\alpha \beta(u, v, x) &= \alpha \beta(x, v, y) = a
\end{align*}
\]

The terminal object is \((1 \overset{0}{\leftarrow} \emptyset)\), with one element on the left, while the initial one is \((\emptyset \overset{0}{\leftarrow} 1)\), with one element on the right.
Now, we use the monoidal operation of \( L \) to define a monoidal product in \( \mathcal{M}_L \text{Set} \). This operation can be viewed as a weaker form of “and” [Bla92]. A winning strategy for Proponent in \( A \otimes B \) is a winning strategy in both \( A \) and \( B \), but, contrary to the categorical product, a winning strategy for Opponent in \( A \otimes B \) consists of a winning strategy both in \( A \) and in \( B \), for every choice of \( u \in U \) and \( v \in V \) by Proponent. The types of \( A \otimes B \) reflect this dynamic: \( U \times V \) for the Proponent side and \( X^V \times Y^U \) for the Opponent side.

**Definition 3.3. (Monoidal product in \( \mathcal{M}_L \text{Set} \))**

Given two objects \( A = (U \xleftarrow{\alpha} X) \) and \( B = (V \xleftarrow{\beta} Y) \) in \( \mathcal{M}_L \text{Set} \), we define their monoidal product \( A \otimes B \) as the following object.

\[
A \otimes B = (U \times V \xleftarrow{\alpha \otimes \beta} X^V \times Y^U)
\]

Where \( X^V \) and \( Y^U \) are internal-hom objects in \( \text{Set} \) and the function \( \alpha \otimes \beta \) is defined by the following composition.

\[
U \times V \times X^V \times Y^U \xrightarrow{\Delta_{U \times V} \times 1_{X^V \times Y^U}} U \times V \times U \times V \times X^V \times Y^U \xrightarrow{1_{U \times V} \times \sigma_{U,V,X^V \times Y^U}} U \times V \times X \times Y \xrightarrow{\alpha \times \beta} L \times L \xrightarrow{\ast} L
\]

where \( \Delta_{U \times V} \) is the diagonal map on \( U \times V \), \( \sigma_{U,V,X^V} \) is a permutation, and \( \text{eval}_{U} \) and \( \text{eval}_{V} \) are the evaluation maps in \( \text{Set} \). Spelling out this definition element wise, we obtain \((\alpha \otimes \beta)(u,v,f,g) = \alpha(u,fv) \ast \beta(v,gu)\).

On morphisms \((f,F) : A \to A' \) and \((g,G) : B \to B'\), we define the monoidal product as follows

\[
(f,F) \otimes (g,G) = (f \times g, F(-)g \times G(-)f) : A \otimes B \to A' \otimes B'
\]

where \( f \times g : U \times V \to U' \times V' \) and \( F(-)g \times G(-)f : X^V' \times Y'^U \to X^V \times Y^U \).

**Proposition 3.3.** The construction above induces a functor \( \otimes : \mathcal{M}_L \text{Set} \times \mathcal{M}_L \text{Set} \to \mathcal{M}_L \text{Set} \), which is a symmetric monoidal product on \( \mathcal{M}_L \text{Set} \).
Proof:
The object $A \otimes B = (U \times V \overset{\alpha \otimes \beta}{\longrightarrow} X^V \times Y^U)$ is clearly an object of $M_L \text{Set}$. The unit is the object $I = (1 \xleftarrow{\varepsilon} 1)$, which assigns to $1 \times 1$ the monoidal unit $e$ of $L$. We need to check that the monoidal product is well defined, which means that $(f, F) \otimes (g, G)$ satisfies the condition on morphisms.

$$\alpha \otimes \beta(u, v, (F(-)g \times G(-)f)(f', g'))$$

$$= \alpha \otimes \beta(u, v, Ff'g, Gg'f)$$

$$= \alpha(u, Ff'Gv) * \beta(v, Gg'fu)$$

$$\sqsubseteq \alpha'(fu, f'Gv) * \beta'(gv, g'fu)$$

$$= \alpha' \otimes \beta'(fu, gv, f', g')$$

$$= \alpha' \otimes \beta'((f \times g)(u, v), f', g')$$

The monoidal product is a functor as it preserves composition

$$((f', F') \circ (f, F)) \otimes ((g', G') \circ (g, G))$$

$$= (f' \circ f, F \circ F') \otimes (g' \circ g, G \circ G')$$

$$= ((f' \circ f) \times (g' \circ g), F'F'(-)g'g \times GG'(-)f'f)$$

$$= ((f' \times g') \circ (f \times g), (F(-)g \times G(-)f) \circ (F'(-)g \times G'(-)f'))$$

$$= (f' \times g', F'(-)g' \times G'(-)f') \circ (f \times g, F(-)g \times G(-)f)$$

$$= ((f', F') \otimes (g', G')) \circ ((f, F) \otimes (g, G))$$

and identities

$$(1_U, 1_X) \otimes (1_V, 1_Y)$$

$$= (1_U \times 1_V, 1_X(-)1_V \times 1_Y(-)1_U)$$

$$= (1_{U \times X}, 1_{X^V \times Y^U})$$

The associator is defined by the following isomorphisms in $\text{Set}$

$$\alpha_{A,B,C} = (\alpha_{U,V,W}, A_{X,Y,Z}) : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$$

where $\alpha_{U,V,W} : (U \times V) \times W \to U \times (V \times W)$ is the associator in $\text{Set}$ with the cartesian product and $A_{X,Y,Z} : X^V \times W \times (Y^W \times Z^V)^U \to (X^V \times Y^U)^W \times Z^U \times V$ is the composition of isomorphisms in $\text{Set}$ given by

$$X^V \times W \times (Y^W \times Z^V)^U \xrightarrow{\cong} X^V \times W \times (Y^W \times Z^U \times V)$$

$$\xrightarrow{\cong} (X^V \times W \times Y^U \times W) \times Z^U \times V \xrightarrow{\cong} (X^V \times Y^U)^W \times Z^U \times V$$

The unitors are defined by the following isomorphisms in $\text{Set}$

$$\lambda_A = (\lambda_U, L_X) : I \otimes A \to A$$

$$\rho_A = (\rho_U, R_X) : A \otimes I \to A$$
where $\lambda_U : 1 \times U \to U$ and $\rho_U : U \times 1 \to U$ are the unitors in Set, and $L_X : X \to 1^U \times X^1$ and $R_X : X \to X^1 \times 1^U$ are the compositions of isomorphisms in Set given by
\[
X \xrightarrow{\cong} 1 \times X \xrightarrow{\cong} 1^U \times X^1 \quad \quad \quad X \xrightarrow{\cong} X \times 1 \xrightarrow{\cong} X^1 \times 1^U
\]

We are left to prove that the above are actually morphisms in $\mathcal{M}_L \text{Set}$, that they are natural isomorphisms and that they satisfy the pentagon and triangle equations [Mac71]. The associator is a morphism because for all $((u, v), w) \in (U \times V) \times W$ and all $(f, (g, h)) \in X^{V \times W} \times (Y^W \times Z^V)^U$
\[
((\alpha \otimes \beta) \otimes \gamma)(((u, v), w), A_{X,Y,Z}(f, (g, h)))
= ((\alpha \otimes \beta) \otimes \gamma)(((u, v), w), ((f, g), h))
= (\alpha(u, f(v, w)) \ast \beta(v, g(u, w))) \ast \gamma(w, h(u, v))
= \alpha(u, f(v, w)) \ast (\beta(v, g(u, w)) \ast \gamma(w, h(u, v)))
= (\alpha \otimes (\beta \otimes \gamma))((u, (v, w)), (f, (g, h)))
= (\alpha \otimes (\beta \otimes \gamma))(\alpha_{U,V,W}((u, v), w), (f, (g, h)))
\]
The unitors are morphisms because for all $u \in U$ and all $x \in X$
\[
(I \otimes \alpha)(((*, u), L_X(x))
= (I \otimes \alpha)(((*, u), (*, x))
= I((*, u) \ast \alpha(u, x)
= e \ast \alpha(u, x)
= \alpha(u, x)
\]
The associator and the unitors are natural isomorphisms because they are natural isomorphisms component wise. The triangle and pentagon equations hold because they hold in Set.

The symmetries $\sigma_{A,B} = (\sigma_{U,V}, \sigma_{Y_{V,X}Y}) : A \otimes B \to B \otimes A$ are well-defined because their components are isomorphisms and they are natural because their components are natural in Set. Finally, the hexagon equations hold because they also hold in Set with the cartesian product. \hfill \Box

**Definition 3.4. (Internal-hom in $\mathcal{M}_L \text{Set}$)**

Given two objects $A = (U \xleftrightarrow{\alpha} X)$ and $B = (V \xleftrightarrow{\beta} Y)$ in $\mathcal{M}_L \text{Set}$ we define their internal-hom, $[A, B]$, as follows:

\[
[A, B] = V^U \times X^Y \xleftarrow{[\alpha, \beta]} U \times Y
\]
The function $[\alpha, \beta]$ is defined by the following composition.
\[
V^U \times X^Y \times U \times Y \xrightarrow{1_{V^U \times X^Y} \times \Delta_{U \times Y}} V^U \times X^Y \times U \times Y \times U \times Y \xrightarrow{\sigma}
\]
\[
U \times Y \times X^Y \times U \times V^U \times Y \xrightarrow{1_{U \times Y} \times \text{eval} \times \text{eval} \times 1_Y} U \times X \times V \times Y \xrightarrow{\alpha \times \beta} L \times L \xrightarrow{\circ} L
\]
Spelling out this definition element wise, we obtain $[\alpha, \beta](f, F, u, y) = \alpha(u, Fy) \circ \beta(fu, y)$. 

E. Di L avore, W. Leal, V. de Paiva / Dialectica Petri Nets
Example 3.3. The monoidal product and internal-hom of the objects \((U \triangleleft X)\) and \((V \triangleleft Y)\) defined in Example 3.1 are below and in Figure 5. The sets \(X^V \cong \{x, x'\}\), \(Y^U \cong \{!\}\), \(V^U \cong \{!\}\) and \(X^Y \cong \{x, x'\}\) contain at most two elements.

\[
\begin{align*}
\alpha \otimes \beta(u, v, x, !) &= a \ast b & [\alpha, \beta](!, u, y) &= a \circ b \\
\alpha \otimes \beta(u', v, x, !) &= a' \ast b & [\alpha, \beta](!, x, u, y) &= b \\
\alpha \otimes \beta(u, v, x', !) &= b & [\alpha, \beta](!, x, u', y) &= a' \circ b \\
\alpha \otimes \beta(u', v, x', !) &= b & [\alpha, \beta](!, x', u', y) &= b
\end{align*}
\]

The monoidal unit is the object \(1 \triangleleft 1\), with two elements that are not related to each other, i.e. related with unit weight \(e\).

Proposition 3.4. The construction above induces an internal-hom functor \([-,-]: M_L\text{Set}^{op} \times M_L\text{Set} \to M_L\text{Set}\).

Proof:

The object \([A, B] = V^U \times X^Y \triangleleft U \times Y\) is clearly an object of \(M_L\text{Set}\). On morphisms \((f, F): A' \to A\) and \((g, G): B \to B'\) in \(M_L\text{Set}\), the internal-hom can be defined as

\[
[(f, F), (g, G)] = (g(-) f \times F(-) G, f \times G): [A, B] \to [A', B']
\]

where \(g(-) f \times F(-) G: V^U \times X^Y \to V'^{U'} \times X'^{Y'}\) and \(f \times G: U' \times Y' \to U \times Y\). We need to check that the internal-hom is well defined, which means that \([(f, F), (g, G)]\) needs to satisfy the condition on morphisms. For all \((h, H) \in V^U \times X^Y\) and all \((u', y') \in U' \times Y'\)

\[
\begin{align*}
[\alpha, \beta](h, H, (f \times G)(u', y')) &= [\alpha, \beta](h, H, fu', Gy') \\
&= \alpha fu', HGy') \circ \beta(hfu', Gy') \\
&\equiv \alpha'(u', FHGy') \circ \beta'(ghfu', y') \\
&= [\alpha', \beta'](ghf, FHG, u', y') \\
&= [\alpha', \beta'][(g(-) f \times F(-) G)(h, H), u', y')
\end{align*}
\]
because $\alpha'(u', FHGY') \sqsupseteq \alpha(fu', HGY')$ and $\beta(hfu', GY') \sqsupseteq \beta'(gfhu', y')$ as $(f, F)$ and $(g, G)$ are morphisms. The internal-hom is a functor as it preserves composition

$$[(f', F') \circ (f, F), (g', G') \circ (g, G)]$$

$$= [(ff', F'F), (gf', GG')]$$

$$= (gf'(-)ff' \times F'F(-)GG', ff' \times GG')$$

$$= ((g'(-)f' \times F'(-)G') \circ (g(-)f \times F(-)G), (f \times G) \circ (f' \times G'))$$

$$= (g'(-)f' \times F'(-)G', f' \times G') \circ (g(-)f \times F(-)G, f \times G)$$

$$= [(f', F'), (g', G')] \circ [(f, F), (g, G)]$$

and identities

$$[1_A, 1_B]$$

$$= [(1_U, 1_X), (1_V, 1_Y)]$$

$$= (1_V(-)1_U \times 1_X(-)1_Y, 1_U \times 1_Y)$$

$$= (1_VU \times 1_Y, 1_U \times 1_Y)$$

$$= 1_{[A,B]}$$

\[\blacksquare\]

The next result combines the structure on $M_L Set$ defined so far. Note how products and coproducts come from products and coproducts in $Set$, while the monoidal closed structure is lifted from the corresponding structure in the lineale $L$.

**Theorem 3.1.** The category $M_L Set$ has products and coproducts as in Definition 3.2 and is a symmetric monoidal closed category with monoidal product as in Definition 3.3 and internal-hom as in Definition 3.4.

**Proof:**

To prove the adjunction $- \otimes B \vdash [B, -]$ we have to show that, for every objects $A$ and $C$ in $M_L Set$, there is a bijection $\psi_{A,C} : \text{Hom}_{M_L Set}(A \otimes B, C) \cong \text{Hom}_{M_L Set}(A, [B, C])$ that is natural in $A$ and $C$.

Let $\phi_{U,Z} : \text{Hom}_{Set}(U \times V, Z) \rightarrow \text{Hom}_{Set}(U, Z^V)$ be the natural isomorphism witnessing the adjunction between the cartesian product and the internal-hom in $Set$ and let $\sigma_{U,V} : U \times V \rightarrow V \times U$ be the symmetry of the cartesian product in $Set$. Define the maps

$$\psi_{A,C}(f, F) = ((\phi(f), \phi(\phi^{-1}(F_2) \circ \sigma_{U,Z})), \phi^{-1}(F_1) \circ \sigma_{V,Z})$$

$$\psi_{A,C}^{-1}(g, G) = (\phi^{-1}(g_1), (\phi(G \circ \sigma_{Z,V}), \phi(\phi^{-1}(g_2) \circ \sigma_{Z,U})))$$

We can check that they are well defined. An element of $\text{Hom}_{M_L Set}(A \otimes B, C)$ is a pair $(f, \langle F_1, F_2 \rangle)$ with $f : U \times V \rightarrow W$ and $F = \langle F_1, F_2 \rangle : Z \rightarrow X^V \times Y^U$ such that $\forall (u, v) \in U \times V \forall z \in Z (\alpha \otimes \beta)(u, v, Fz) \sqsubseteq \gamma(f(u, v), z)$, which is equivalent to $\alpha(u, (F_1(z))(v)) \ast \beta(v, (F_2(z))(u)) \sqsubseteq \gamma(f(u, v), z)$. On the other hand, an element of $\text{Hom}_{M_L Set}(A, [B, C])$ is a pair $(\langle g_1, g_2 \rangle, G)$ with
Definition 4.1. (Category \text{Net}_L)

Given a lineale \( M \) together two copies of \( M \) relations will be valued in a generic lineale \( L \). The category \( \text{Net}_L \) is given by a set of places \( A \) Petri net is given by a set of places \( A \), a set of transitions \( X \), and has two relations between these two sets that specify the precondition relation \( \alpha^\triangleright \) and the postcondition relation \( \alpha^\bowtie \). In our case these relations will be valued in a generic lineale \( L \) and the pre- and post- conditions will be objects in \( \text{M}_L \text{Set} \). The category of Petri nets that we consider has Petri nets as objects and is obtained by putting together two copies of \( \text{M}_L \text{Set} \) by taking a pullback in \( \text{Cat} \): the first copy represents preconditions \( \alpha^\triangleright : U \times X \to \text{L} \) and the other one represents postconditions \( \alpha^\bowtie : U \times X \to \text{L} \).

Definition 4.1. (Category \text{Net}_L)

Given a lineale \((L, \sqsubseteq, \ast, e, \circ)\), the category \( \text{Net}_L \) is defined by the following data.

- An object is a pair \( A = (\triangleright A, \bowtie A) \) of objects \( U \leftarrow \triangleright A \leftrightarrow X \) and \( U \leftarrow \bowtie A \leftrightarrow X \) in \( \text{M}_L \text{Set} \).
- A morphism \((f, F) : (\triangleright A, \bowtie A) \to (\triangleright B, \bowtie B)\) is a morphism both \((f, F) : \triangleright A \to \triangleright B\) and \((f, F) : \bowtie A \to \bowtie B\) in \( \text{M}_L \text{Set} \).

Remark 4.1. Definition 4.1 can be restated more abstractly. The category \( \text{Net}_L \) is the pullback in \( \text{Cat} \) of the forgetful functor \( \mathcal{U}_L : \text{M}_L \text{Set} \to \text{Set} \times \text{Set}^{op} \) along itself. The functor \( \mathcal{U}_L \) assigns to an object \((U, X, \alpha)\) the pair of sets \((U, X)\), and to a morphism \((f, F)\) the corresponding pair of functions \((f, F)\).

Example 4.1. The relations in Example 3.1 may represent the precondition relations, \((U \leftarrow \triangleright X)\) and \((V \leftarrow \bowtie Y)\), of two nets \( A \) and \( B \). We define below two other relations, \((U \leftarrow \bowtie X)\) and \((V \leftarrow \triangleright Y)\), that specify the postconditions of the nets \( A \) and \( B \). Figure 6 shows the net \( A \) on the top left and
bottom right, and the net \( B \) on the bottom left and top right.

\[
\begin{align*}
\preceq \alpha(u, x) &= a \\
\preceq \alpha(u', x) &= a' \\
\preceq \alpha(u, x') &= e \\
\preceq \alpha(u', x') &= e \\
\preceq \beta(v, y) &= b \\
\preceq \beta(v, y) &= e
\end{align*}
\]

The morphism \((f, F)\) defined in Example 3.1 is also a morphism \((f, F) : A \to B\) of nets because it is a morphism \((f, F) : (U \xrightarrow{\preceq a} X) \to (V \xrightarrow{\preceq b} Y)\), see Figure 6, left. The morphism \((g, G)\) in the same example is a morphism of nets whenever \(e \sqsupseteq c\), see Figure 6, right. This condition, for the lineales like \(2, 3\) or \(\mathbb{N}\) where \(e\) is the smallest element, is satisfied only for \(c = e\) so the elements \(u\) and \(x'\) cannot be related.

![Figure 6](image.png)

Figure 6: Two net morphisms, if \(a \sqsupseteq b\) (left), and if \(b \sqsupseteq a, a'\) and \(e \sqsupseteq c\) (right).

The structure of \(M_L\text{-}Set\) defines analogous structure in \(\text{Net}_L\).

**Definition 4.2. (Structure of \(\text{Net}_L\))**

The category \(\text{Net}_L\) inherits the structure of \(M_L\text{-}Set\). All the connectives are defined componentwise:

- \(A \otimes B = (\preceq A \otimes \preceq B, A^\circ \otimes B^\circ)\).
- \([A, B] = (\preceq A, \preceq B), [A^\circ, B^\circ]).\)
- \(A \& B = (\preceq A \& \preceq B, A^\circ \& B^\circ)\).
- \(A \oplus B = (\preceq A \oplus \preceq B, A^\circ \oplus B^\circ)\).

Examples of Petri nets modelled in this category are in the next section, where we will show how, with the possibility of changing the lineale, we can encompass different kinds of nets.

**Example 4.2.** Figure 7 shows the product and coproduct of the nets defined in Example 4.1. These are obtained by computing their product and coproduct component-wise, as shown in Example 3.2.
Figure 7: Product (left) and coproduct (right) of the nets $A$ and $B$.

Figure 8 shows the terminal and initial nets, which are the units for the product and coproduct respectively.

![Terminal and Initial Nets](image)

Figure 8: Terminal net (left), initial net (centre) and unit net (right).

**Example 4.3.** Figure 9 shows the monoidal product and internal-hom of the nets defined in Example 4.1. These are obtained by computing their monoidal product and internal-hom component-wise, as shown in Example 3.3. Figure 8 shows the monoidal unit net.

![Monoidal Product and Internal-hom](image)

Figure 9: Monoidal product (left) and internal-hom (right) of the nets $A$ and $B$. 
5. Changing the logic of Petri nets

This section studies in detail the examples mentioned in Section 2.3 to showcase the different interpretations of the arcs that we can achieve by just changing the lineale $L$ in the construction of the category $\text{Net}_L$. Section 5.7 concludes by making the constructions in Section 3 and Section 4 functorial: for a morphism of lineales $h : L \to L'$, specifying how to change the logic, we construct a functor $\mathcal{M}_L \text{Set} \to \mathcal{M}_{L'} \text{Set}$ that preserves the structure of these categories.

5.1. Elementary Petri nets: $L = 2$

By considering relations with values on this lineale, which correspond to ordinary relations, we obtain the elementary Petri nets, those where pre- and post-conditions only say whether or not a place is a pre- or post-condition for a transition [RE96].

5.2. Kleene Petri nets: $L = 3$

Thanks to Example 2.2, we can define the Dialectica construction over $(3, \leq, *, 1, \to)$ and Petri nets with weights in 3 accordingly.

We take as motivating example the model of the chemical reactions regulating the circadian clock of *Synechococcus Elongatus* [HLE+09] that is composed of two successive phosphorylations and two successive dephosphorilations (which are the transitions labelled with “p” and “d”, respectively, in Figure 10). There is experimental evidence [ALH07] for the existence of further feedback loops in this model. However, the precise underlying mechanism is still unknown. We can take into account these unknowns in our model for Petri nets by adding arcs with $u$ weight (presence and absence are represented by 1 and 0 respectively). The Petri net in Figure 10 shows the values of the pre- and post-conditions relations as weights on the arcs. Note that the arcs not shown are those with value 0.

5.3. Multirelation Petri nets: $L = \mathbb{N}$

As every chemical reaction, the one to obtain water from oxygen and hydrogen needs stoichiometric coefficients to be represented properly. Multirelations take these into account, as shown in Figure 11.

5.4. Integers Petri nets: $L = \mathbb{Z}$

Empirical systems often need to locally reverse the logic of preconditions to express that the presence of tokens in a given place “disables” a transition. Several different concepts of inhibitor arcs can be modelled by Petri nets including the “threshold inhibitor arc”. Reaction inhibitors in chemistry illustrate the situation: in Figure 12 chemical reaction $r$ will not take place if the amount of substance I exceeds 3, a condition that is expressed by its inverse $-3$.

5.5. Probabilistic Petri nets: $L = [0, 1]$

The SIR (Susceptible, Infectious, Recovered) model is a simple compartmental model for infectious diseases. A susceptible individual has a contact with an infectious individual with probability $p_c$ and,
after the contact, it can be infected with probability $p_I$, or remain susceptible with probability $1 - p_I$. On the other hand, an infectious individual can recover with probability $p_R$ or remain infectious with probability $1 - p_R$. This setting can be represented with a Petri net where the relations between places and transitions are valued in $[0, 1]$ (Figure 13).
5.6. Product of lineales

There is a dual situation to inhibition in chemistry, namely, catalysis. A catalyst is a substance that increases the reaction rate without being consumed by the reaction. The presence of a substance $S$ in a chemical reaction might then play one of three roles: reactant/product, inhibitor, or catalyst. We claim that the product of the lineales $\mathbb{R}^+$ and $\mathbb{Z}$ has enough expressive power to model reaction rates in the presence of both inhibitors and catalysts. In Figure 14 pairs of the form $(r, 0)$, state that those substances are not inhibitors nor catalysts, and $r$ is the rate at which a substance is consumed or produced. The negative number in the label $(r_4, -3)$ expresses that $I$ is an inhibitor of reaction $r$, and -3 the minimum amount of $I$ required to slow down the reaction by the rate $r_4$. Finally, the label $(r_5, 5)$ indicates that $C$ is a catalyst and 5 is the minimum amount of $C$ required to increase the reaction rate by $r_5$.

5.7. Changing the logic, functorially

This section concludes by extending the constructions $\mathcal{M}_L \mathbf{Set}$ and $\mathcal{Net}_L$ to functors. Each morphism of lineales $h: L \to L'$ determines a functor $\mathcal{M}_L \mathbf{Set} \to \mathcal{M}_{L'} \mathbf{Set}$ that maps $L$-valued relations to $L'$-valued relations.
valued relations and, thus, changes their logic. Similarly, we obtain a functor $\text{Net}_L \to \text{Net}_{L'}$ that maps $L$-valued nets to $L'$-valued nets.

Consider the category $\text{LinCat}$ of symmetric monoidal closed categories with products and coproducts, where morphisms are lax monoidal functors preserving the products and coproducts. Theorem 3.1 and Proposition 3.2 show that, for every lineale $L$, $M_L \text{Set}$ is an object of $\text{LinCat}$. The next proposition shows that this construction extends to a functor preserving the closed monoidal structure, products and coproducts.

**Proposition 5.1.** There is a functor $M_{(-)} \text{Set} : \text{Lin} \to \text{LinCat}$ defined on objects $L$ by the construction $M_L \text{Set}$ in Section 3.

**Proof:**
For a morphism of lineales $h : (L, \sqsubseteq, *, e, -\circ) \to (L', \sqsubseteq', *, e', -\circ')$, define the action of $M_{(-)} \text{Set}$ on it as a functor $M_h \text{Set} : M_L \text{Set} \to M_{L'} \text{Set}$. On an object $A = U \sqsubseteq X$, it acts by precomposing with $h : M_h \text{Set}(U \sqsubseteq X) = U \overset{h \circ \alpha}{\sqsubseteq} X$. On morphisms, it acts as the identity: $M_h \text{Set}(f, F) = (f, F)$. This is well-defined because $h$ is monotone:

$$h \circ \alpha \circ (1_U \times F) \sqsubseteq h \circ \beta \circ (f \times 1_Y).$$

$M_h \text{Set}$ is a functor as it trivially preserves compositions and identities. We check that it preserves products and coproducts.

\[
\begin{align*}
M_h \text{Set}(A \& B) &= U \times V \overset{h \circ (\alpha \& \beta)}{\sqsubseteq} X + Y \\
&= U \times V \overset{h \circ [\alpha \circ \lambda \& \beta \circ \rho]}{\sqsubseteq} X + Y \\
&= U \times V \overset{h \circ (\alpha \& \lambda \circ \beta \circ \rho)}{\sqsubseteq} X + Y \\
&= U \times V \overset{(h \alpha) \& (h \beta)}{\sqsubseteq} X + Y \\
&= M_h \text{Set}(A) \& M_h \text{Set}(B)
\end{align*}
\]

\[
\begin{align*}
M_h \text{Set}(A \oplus B) &= U + V \overset{h \circ (\alpha \& \beta)}{\sqsubseteq} X \times Y \\
&= U + V \overset{h \circ [\alpha \circ \lambda \& \beta \circ \rho]}{\sqsubseteq} X \times Y \\
&= U + V \overset{h \circ (\alpha \& \lambda \circ \beta \circ \rho)}{\sqsubseteq} X \times Y \\
&= U + V \overset{(h \alpha) \& (h \beta)}{\sqsubseteq} X \times Y \\
&= M_h \text{Set}(A) \oplus M_h \text{Set}(B)
\end{align*}
\]

Finally, we show that $M_h \text{Set}$ is a lax monoidal functor. Since $h$ is a lax monoid homomorphism, the pair $\mu_{A,B} = (1_U \times V, 1_{X \times Y}) : M_h \text{Set}(A) \otimes M_h \text{Set}(B) \to M_h \text{Set}(A \otimes B)$ is a morphism in $M_L \text{Set}$.

\[
M_h \text{Set}(A) \otimes M_h \text{Set}(B) = U \times V \overset{(h \alpha) \& (h \beta)}{\sqsubseteq} X^V \times Y^U \sqsubseteq U \times V \overset{h \circ (\alpha \& \beta)}{\sqsubseteq} X^V \times Y^U = M_h \text{Set}(A \otimes B)
\]

Similarly, $\varepsilon = (1_1, 1_1) : I \to M_h \text{Set}(I)$ is a morphism in $M_L \text{Set}$.

\[
I = 1 \overset{\varepsilon'}{\sqsubseteq} 1 \sqsubseteq 1 \overset{h \circ e}{\sqsubseteq} M_h \text{Set}(I)
\]

Naturality, associativity and unitality of $\mu_{A,B}$ are easy to check as its components are identities and $M_h \text{Set}$ is the identity on morphisms. Finally, $M_h \text{Set}$ is symmetric because it is the identity on morphisms. \qed
Every category constructed by the functor in Proposition 5.1 determines a functor $U_L : M_L \text{Set} \to \text{Set} \times \text{Set}^{\text{op}}$ that forgets the relations $U \leftarrow X$ and only keeps the underlying sets $(U, X)$. Consider the subcategory $\text{DialCat}$ of $\text{LinCat}$ spanned by the functor $M(\_)$.

**Lemma 5.1.** There is a functor $U : \text{DialCat} \to \text{DialCat}/(\text{Set} \times \text{Set}^{\text{op}})$ that assigns the forgetful functor $U_L : M_L \text{Set} \to \text{Set} \times \text{Set}^{\text{op}}$ to each category $M_L \text{Set}$.

**Proof:**
For a functor $H : M_L \text{Set} \to M_L \text{Set}$, we define $U(H) = H$. This assignment is functorial as every such functor $H$ commutes with the forgetful functors: on objects $A = U \leftarrow X$, $U_L(H(A)) = (U, X) = U_L(A)$; on morphisms $(f, F) : U \leftarrow X \to V \leftarrow Y$, $U_L(H(f, F)) = (f, F) = U_L(f, F)$. $\square$

As mentioned in Remark 4.1, the category $\text{Net}_L$ is a pullback of the functor $U_L : M_L \text{Set} \to \text{Set} \times \text{Set}^{\text{op}}$ along itself. This construction is functorial.

**Proposition 5.2.** There is a functor $\text{Ker} : \text{Cat}/(\text{Set} \times \text{Set}^{\text{op}}) \to \text{Cat}$ that, for a functor $F : C \to \text{Set} \times \text{Set}^{\text{op}}$, constructs the pullback of $F$ along itself.

**Proof:**
For a functor $F : C \to \text{Set} \times \text{Set}^{\text{op}}$, construct the pullback $\hat{C}$ of $F$ along itself and define $\text{Ker}(F) = \hat{C}$. Let $F : C \to \text{Set} \times \text{Set}^{\text{op}}$ and $G : D \to \text{Set} \times \text{Set}^{\text{op}}$ be objects in $\text{Cat}/(\text{Set} \times \text{Set}^{\text{op}})$. For a functor $H : C \to D$ such that $G \circ H = F$, define $\text{Ker}(H)$ as the unique functor $\hat{C} \to \hat{D}$ given by the universal property of the pullback defining $\hat{D}$. This defines a functor as it is the composition of the diagonal and product functors in $\text{Cat}/(\text{Set} \times \text{Set}^{\text{op}})$ with the forgetful functor $\text{Cat}/(\text{Set} \times \text{Set}^{\text{op}}) \to \text{Cat}$. $\square$

**Corollary 5.1.** There is a functor $\text{Net}_{(\_)} : \text{Lin} \to \text{Cat}$ defined on objects $L$ by the construction $M_L \text{Set}$ in Section 4.

**Proof:**
The functor $\text{Net}_{(\_)}$ is the composition below of functors from Propositions 5.1 and 5.2 and Lemma 5.1.

$$\text{Net}_{(\_)} : \text{Lin} \xrightarrow{M_{(\_)} \text{Set}} \text{DialCat} \xrightarrow{U} \text{DialCat}/(\text{Set} \times \text{Set}^{\text{op}}) \xrightarrow{\text{Ker}} \text{Cat}$$

$\square$

**Example 5.1.** Consider the morphism of lineales $h : \mathbb{N} \to 2$ in Example 2.6. By Corollary 5.1, we may construct a functor $\text{Net}_h : \text{Net}_\mathbb{N} \to \text{Net}_2$ that forgets the amount of tokens consumed or produced by transitions and only keeps which places appear as pre and post conditions for transitions. For example, the Petri net in Figure 11 would be mapped to a Petri net without labels, i.e. where all the arcs are labelled by 1.
6. Conclusions and further work

Our work follows the lines of recent work on categorical compositionality of Petri nets: they can be composed along shared places [BM20] or along shared transitions [RSS14]. We have explored composing nets with some of the connectives typical of linear logic: products, coproducts, tensor and internal hom. We have presented a categorical model for Petri nets that focuses on the diverse nature of network relations. This approach allows the use of Petri nets with different kinds of transitions (different kinds of labels in their graphs), while maintaining their compositional.

Our model can handle different kinds of transition whose labels can be represented as a lineale (a poset version of a symmetric monoidal closed category). Several sets of labels, from those often used in empirical data modelling, can be endowed with the structure of a lineale, including: stoichiometric coefficients in chemical reaction networks ($L = \mathbb{N}$), reaction rates ($L = \mathbb{R}^+$), inhibitor arcs ($L = \mathbb{Z}$), gene interactions ($L = \{0, 1\}$), unknown or incomplete data ($L = \{-1, 0, 1\}$), and probabilities ($L = [0, 1]$).

The structure of the lineale is lifted to the category $M_L Set$ from which the category $Net_L$ of Petri nets is built. Both $M_L Set$ and $Net_L$ are symmetric monoidal closed categories with finite products and coproducts, providing a compositional way to put together smaller nets into bigger ones, making sure that morphisms between the component nets also assemble into morphisms between the resulting nets. Both these constructions are functorial.

The category $Net_L$ is a model for weighted and directed bipartite relations and therefore we anticipate applications of the compositionality of $Net_L$ in the broader context of directed bipartite graphs [IK00, Ham09], in particular, for their applications to real-world networks. For instance, the labelled wiring of these graphs is key to the empirical analysis of metabolic networks, where the metabolism of an organism is studied in terms of the concurrence of smaller functional subnets called modules. We wonder whether our formal connectives may assist in the reconstruction and understanding of the whole metabolism in terms of the concurrence of the modules.

There is much more work to be done still. Both in the applications we are pursuing and on the theory of Dialectica Petri nets. On the theory side notions of behaviour (token game) should be investigated and on the practical side we have still to investigate how the implemented systems can be modified to deal with our nets. Moreover, our exposition fixed the base category to be $Set$ and only varied the lineale parameter $L$. The possibility of changing the base category remains to be investigated. In particular, in [dP91a], the author shows how to construct a model of Intuitionistic Linear Logic with a modality “!” on the category $M_L C$, provided the base category $C$ is cartesian closed and has free monoids (such as $Set$). The reader interested in this construction and its interpretation in terms of Petri nets is referred to [dP91a]. In $M_{\mathbb{N}} C$, the construction works for a fixed lineale $(\mathbb{N}, \geq, +, 0, -\infty)$ and is parametrized by the category $C$. In light of our present work, this opens several avenues for further research. For instance, one could explore building a modality “!” in $M_L Set$ for any lineale $L$, or generalizing the $M_{\mathbb{N}} C$ construction to lineales other than $(\mathbb{N}, \geq, +, 0, -\infty)$ using our approach.

Finally, we leave for future work the investigation of the categorical properties of the constructions $M(-) Set$ and $Net(-)$, including their potential connections with fibrations and their double-categorical aspects. We also leave for future work the investigation of categorical properties of the constructions $M(-) Set$ and $Net(-)$, their connections with fibrations or their double categorical aspects.
Dialectica Petri nets share some of the pros and cons of other Linear Logic based nets. As far as we know no one has investigated Differential Linear Logic Petri nets, yet (see [EL10] for relating differential interaction nets to the $\pi$-calculus). We wonder if this would make the exchange of information with modellers somewhat easier. Finally we would like to investigate whether we could code our nets using Catlab https://github.com/AlgebraicJulia/Catlab.jl, a framework for computational category theory, written in the Julia language and already used for other styles of Petri nets.

**Acknowledgements.** We would like to thank the ACT Adjoint School 2020 and its organisers for first setting up our collaboration. We also thank Jade Master, Xiaoyan Li and Eigil Rischel for their initial work in this project.

**References**

[ALH07] Ilka M Axmann, Stefan Legewie, and Hanspeter Herzel. A minimal circadian clock model. In *Genome Informatics 2007: Genome Informatics Series Vol. 18*, pages 54–64. World Scientific, 2007.

[BG90] Carolyn Brown and Doug Gurr. A categorical linear framework for Petri nets. In *1990 Proceedings. Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 208–218. IEEE, 1990.

[BG95] Carolyn Brown and Doug Gurr. A categorical linear framework for Petri nets. *Information and Computation*, 122(2):268 – 285, 1995.

[BGdP91] Carolyn Brown, Doug Gurr, and Valeria de Paiva. A Linear Specification Language for Petri Nets (Aarhus Technical Report), 1991.

[BGMS21] John C Baez, Fabrizio Genovese, Jade Master, and Michael Shulman. Categories of nets. *arXiv preprint arXiv:2101.04238*, 2021.

[BJ77] Nuel D Belnap Jr. A useful four-valued logic. In *Modern uses of multiple-valued logic*, pages 5–37. Springer, 1977.

[Bla92] Andreas Blass. A game semantics for linear logic. *Annals of Pure and Applied logic*, 56(1-3):183–220, 1992.

[BM20] John C Baez and Jade Master. Open Petri nets. *Mathematical Structures in Computer Science*, 30(3):314–341, 2020.

[BMMS01] Roberto Bruni, José Meseguer, Ugo Montanari, and Vladimiro Sassone. Functorial models for Petri nets. *Information and Computation*, 170(2):207–236, 2001.

[Bor94] Francis Borceux. *Handbook of Categorical Algebra*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1994.

[dP89a] Valeria de Paiva. The Dialectica Categories. In *Categories in Computer Science and Logic: Proceedings of the AMS-IMS-SIAM Joint Summer Research Conference Held June 14-20, 1987 with Support from the National Science Foundation*, volume 92, page 47. American Mathematical Society, 1989.
Valeria de Paiva. A dialectica-like model of linear logic. In *Category Theory and Computer Science*, pages 341–356. Springer Berlin Heidelberg, 1989.

Valeria de Paiva. Categorical Multirelations, Linear Logic and Petri Nets, Technical Report, 1991.

Valeria de Paiva. The Dialectica Categories. University of Cambridge, Computer Lab Technical Report, PhD thesis, 1991.

Valeria de Paiva. Lineales: Algebras and categories in the semantics of linear logic. *Words, Proofs and Diagrams, CSLI Publications, Stanford*, pages 123–142, 2002.

Yrvann Emzivat, Benoit Delahaye, Didier Lime, and Olivier H Roux. Probabilistic time Petri nets. In *International Conference on Application and Theory of Petri Nets and Concurrency*, pages 261–280. Springer, 2016.

Marzieh Eidi, Amirhossein Farzam, Wilmer Leal, Areejit Samal, and Jürgen Jost. Edge-based analysis of networks: curvatures of graphs and hypergraphs. *Theory in Biosciences*, 139(4):337–348, 2020.

Thomas Ehrhard and Olivier Laurent. Interpreting a finitary pi-calculus in differential interaction nets. *Information and Computation*, 208(6):606–633, 2010. Special Issue: 18th International Conference on Concurrency Theory (CONCUR 2007).

Uffe Engberg and Glynn Winskel. Petri nets as models of linear logic. In *CAAP’90: 15th Colloquium on Trees in Algebra and Programming Copenhagen, Denmark, May 15–18, 1990 Proceedings 15*, pages 147–161. Springer, 1990.

Jean-Yves Girard. Linear logic. *Theoretical computer science*, 50(1):1–101, 1987.

Richard H. Hammack. Proof of a conjecture concerning the direct product of bipartite graphs. *European Journal of Combinatorics*, 30(5):1114–1118, 2009. Part Special Issue on Metric Graph Theory.

Thomas Hinze, Thorsten Lenser, Gabi Escuela, Ines Heiland, and Stefan Schuster. Modelling signalling networks with incomplete information about protein activation states: a p system framework of the kaiabc oscillator. In *International Workshop on Membrane Computing*, pages 316–334. Springer, 2009.

Reinhard W. Hoffmann. Introduction. In *Elements of Synthesis Planning*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

U. Höhle. *Commutative, residuated l—monoids*, pages 53–106. Springer Netherlands, Dordrecht, 1995.

Wilfried Imrich and Sandi Klavžar. *Product graphs, structure and recognition*. Wiley, New York, 2000.

Joachim Kock. Elements of Petri nets and processes. *arXiv preprint arXiv:2005.05108*, 2020.

Piergiulio Katis, Nicoletta Sabadini, and Robert FC Walters. Representing place/transition nets in span(graph). In *International Conference on Algebraic Methodology and Software Technology*, pages 322–336. Springer, 1997.

Wilmer Leal, Guillermo Restrepo, Peter F. Stadler, and Jürgen Jost. Forman-ricci curvature for hypergraphs. *Advances in Complex Systems*, 24:2150003, 2021.
[Mac71] Saunders MacLane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.

[Mas19] Jade Master. Generalized Petri nets. *arXiv preprint arXiv:1904.09091*, 2019.

[Mas20] Jade Master. Petri nets based on Lawvere theories. *Mathematical Structures in Computer Science*, 30(7):833–864, 2020.

[MM90] José Meseguer and Ugo Montanari. Petri nets are monoids. *Information and computation*, 88(2):105–155, 1990.

[MOM05] Narciso Marti-Oliet and José Meseguer. From petri nets to linear logic. In *Category Theory and Computer Science: Manchester, UK, September 5–8, 1989 Proceedings*, pages 313–340. Springer, 2005.

[NM20] Zachary G. Nicolaou and Adilson E. Motter. Missing links as a source of seemingly variable constants in complex reaction networks. *Physical Review Research*, 2:043135, Oct 2020.

[RE96] Grzegorz Rozenberg and Joost Engelfriet. Elementary net systems. In *Advanced Course on Petri Nets*, pages 12–121. Springer, 1996.

[RSS14] Julian Rathke, Paweł Sobociński, and Owen Stephens. Compositional reachability in Petri nets. In *International Workshop on Reachability Problems*, pages 230–243. Springer, 2014.

[Sch98] Joachim Schummer. The chemical core of chemistry I: A conceptual approach. *Hyle*, 4(2):129–162, 1998.

[SNMM11] Ayumu Saito, Masao Nagasaki, Hiroshi Matsuno, and Satoru Miyano. *Hybrid Functional Petri Net with Extension for Dynamic Pathway Modeling*, pages 101–120. Springer London, London, 2011.

[SS18] Bärbel M. R. Stadler and Peter Stadler. Reachability, connectivity, and proximity in chemical spaces. *MATCH Communications in Mathematical and in Computer Chemistry*, 80(3):639–659, 2018.

[ST14] Gavin J. Seal and Walter Tholen. *Monoidal structures*, page 18–144. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2014.

[Win87] Glynn Winskel. Petri nets, algebras, morphisms, and compositionality. *Information and Computation*, 72(3):197–238, 1987.

[Win88] Glynn Winskel. A category of labelled Petri nets and compositional proof system. In *1988 Proceedings. Third Annual Symposium on Logic in Computer Science*, pages 142–154. IEEE, 1988.