COLORED PETRI NETS ARE LAX DOUBLE FUNCTORS

JADE MASTER AND JOE MOELLER

Abstract.

Contents

1. Petri Nets and Their Semantics 3
2. Colored Petri Nets 5
3. Operational Semantics for Colored Petri Nets 10
4. The Operational Semantics is Free on the Unfolding 13
5. Conclusion 15

Appendix A. Double categories 15

Appendix B. Displayed Categories 17

References 23

Petri nets are a widely studied formalism for representing networks of processes and resources. When a system has sufficient complexity, modeling with ordinary Petri nets quickly becomes unwieldy. Colored Petri nets alleviate this problem by allowing repetitive structures to be encoded as extra data attached to the Petri net. This extra data can be thought of as a more detailed interpretation of the places and transitions in the underlying Petri net. From the ordinary definition of colored Petri net [Jen13], it is not clear how a colored Petri net can be obtained by applying a semantic interpretation to an ordinary Petri net. In this paper, we make precise the way in which colored Petri nets are syntax applied to semantics. Indeed, in Proposition 2.4, we show how a colored Petri net naturally determines a symmetric monoidal double functor

$$K: FP \to \mathsf{Span}(\mathsf{Kl}(\mathbb{N})).$$

Here P is an ordinary Petri net, FP is the categorical semantics of P upgraded trivially to a double category, and $\mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ is the double category whose loose morphisms are spans of free commutative monoids. Here FP is regarded as the "syntax", and $\mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ is the "semantics".

Rephrasing colored Petri nets in this way has many advantages. In particular,

- Transformations between double functors give a natural definition of morphism between colored Petri nets (Definition 2.6). In this paper, we compare this definition to the definition of morphism of colored Petri nets to the definition given by Lakos [Lak97, Lak00]. We find that our more abstract definition retains many of the desirable qualities of Lakos' definition.
- In 1990 [MM90] Montanari and Meseguer elegantly illustrated the connection between Petri nets and monoidal category theory by giving an operational semantics functor which turns each Petri net into a free symmetric monoidal category. The presentation of colored Petri nets as functors allows for an

analogous operational semantics to be defined based on the Grothendieck construction. In Definition 3.7 we construct this operational semantics category and in Proposition 3.8 we prove that it faithfully captures the firing sequences of a colored Petri net.

- Unfolding is an algorithmic process which turns a colored Petri net into an ordinary one for the purposes of analysis [Jen13]. In this paper we describe unfolding as a functor

$$\mathtt{UNFOLD}\colon \mathsf{cPetri} \to \mathsf{Petri}.$$

The correctness of unfolding can be expressed concisely using category theory. In Theorem 4.2, we show that there is a diagram

$$\begin{tikzcd} & \mathsf{Petri} \arrow[dr, "F"] & \\ \mathsf{cPetri} \arrow[ur, "\mathtt{UNFOLD}"] \arrow[rr, "\int"'] & & \mathsf{CMC}. \end{tikzcd}$$

which commutes up to isomorphism. This commutativity expresses the fact that unfolding preserves processes by providing an isomorphism between the sequences of processes of a colored Petri net and the sequences of processes of its unfolding.

A plan of this paper is as follows:

- In Section 1 we give a review of Petri nets and their process semantics from the perspective of category theory.
- In Section 2 we introduce a definition of colored Petri nets and their morphisms and discuss their relationship to the traditional notions of colored Petri nets and their morphisms.
- In Section 3 we describe unfolding of colored Petri nets as a functor. We use this to give a categorical process semantics for colored Petri nets.
- In Appendix B, we prove a key technical result: the equivalence between displayed categories and functors. This is a result which is well-known as folklore but lacked a formal proof in the literature until now.

0.1. **Related Work.** A characterization of colored Petri nets as functors has already appeared in [GS20]. Although [GS20] and the current paper use similar ideas, they were developed independently. Furthermore, the colored Petri nets studied in this paper have some differences from the colored Petri nets studied by Genovese and Spivak:

- The domain is not the same: The free symmetric monoidal category used by Genovese and Spivak is the non-commutative variant developed by Sassone [Sas95]. We use the free commutative monoidal category functor originated in [MM90] and clarified in [Mas20]. Our approach gives an operational semantics which highlights the *collective token philosophy*; the order and identities of the tokens are not represented in this operational semantics. On the other hand, the construction of Sassone follows the *individual token philosophy* which does keep track of this data. We chose the free commutative monoidal category because it's existence follows from more general categorical principles. To freely generate symmetric monoidal categories which exhibit the individual token philosophy, we suggest using pre-nets, a non-commutative variant of Petri nets which have a more straightforward adjunction into the right sort of symmetric monoidal category [BMMS01].

- The arc inscriptions are not the same: In Genovese and Spivak's approach, each transition is mapped to an isomorphism class of spans of sets. On the other hand, in our approach each transition is annotated with a span in the Kleisli category $\mathsf{Kl}(\mathbb{N})$. Using the Kleisli category allows the colors of the transitions to have multisets of inputs and outputs whereas the colors on the transitions in [GS20] may only map unique inputs to unique outputs. As proven in Proposition 2.4, our definition matches the original definition in [Jen81].
- The definition in [GS20] is evil: Evil is not meant as a moral judgement, just that the model does not satisfy the principle of equivalence (i.e. [evi]). The codomain of a colored Petri net in Genovese and Spivak's approach is a decategorification of the double category $\mathsf{Span}(\mathsf{Set})$ whose morphisms are equivalence classes of spans. Defining a functor into this category requires choosing a non-canonical representatives from these equivalence classes. If these representatives are chosen inconsistently, two colored Petri nets which are equal may have totally different presentations. To make matters worse, in [GS20] the authors have also decategorified the cartesian monoidal structure of $\mathsf{Span}(\mathsf{Set})$ to make it strictly associative and unital. With so many equivalence relations imposed on spans of sets, it is unclear how to choose representatives in consistent way. In this paper we have remedied this issue by using lax functors and double categories so that each transition is annotated by an actual span rather than an equivalence class.

In [GLP22], the authors define a categorical semantics for *bounded* Petri nets using the displayed category construction. They do not record a proof of the equivalence in Appendix B.

1. Petri Nets and Their Semantics

One one hand, Petri nets can be thought of as (directed, multi-) graphs with two distinct types of nodes. However, it is more in line with the spirit of Petri nets to think of them as graphs in which arrows are permitted to have a multiset of source nodes, and a multiset of target nodes.

Recall that the free commutative monoid on a set X is given by

$$\mathbb{N}[X] = \{a \colon X \to \mathbb{N} \mid f(x) \neq 0 \text{ for only finitely many elements of } X\}$$

with addition given pointwise. To keep notational overhead low, we let $\mathbb{N}$ denote the free commutative monoid monad $\mathbb{N} \colon \mathsf{Set} \to \mathsf{Set}$.

Definition 1.1. A **Petri net** is a pair of functions of the following form.

$$T \overset{s}{\underset{t}{\rightrightarrows}} \mathbb{N}[S]$$

We call T the set of **transitions**, S the set of **places**, s the **source** function and t the **target** function. A **Petri net morphism** $(f, g) \colon (T, S, t, s) \to (T', S', t', s')$ consists of a pair of functions $f \colon T \to T'$ and $g \colon S \to S'$ such that the following diagrams commute.

$$\begin{tikzcd} T \arrow[r, "s"] \arrow[d, "f"'] & \mathbb{N}[S] \arrow[d, "\mathbb{N}[g]"] \\ T' \arrow[r, "s'"'] & \mathbb{N}[S'] \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "t"] \arrow[d, "f"'] & \mathbb{N}[S] \arrow[d, "\mathbb{N}[g]"] \\ T' \arrow[r, "t'"'] & \mathbb{N}[S'] \end{tikzcd}$$

Let Petri denote the category of Petri nets and Petri net morphisms, with composition defined pointwise.

Our definition of morphisms between Petri nets follows that found in the work of Baez and Master [Mas20, BM20]. This differs from the earlier definition used by Sassone [Sas95, Sas96] and Degano–Meseguer–Montanari [DMM89], in that the definition presented here requires that the homomorphism between free commutative monoids come from a function between the sets of places. We have chosen our definitions because they are more well-behaved categorically. In particular, our category Petri is cocomplete ([Mas20], Prop. 3.10).

Petri nets have a natural semantics which is described by *the token game*. Each place of a Petri net is equipped with a natural number of tokens. Such an assignment is called a **marking**, and is formally represented by an element $m \in \mathbb{N}[S]$, or equivalently, a function $m: S \to \mathbb{N}$. Players are then allowed to shuffle the tokens around by *firing* transitions. Given a particular marking m, a transition τ can be fired if there are enough tokens in all of its source places. Specifically, τ can be fired if and only if $s(\tau) \le m$ under the pointwise order on $\mathbb{N}[S]$. Formally, a **firing** of P is a tuple (τ, x, y), where τ is a transition and x and y are markings such that $x - s(\tau) + t(\tau) = y$. Evocatively, we write such tuples as $\tau: x \to y$.

Notice that the same transition is generally a component of many firings which while technically distinct, exhibit essentially the same action.

If there are enough tokens in an initial marking, transitions can be fired in sequence and in parallel. This is how Petri nets are used to model distributed systems. Given two firings $\tau: x \to y$ and $\tau': x' \to y'$, we write their parallel firing as $\tau + \tau': x + x' \to y + y'$, where the sum of the markings is the sum in $\mathbb{N}[S]$. Given two firings $\tau: x \to y$ and $\sigma: y \to z$, we write their sequential firing as $\sigma \circ \tau: x \to z$.

As suggested by our choice of notation, these operations fit together into the structure of a monoidal category, which we will denote as FP. The objects of FP are markings of P, the morphisms are "firing patterns", or legal sequences of transition firings, potentially in parallel, up to a very natural equivalence of patterns. Meseguer and Montanari were the first to demonstrate the relationship between Petri nets and symmetric monoidal categories [MM90]. Petri nets generate a specific kind of symmetric monoidal category.

Definition 1.2. A **commutative monoidal category** is a commutative monoid object internal to the 1-category Cat. Explicitly, a commutative monoidal category is a strict monoidal category $(C, \otimes, I)$, such that each symmetry map $\sigma_{a,b}: a \otimes b \to b \otimes a$ is an identity map.

A commutative monoidal category is precisely a category where the objects and morphisms form commutative monoids and the structure maps are commutative monoid homomorphisms.

Definition 1.3. Let CMC be the category whose objects are commutative monoidal categories and whose morphisms are strict monoidal functors.

Note that every monoidal functor between commutative monoidal categories is automatically a strict symmetric monoidal functor, so the adjective symmetric is not included in the above definition.

Proposition 1.4. *There is an adjunction*

$$\begin{tikzcd} \mathsf{Petri} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathsf{CMC} \arrow[l, bend left, "U"] \end{tikzcd}$$

whose left adjoint generates the free commutative monoidal category on a Petri net.

Proof. This is a special case of [Mas20, Thm. 5.1] which shows that there is similar adjunction for any Lawvere theory Q. When Q is the Lawvere theory for commutative monoids this theorem gives the desired adjunction. The above Theorem also contains a description of this left adjoint in more detail. □

2. Colored Petri Nets

Jensen defined *colored Petri nets* [Jen81], and Lakos rephrased this definition as follows [Lak97]. Note that some sources refer to the following as *colored Petri nets with guards.*

Definition 2.1 ([Lak97] Def. 4.8)**.** An (unmarked) **colored Petri net with guards** is a tuple

$$(P, T, A, C, E)$$

where

- P is a set of **places**
- T is a set of **transitions**
- $A \subseteq P \times T \cup T \times P$ is a set of **arcs**
- C is an assignment of each place and transition to a color set i.e. a mapping

$$C \colon P + T \to \mathsf{Set}$$

- E is the set of **arc inscriptions**. This is a set of functions

$$E(a) \colon C(t) \to \mathbb{N}[C(p)]$$

  for every arc $a = (p, t)$ or $a = (t, p)$ in A

By this definition, a colored Petri net is a Petri net with a set of colors associated to each place, a set of colors associated to each transition, and functions between the free commutative monoids on each arc. A few remarks:

- the underlying Petri net $N = (P, T, A)$ of the above defintion can be interpreted as a Petri net in the sense of Definition 1.1. N can be thought of as a pair of functions

$$T \overset{s}{\underset{t}{\rightrightarrows}} \mathbb{N}[P]$$

  where s and t send a transition τ to the sum of all places which are connected to τ by an input arc and output arc respectively.
- The guards are defined implicitly. Transitions which do not fire unless a condition is met are represented by arc inscriptions which produce colorings on the transitions which have no capability of leaving along an output arc inscription.

Colored Petri nets with guards can be interpreted as an assignment of syntax to semantics. The semantics will be the following double category which is a special case of the double category of spans $\mathsf{Span}(C)$ for a category C with pullbacks (e.g. [DPP10, §3]).

Proposition 2.2. *Let* $\mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ *denote the double category where*

- *objects are sets,*
- *horizontal 1-cells are spans in* $\mathsf{Kl}(\mathbb{N})$

$$\mathbb{N}[X] \longleftarrow A \longrightarrow \mathbb{N}[Y]$$

- *horizontal composition is given by pullback in* $\mathsf{Kl}(\mathbb{N})$*,*
- *vertical 1-cells are functions,*
- *vertical 2-cells are commuting squares*

$$\begin{tikzcd} \mathbb{N}[X] \arrow[d, "\mathbb{N}(f)"] & A \arrow[l] \arrow[r] \arrow[d, "h"] & N[Y] \arrow[d, "\mathbb{N}(g)"] \\ \mathbb{N}[X'] & A' \arrow[l] \arrow[r] & \mathbb{N}[Y'] \end{tikzcd}$$

- *and vertical composition is given by composition of functions.*

Proposition 2.3. $\mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ *may be equipped with the structure of a symmetric monoidal double category (e.g. A.3)* $(\mathsf{Span}(\mathsf{Kl}(\mathbb{N})), +)$ *where*

- *For objects* X, Y *we have* $X + Y$ *as their disjoint union,*
- *For functions* $f : X \to Y, g : X' \to Y'$, $f + f' : X + X' \to Y + Y'$ *is their disjoint union.*
- *on horizontal 1-cells we have*

$$\mathbb{N}[X] \xleftarrow{l} A \xrightarrow{r} \mathbb{N}[Y] \qquad + \qquad \mathbb{N}[X'] \xleftarrow{l'} A' \xrightarrow{r'} \mathbb{N}[Y']$$

$$\begin{tikzcd} & A + A' \arrow[dl, "l+l'"'] \arrow[dr, "r+r'"] & \\ \mathbb{N}[X] + \mathbb{N}[X'] \arrow[d, "st_{X,X'}"'] & & \mathbb{N}[Y] + \mathbb{N}[Y'] \arrow[d, "st_{Y,Y'}"] \\ \mathbb{N}[X + X'] & & \mathbb{N}[Y + Y'] \end{tikzcd}$$

  where $st_{X,X'}$ *and* $st_{Y,Y'}$ *are the components of the strength for* $\mathbb{N}$ *i.e. the natural transformation* $st : + \circ (\mathbb{N} \times \mathbb{N}) \Rightarrow \mathbb{N} \circ +$ *is the copairing of the "extension" maps* $\mathbb{N}[X] \to \mathbb{N}[X + X']$ *which send a marking* m *to the marking that agrees with* m *and is* 0 *elsewhere.*
- *and the action of plus on 2-cells is given by*

$$\begin{tikzcd} \mathbb{N}[X] \arrow[d, "\mathbb{N}[f]"'] & A \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "g"] & \mathbb{N}[Y] \arrow[d, "\mathbb{N}[h]"] \\ \mathbb{N}[Q] & B \arrow[l, "s"] \arrow[r, "t"'] & \mathbb{N}[R] \end{tikzcd} + \begin{tikzcd} \mathbb{N}[X'] \arrow[d, "\mathbb{N}[f']"'] & A' \arrow[l, "l'"'] \arrow[r, "r'"] \arrow[d, "g'"] & \mathbb{N}[Y'] \arrow[d, "\mathbb{N}[h']"] \\ \mathbb{N}[Q'] & B' \arrow[l, "s'"] \arrow[r, "t'"'] & \mathbb{N}[R'] \end{tikzcd}$$

$$= \begin{tikzcd} \mathbb{N}[X+X'] \arrow[d, "\mathbb{N}[f+f']"'] & A+A' \arrow[l, "st\circ(l+l')"'] \arrow[r, "st\circ(r+r')"] \arrow[d, "g+g'"] & \mathbb{N}[Y+Y'] \arrow[d, "\mathbb{N}[h+h']"] \\ \mathbb{N}[Q+Q'] & B+B' \arrow[l, "st\circ(s+s')"] \arrow[r, "st\circ(t+t')"'] & \mathbb{N}[R+R'] \end{tikzcd}$$

Proof. [Shu08, Ex. 9.2] gives a symmetric cartesian monoidal structure $(\mathsf{Span}(C), \times)$ whenever C has finite limits. We instantiate this in the case when $C = \mathsf{Kl}(\mathbb{N})$. Note that $+$ is a biproduct on $\mathsf{Kl}(\mathbb{N})$ which is in particular a cartesian product. □

Proposition 2.4. *A colored petri net with guards (P, T, A, C, E) defines a symmetric monoidal lax double functor*

$$K : FN \to (\mathsf{Span}(\mathsf{Kl}(\mathbb{N})), +)$$

where FN is the free commutative monoidal category on the underlying Petri net

$$N = \begin{tikzcd} T \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] & \mathbb{N}[P] \end{tikzcd} .$$

Proof. Because FN is freely generated both monoidally and compositionally, a functor out of FN which preserves those two operations is uniquely defined by its assignment on generators. A place $p \in P$ is assigned to the free commutative monoid $\mathbb{N}[C(p)]$ and a transition $\tau : s(\tau) \to t(\tau) \in T$ is assigned to the span

$$\begin{tikzcd} & \mathbb{N}[C(\tau)] \arrow[dl, "(E(x,t))_{x\in s(\tau)}"'] \arrow[dr, "(E(t,y))_{y\in t(\tau)}"] & \\ \bigoplus_{x\in s(\tau)} \mathbb{N}[C(x)] & & \bigoplus_{y\in t(\tau)} \mathbb{N}[C(y)] \end{tikzcd}$$

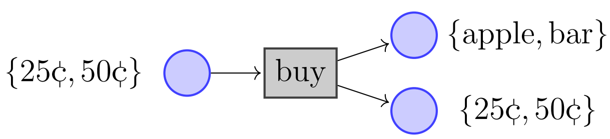
Where $(E(t,y))_{y\in t(\tau)}$ denotes the pairing of the arc inscriptions in the input of τ and $(E(t,y))_{y\in t(\tau)}$ denotes the pairing of the arcs in the output of τ. The rest of the objects and morphisms in FN are freely generated by the places and transitions of N so their assignment under K is given by composites and direct sums of the above assignments. There is just one subtlety, $(FP, +)$ is strictly monoidal and commutative whereas $(\mathsf{Span}(\mathsf{Kl}(\mathbb{N})), \otimes)$ only satisfies those properties up to coherent isomorphism. Luckily, the functor K does not need to be strictly monoidal. This means that instead of having $K(a+b) = K(a) \otimes K(b)$, we have a coherent isomorphism

$$\gamma_{a,b} : K(a) \times K(b) \to K(a+b).$$

After choosing an initial association and permutation for the image of every sum in FN, this isomorphism will consist of whatever symmetries or associators in $\mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ are necessary to preserve the commutativity and associativity equations in FN. □

To understand this characterization, it is helpful to look at an example.

Example 2.5. The following colored Petri net models the operation of a vending machine which dispense candy bars for 25¢ and apples for 75¢:



There are three species representing coins going in, food products going out, and coins going out. For each place, its set of colors is indicated next to it. The set of colors and their arc inscriptions are indicated as follows:

$$\begin{aligned} 25¢ + 50¢ &\leftarrow\!\shortmid \mathrm{a1} \mapsto (\mathrm{apple}, 0) \\ 2 \cdot 50¢ &\leftarrow\!\shortmid \mathrm{a2} \mapsto (\mathrm{apple}, 50¢) \\ 3 \cdot 25¢ &\leftarrow\!\shortmid \mathrm{a3} \mapsto (\mathrm{apple}, 0) \\ 25¢ &\leftarrow\!\shortmid \mathrm{b1} \mapsto (\mathrm{bar}, 0) \\ 50¢ &\leftarrow\!\shortmid \mathrm{b2} \mapsto (\mathrm{bar}, 25¢) \\ 25¢ &\leftarrow\!\shortmid \mathrm{e1} \mapsto (0, 25¢) \\ 50¢ &\leftarrow\!\shortmid \mathrm{e2} \mapsto (0, 50¢) \end{aligned}$$

The middle column gives the colors of the transition buy and the columns to its left and right give the arc inscriptions for these colors on the input and output places respectively. Note that this table can be read as a span of free commutative monoids

$$\mathbb{N}[\{25¢, 50¢\} \longleftarrow \mathbb{N}[C(\mathrm{buy})] \longrightarrow \mathbb{N}[25¢, 50¢] \oplus \mathbb{N}[\mathrm{apple}, \mathrm{bar}]$$

where $C(\mathrm{buy})$ is the set whose elements are given by the middle column above. Let P be the underlying Petri net.The above data defines a symmetric monoidal double functor $K \colon FP \to \mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$. The assignment on generators of FP is shown above, and this assignment is freely extended to monoidal products and composites.

One advantage of phrasing colored Petri nets in categorical terms is that it gives a natural definition of morphism between colored Petri nets:

Definition 2.6. Let $K \colon FP \to \mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ and $K' \colon FP' \to \mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ be colored Petri nets. Then a **morphism of colored Petri nets** (f, g, α) is a morphism of Petri nets $(f, g) \colon P \to P'$ and a monoidal vertical transformation α of the following shape.

$$\begin{tikzcd} FP \arrow[dd, "F(f,g)"'] \arrow[dr, "K"] & \\ \arrow[r, phantom, "\Downarrow\alpha"] & \mathsf{Span}(\mathsf{Kl}(\mathbb{N})) \\ FP' \arrow[ur, "K'"'] & \end{tikzcd}$$

Because this vertical transformation is monoidal, it suffices to define it on the generators of FN, i.e. the places and transitions of N and not arbitrary sums of those places and transitions. Therefore, a vertical transformation as above consists of the following:

- For each place p a function between the color sets $\alpha_p \colon K(p) \to K'(g(p))$. Because FP is made into a double category trivially, the naturality square at this level is trivial.

- For each transition τ, a morphism of spans i.e. a commuting diagram of the form

$$\begin{tikzcd}
K(s(\tau)) \arrow[d] & K(\tau) \arrow[l] \arrow[r] \arrow[d] & K(t(\tau)) \arrow[d] \\
K'(s'(f(\tau)))) & K'(f(\tau)) \arrow[l] \arrow[r] & K'(t'(f(\tau)))
\end{tikzcd}$$

  Because FP has trivial vertical 2-cells, the naturality square at this level is also trivial.
- vertical transformations between double categories are required to respect composition and identities. Respecting identities is the condition that the morphism between identity spans

$$\begin{tikzcd}
K(x) \arrow[d, "\alpha_x"] \arrow[r, equal] & K(x) \arrow[d] \arrow[r, equal] & K(x) \arrow[d, "\alpha_x"] \\
K'(g(x)) \arrow[r, equal] & K'(g(x)) \arrow[r, equal] & K'(g(x))
\end{tikzcd}$$

  has all three components given by α_x. Respecting composition boils down to the requiring that α is built by extending its value on generators. For composable morphisms $\tau\colon x \to y$ and $\kappa\colon y \to z$ in FN, α gives the morphism of spans

$$\begin{tikzcd}
K(x) \arrow[d, "\alpha_x"] & K(\tau) \times_{K(y)} K(\kappa) \arrow[l] \arrow[r] \arrow[d, "\alpha_{K(\kappa\circ\tau)}"] & K(z) \arrow[d, "\alpha_z"] \\
K'(g(x)) & K'(f(\tau)) \times_{K'(g(y))} K'(f(\kappa)) \arrow[l] \arrow[r] & K'(g(z))
\end{tikzcd}$$

  respecting composition requires that the arrow $\alpha_{K(\kappa\circ\tau))}$ is equal to the pullback of arrows $\alpha_{K(\tau)} \times_{\alpha_{K(y)}} \alpha_{K(\kappa)}$.

In [Lak97], Lakos also defined morphisms of colored Petri nets. Roughly, a morphism from a colored Petri net $K = (P, T, A, C, E)$ to a colored Petri net $K' = (P', T', A', C', E')$ is a function between each set which can optionally satisfy the following properties:

- **Structure Respecting:** A morphism of colored Petri nets is structure respecting if it preserves arcs. This means that if an arc a of K connects a transition t and a place p, then the image of this arc should connect the image of t to the image of p. The above definition is structure respecting because it contains a morphism of the underlying Petri nets
- **Color Respecting:** A morphism is color respecting if it sends the color on a place or a transition to a subset of the colors on the image of that place or transition. Our definition weakens this definition by requiring that there is an arbitrary monoid homomorphism between the two relevant color sets rather than an inclusion. A color respecting morphism must also preserve arc inscriptions. This is reflected in our definition by that fact that the components of α on the transitions are commuting diagrams of spans.

Two morphisms of colored Petri nets can be composed by composing each component separately. This forms the structure of a category.

Definition 2.7. Let **cPetri** be the category of colored Petri nets and their morphisms.

3. Operational Semantics for Colored Petri Nets

Given a colored Petri net, there is a way to associate a category whose morphisms represent executions of your colored Petri net. To make this precise we will use the notion of step sequence. In [Lak97] these concepts are defined, here we adapt these definitions to our purposes:

Definition 3.1. Let P be the petri net $T \rightrightarrows \mathbb{N}[S]$. A **marking** of a colored Petri Net $K\colon FP \to \mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ is a pair (m,x) where m is a marking of the Petri P and x is an element of $K(m)$. A **step** of K with **source** (m,x) and **target** (n,y) is a tuple (τ, f) where

- τ is an element of $\mathbb{N}[T]$ with source given by m and target given by n and,
- f is an element of the apex of $K(\tau)$ whose image under the left and right legs of $K(\tau)$ are given by x and y respectively.

A **step sequence** is a sequence of composable steps $\{(\tau_i, f_i)\}_{i=1}^k$. Composable means that the target of (τ_i, f_i) is equal to the source of (τ_{i+1}, f_{i+1}) for all $1 < i < k$. In particular, setting $k = 0$ gives a step sequence from every marking to itself. The source of a step sequence is defined to be the source of its first step and the target of a step sequence is the target of its last step.

Our categorical operational semantics will use the Grothendieck construction for displayed categories.

3.1. Displayed Categories.

Definition 3.2. A **displayed category** over C is a lax double functor

$$F\colon C \to \mathsf{Span}(\mathsf{Set})$$

Displayed categories have been studied by many authors. For example, see Jean Bénabou's lecture notes [Bén00]. The term displayed categories was first coined in [AL19]. The construction of $\int D$ is a generalization of the Grothendieck construction [GR03].

Definition 3.3. For a displayed category $F\colon C \to \mathsf{Span}(\mathsf{Set})$ its total category $\int F$ is a category where:

- An object is a tuple (c,x) where c is an object of C and $x \in F(c)$.
- Let $f\colon c \to d$ be a morphism in C with image

$$\begin{tikzcd} & F(f) \arrow[dl, "a"'] \arrow[dr, "b"] & \\ F(c) & & F(d) \end{tikzcd}$$

  then for every object $r \in F(f)$, there is a morphism $(f,r)\colon (c, a(r)) \to (d, b(r))$.
- For morphisms $(f,r)\colon (c,x) \to (d,y)$ and $(g,s)\colon (d,y) \to (e,z)$ their composite is defined to be $(g \circ f, t)\colon (c,x) \to (e,z)$ where t is the image of the pair (r,s) under the laxator of F

$$F(f) \times_{F(d)} F(g) \to F(g \circ f).$$

$\int F$ is equipped with a functor

$$\int F \to C$$

which sends every object and morphism to its first component.

Displayed categories form a category and the total category construction is a functor.

Definition 3.4. Let $F : C \to \mathsf{Span}(\mathsf{Set})$ and $G : D \to \mathsf{Span}(\mathsf{Set})$ be displayed categories. Then a **morphism of displayed categories** $(j, \alpha) : F \to G$ is a functor $j : C \to D$ and a vertical transformation between double functors filling the following diagram

$$\begin{tikzcd} C \arrow[dr, "F"] \arrow[dd, "j"'] & \\ \arrow[r, phantom, "\Downarrow\alpha"] & \mathsf{Span}(\mathsf{Set}) \\ D \arrow[ur, "G"'] & \end{tikzcd}$$

This defines a category Disp with composition defined as composition of vertical transformations. Let Cat^2 be the category where objects are functors $D \to C$ and morphisms are commuting squares

$$\begin{tikzcd} D \arrow[r] \arrow[d] & D' \arrow[d] \\ C \arrow[r] & C' \end{tikzcd}$$

The following result, although very fundamental, has no reference as explained in [PA97]. Therefore it is a result of folklore. In this paper, we remove the folklore status of this theorem and prove it in Section B.

Theorem 3.5. *The total category construction gives an equivalence*

$$\int : \mathsf{Disp} \xrightarrow{\sim} \mathsf{Cat}^2.$$

Proposition 3.6. *There is a symmetric monoidal double functor*

$$U : \mathsf{Span}(\mathsf{Kl}(\mathbb{N})) \to \mathsf{Span}$$

given by

- *on objects* $\lambda X \mapsto \mathbb{N}[X]$,
- *leaving horizontal 1-cells unchanged,*
- *on vertical 1-cells sending a function* $f : X \to Y$ *to the function* $\mathbb{N}[f] : \mathbb{N}[X] \to \mathbb{N}[Y]$,
- *and leaving vertical 2-cells unchanged.*

We may compose a colored Petri net $K : FN \to \mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ with the double functor U to obtained a displayed category $U \circ K : FN \to \mathsf{Span}$.

Definition 3.7. The **categorical operational semantics functor** for colored Petri nets

$$\int : \mathsf{cPetri} \to \mathsf{Cat}$$

is the composition of $\int : \mathsf{Disp} \to \mathsf{Cat}$ with $\mathsf{Span}(U) : \mathsf{cPetri} \to \mathsf{Disp}$. Explicitly, for a colored Petri net $K : FP \to \mathsf{Span}(\mathsf{Kl}(\mathbb{N}))$ to the category $\int K$ where:

- An object is a pair (m, x) where m is an object of FP and x is an element of the free commutative monoid $K(m)$.
- A morphism from (m, x) to (n, y) is a pair (f, g) where $f\colon m \to n$ is a morphism in FP and g is an element in the apex of $F(f)$ which is mapped to x and y by the left and right legs of $F(f)$ respectively.

For a morphism of colored Petri nets

$$\begin{tikzcd}
FP \arrow[dd, "F(f,g)"'] \arrow[dr, "K"] & \\
{} \arrow[r, phantom, "\alpha\Downarrow"] & \mathsf{Span}(\mathsf{Kl}(\mathbb{N})) \\
FP' \arrow[ur, "K'"'] &
\end{tikzcd}$$

The category $\int K$ is an operational semantics for the colored Petri net K in the sense that morphisms of $\int K$ represent sequences of processes which can occur in K.

Proposition 3.8. *There is bijection between step sequences in K and morphisms in $\int K$.*

Proof. Each step (τ, f) with source (m, x) and target (n, y) gives a morphism $(\tau, f)\colon (m, x) \to (n, y)$. A step sequence $\{(\tau_i, f_i)\}_{i=1}^k$ gives a morphism by composing all the morphisms (τ_i, f_i) with each other in the natural way.

Conversely, let $(f, g)\colon (m, x) \to (n, y)$ be a morphism in $\int K$. FP is freely generated by the transitions of P under $+$ and composition. Therefore f will be equal to an expression built from the basic transitions using $+$ and $\circ$. The operation $+$ satisfies the interchange law

$$c \circ a + d \circ b = (c + d) \circ (a + b)$$

whenever all composites are defined. This law can be used inductively to turn f into a composite of sums

$$f_1 \circ f_2 \circ \ldots \circ f_n$$

where each $f_i\colon x_i \to x_{i+1}$ is a sum of transitions in P. Because K preserves composition strongly, there is an isomorphism

$$K(f) = K(f_1 \circ f_2 \ldots \circ f_n) \cong K(f_1) \times_{K(x_2)} K(f_2) \times_{K(x_3)} \ldots \times_{K(x_n)} K(f_n).$$

This isomorphism gives a way to decompose the element $g \in K(f)$ into a tuple of composable elements $g_1, g_2, \ldots, g_n$ where each g_i is an element of $K(f_i)$. The pairs (τ_i, g_i) give a step sequence of K which corresponds to the morphism (f, g). □

Note that functoriality of $\int$ justifies the definition of morphism for colored Petri nets. Definition 3.7 shows how morphisms of colored Petri nets lift to functors between their categories of step sequences. For a morphism of colored Petri nets $(j, \alpha)\colon K \to K'$, functoriality of $\int (j, \alpha)\colon \int K \to \int K'$ says that the mapping between step sequences induced by $\int (j, \alpha)$ respects concatenation.

4. The Operational Semantics is Free on the Unfolding

Unfolding is a useful technique which reduces the analysis of colored Petri nets to that of ordinary Petri nets. This correspondence was first noticed by Jensen in the first volume of his book *Coloured Petri nets: basic concepts, analysis methods and practical use* [Jen13]. Since then, efficient algorithms have been developed for unfolding colored Petri nets [Liu12], [LHY12], [KLPA06]. It turns out that the operational semantics defined in the previous section is always the CMC on a some Petri net. The well-known unfolding construction gives this Petri net which generates $intK$.

Definition 4.1. For a colored Petri net K with underlying Petri net $T \rightrightarrows \mathbb{N}[S]$. For a free commutative monoid M, let $\#M$ denote it's generating set. The **unfolding** of K is the Petri net

$$\mathtt{UNFOLD}(K) = \hat{T} \underset{\hat{t}}{\overset{\hat{s}}{\rightrightarrows}} \mathbb{N}[\hat{S}]$$

where

- $\hat{T} = \{(\tau, f) \,|\, \tau \in T \text{ and } f \in \#K(\tau)\}$,
- $\hat{S} = \{(p, x) \,|\, p \in S \text{ and } x \in \#K(p)\}$,
- $\hat{s}\colon \hat{T} \to \mathbb{N}[\hat{S}]$ sends a pair (τ, f) to the pair $(s(\tau), K(\tau)_{s(\tau)}(f)$ and,
- $\hat{t}\colon \hat{T} \to \mathbb{N}[\hat{S}]$ sends a pair (τ, f) to the pair $(t(\tau), K(\tau)_{t(\tau)}(f))$.

One advantage of reframing colored Petri nets in terms of category theory is that the definition of unfolding can be justified by categorical means. The categorical semantics for Petri nets and colored Petri nets can be drawn together to get

$$\begin{tikzcd} & \mathsf{Petri} \arrow[dr, "F"] & \\ \mathsf{cPetri} \arrow[rr, "\int"'] & & \mathsf{CMC}. \end{tikzcd}$$

To unfold a colored Petri net, we wish to find a functor $\mathtt{UNFOLD}\colon \mathsf{cPetri} \to \mathsf{Petri}$ such that for a colored Petri net K, the semantics of it's unfolding $F(\mathtt{UNFOLD}(K))$ matches it's original semantics $\int K$. This property says roughly that the behavior of an unfolded colored Petri must match the behavior of colored Petri net you started with. In other words $\mathtt{UNFOLD}$ must make the following diagram commute:

$$\begin{tikzcd} & \mathsf{Petri} \arrow[dr, "F"] & \\ \mathsf{cPetri} \arrow[ur, "\mathtt{UNFOLD}"] \arrow[rr, "\int"'] & & \mathsf{CMC} \end{tikzcd}$$

The unfolding defined above does indeed satisfy this property.

Theorem 4.2. *The unfolding construction extends to a functor*

$$\mathtt{UNFOLD}\colon \mathsf{cPetri} \to \mathsf{Petri}$$

satisfying the commutative diagram

$$\begin{tikzcd} & \mathsf{Petri} \arrow[dr, "F"] & \\ \mathsf{cPetri} \arrow[ur, "\texttt{UNFOLD}"] \arrow[rr, "\int"'] & & \mathsf{CMC}. \end{tikzcd}$$

Proof. $\texttt{UNFOLD}$ is extended to morphisms as follows: For a morphism of colored Petri nets $(f,g,\alpha): K \to K'$, $\texttt{UNFOLD}(f,g,\alpha): \texttt{UNFOLD}(K) \to \texttt{UNFOLD}(K')$ consists of the maps

$$\begin{aligned} \hat{f}: \hat{T} &\to \hat{T}' \\ (\tau, k) &\mapsto (f(\tau), \alpha_\tau(k)) \end{aligned} \qquad \text{and} \qquad \begin{aligned} \hat{g}: \hat{S} &\to \hat{S}' \\ (p, x) &\mapsto (g(p), \alpha_p(x)) \end{aligned}$$

$(\hat{f}, \hat{g})$ is indeed a morphism of Petri nets. Commuting with the source and target in the first component follows from (f,g) being a morphism of Petri nets. Commuting with the source and target in the second component follows from α_τ taking part in the commutative diagram

$$\begin{tikzcd} K(s(\tau)) \arrow[d, "\alpha_{s(\tau)}"'] & K(\tau) \arrow[l] \arrow[r] \arrow[d, "\alpha_\tau"] & K(t(\tau)) \arrow[d, "\alpha_{t(\tau)}"] \\ K'(s'(f(\tau)) & K'(f(\tau)) \arrow[l] \arrow[r] & K'(t'(f(\tau)). \end{tikzcd}$$

Functoriality of $\texttt{UNFOLD}$ follows very quickly from the definitions.

Next we show that $\texttt{UNFOLD}$ respects the categorical semantics functors for cPetri and Petri. For a colored Petri net K, we have that an isomorphism of categories

$$F \circ \texttt{UNFOLD}(K) \cong \int K$$

First note that we have a bijection

$$\hat{S} \cong \coprod_{p \in S} \# K(p)$$

between $\hat{S}$ and the disjoint union of all possible colors. Therefore, the objects of $F \circ \texttt{UNFOLD}(K)$ are given by

$$\mathbb{N}[\hat{S}] \cong \mathbb{N}[\coprod_{p \in S} \# K(p)] \cong \otimes_{p \in S} \mathbb{N}[\# K(p)]$$

where the last isomorphism follows from $\mathbb{N}: \mathsf{Set} \to \mathsf{CMon}$ preserving coproducts. On the other hand,

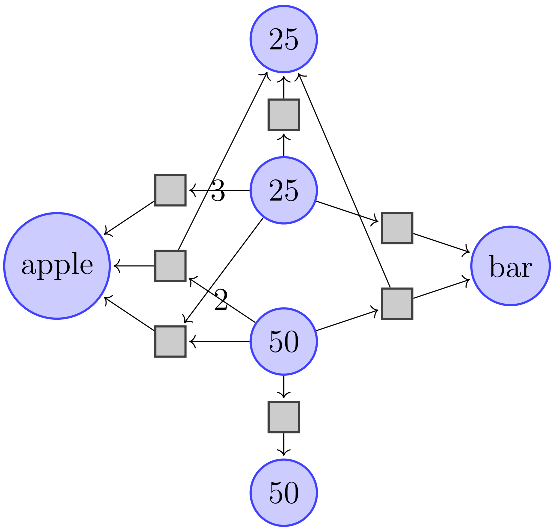
$$\mathrm{Ob} \int K = \{(m, x) \,|\, m \in \mathbb{N}[S] \text{ and } x \in K(m)\}.$$

Because K preserves the monoidal product, $K(m)$ will be isomorphic to the direct sum of commutative monoids $K(p)$ for $p \in S$. Therefore, m will be a tuple of markings $(m_1, m_2, \ldots, m_k)$ where each m_i is an element of the free commutative monoid on the color set for some species. An isomorphism,

$$\mathrm{Ob} \int K \to \mathbb{N}[\hat{S}]$$

of commutative monoids is constructed by sending the pair $(x, (m_1, m_2, \ldots, m_k)$ to the sum $\sum_i m_i$ which is evidently an element of $\mathbb{N}[\hat{S}]$. This map is injective because $\mathbb{N}[\hat{S}]$ is free □

Example 4.3. We can unfold the colored Petri net in Example 2.5 to get the following:



The categorical semantics of this Petri net...

5. Conclusion

Characterizing colored Petri nets categorically will allow them to be fit into the larger program of applied category theory. In particular, we conjecture that using the structured cospan construction [BC19], we can create a compositional framework for open colored Petri nets. These would be colored Petri nets which are equipped with a boundary on their inputs and outputs. One advantage of this would be a categorical description of the reachability problem for colored Petri nets as shown in [BM20].

Appendix A. Double categories

Double Categories. What follows are brief definitions of vertical transformations, monoidal double categories, and lax monoidal double functors. A more detailed exposition can be found in the work of Grandis and Paré [**?**, **?**] missing citation, and for monoidal double categories the work of Shulman [**?**] missing citations. We use ‘double category’ to mean what earlier authors called a ‘pseudo double category’.

Definition A.1. Let $F\colon \mathbb{A} \to \mathbb{B}$ and $G\colon \mathbb{A} \to \mathbb{B}$ be lax double functors. A **vertical transformation** $\beta\colon F \Rightarrow G$ consists of natural transformations $\beta_0\colon F_0 \Rightarrow G_0$ and $\beta_1\colon F_1 \Rightarrow G_1$ (both usually written as β) such that

- $S(\beta_M) = \beta_{SM}$ and $T(\beta_M) = \beta_{TM}$ for any object $M \in A_1$, (I don’t know what kind of font you wanted but slash A isn’t defined.)
- β commutes with the composition comparison, and
- β commutes with the identity comparison.

Shulman defines a 2-category **Dbl** of double categories, double functors, and vertical transformations [**?**] Which Shulman paper?. This has finite products. In any 2-category with finite products we can define a pseudomonoid [DS97], which is a categorification of the concept of monoid. For example, a pseudomonoid in Cat is a monoidal category.

Definition A.2. A **monoidal double category** is a pseudomonoid in **Dbl**. Explicitly, a monoidal double category is a double category equipped with double functors $\otimes: \mathbb{D} \times \mathbb{D} \to \mathbb{D}$ and $I: * \to \mathbb{D}$ where $*$ is the terminal double category, along with invertible vertical transformations called the **associator**:

$$A: \otimes \circ (1_{\mathbb{D}} \times \otimes) \Rightarrow \otimes \circ (\otimes \times 1_{\mathbb{D}}),$$

left unitor:

$$L: \otimes \circ (1_{\mathbb{D}} \times I) \Rightarrow 1_{\mathbb{D}},$$

and **right unitor**:

$$R: \otimes \circ (I \times 1_{\mathbb{D}}) \Rightarrow 1_{\mathbb{D}}$$

satisfying the pentagon axiom and triangle axioms.

This definition neatly packages a large quantity of information. Namely:

- $\mathbb{D}_0$ and $\mathbb{D}_1$ are both monoidal categories.
- If I is the monoidal unit of $\mathbb{D}_0$, then U_I is the monoidal unit of $\mathbb{D}_1$.
- The functors S and T are strict monoidal.
- $\otimes$ is equipped with composition and identity comparisons

$$\chi: (M_1 \otimes N_1) \odot (M_2 \otimes N_2) \xrightarrow{\sim} (M_1 \odot M_2) \otimes (N_1 \odot N_2)$$

$$\mu: U_{A \otimes B} \xrightarrow{\sim} (U_A \otimes U_B)$$

  making three diagrams commute as in Def. **??**.
- The associativity isomorphism for *slashten* (slash ten is undefined) is a vertical transformation between double functors.
- The unit isomorphisms are vertical transformations between double functors.

Definition A.3. A **braided monoidal double category** is a monoidal double category equipped with an invertible vertical transformation

$$\beta: \otimes \Rightarrow \otimes \circ \tau$$

called the **braiding**, where $\tau: \mathbb{D} \times \mathbb{D} \to \mathbb{D} \times \mathbb{D}$ is the twist double functor sending pairs in the object and arrow categories to the same pairs in the opposite order. The braiding is required to satisfy the usual two hexagon identities [**?**, Sec. XI.1]. If the braiding is self-inverse we say that $\mathbb{D}$ is a **symmetric monoidal double category**.

In other words:

- $\mathbb{D}_0$ and $\mathbb{D}_1$ are braided (resp. symmetric) monoidal categories,
- the functors S and T are strict braided monoidal functors, and
- the braiding is a vertical transformation between double functors.

Definition A.4. A **monoidal lax double functor** $F: \mathbb{C} \to \mathbb{D}$ between monoidal double categories $\mathbb{C}$ and $\mathbb{D}$ is a lax double functor $F: \mathbb{C} \to \mathbb{D}$ such that

- F_0 and F_1 are monoidal functors,
- $SF_1 = F_0 S$ and $TF_1 = F_0 T$ are equations between monoidal functors, and
- the composition and unit comparisons $\phi(N_1, N_2): F_1(N_1) \odot F_1(N_2) \to F_1(N_1 \odot N_2)$ and $\phi_A: U_{F_0(A)} \to F_1(U_A)$ are monoidal natural vertical transformations.

The monoidal lax double functor is **braided** if F_0 and F_1 are braided monoidal functors and **symmetric** if they are symmetric monoidal functors.

A fundamental example is the symmetric monoidal double category of spans.

Definition A.5. Let Span denote the double category where

- the category of objects is given by Set: the category of sets and functions,
- the category of morphisms Span_1 has objects given by spans of functions

$$A \longleftarrow X \longrightarrow B$$

  and a morphism from $A \longleftarrow X \longrightarrow B$ to $A' \longleftarrow Y \longrightarrow B'$ is a commuting diagram

$$\begin{tikzcd} A \arrow[d] & X \arrow[l] \arrow[r] \arrow[d] & B \arrow[d] \\ A' & Y \arrow[l] \arrow[r] & B'. \end{tikzcd}$$

- The source and target functors $S, T\colon \mathsf{Span}_1 \to \mathsf{Set}$ send a span $A \longleftarrow X \longrightarrow B$ to A and B respectively.
- The identity assigning functor $U\colon \mathsf{Set} \to \mathsf{Span}_1$ sends a set X to the span $X \xleftarrow{1_x} X \xrightarrow{1_x} X$.
- The composition functor, $\circ\colon \mathsf{Span}_1 \times_{\mathsf{Set}} \mathsf{Span}_1 \to \mathsf{Span}_1$, sends a pair of spans to their pullback and a pair of 2-morphisms to the universal map induced by the pullback.
- The associator, the left unitor and, the right unitor are all given by canonical isomorphisms.

Appendix B. Displayed Categories

Technically, a displayed category is a lax functor into the category of spans. There is a procedure for constructing the category which is being displayed, called the total category. This is a generalization of the Grothendieck construction [GR03]. Like the Grothendieck construction, this gives an equivalence between the category of displayed categories and the arrow category of Cat. The term "displayed category" was introduced in [AL19], but they have been studied before [PA97, Bén00]. In [PA97], they say that this equivalence is somewhat obvious, but a proof could not be found anywhere.

In this appendix, we give the definition of displayed categories and the construction of the total category. In Proposition B.3, we prove that the construction of the total category (and its projection to the base) is functorial. In Proposition B.5, we prove that there is a functorial way of constructing a displayed category from any functor. Finally, we prove the equivalence $\mathsf{Disp} \cong \mathsf{Cat}^{\rightarrow}$ in Theorem B.6. Thus, retrieving this result from the realm of folklore.

Constructing the category which is displayed. Given a displayed category $F\colon X \to \mathsf{Span}$, we construct the **total category**, $\int F$, where

- objects are pairs $(x \in X, c \in Fx)$,
- morphisms are of the form $(f, g)\colon (x, c) \to (y, d)$

where f is a morphism in X, and $g \in Ff$ such that $Ff_x(g) = c$ and $Ff_y(g) = d$. We draw such a morphism as

$$\begin{tikzcd}
 & (Ff, g) \arrow[dl, "Ff_x"'] \arrow[dr, "Ff_y"] & \\
(Fx, c) & & (Fy, d)
\end{tikzcd}$$

and we say that g **witnesses** that F relates c to d. Composition is given by

$$\begin{tikzcd}
 & & & (F(f \circ h), \psi(g,i)) \arrow[dlll, bend right] \arrow[drrr, bend left] & & & \\
 & & & (Ff \times_{Fy} Fh, (g,i)) \arrow[u, "\psi"] \arrow[dll] \arrow[drr] & & & \\
 & (Ff, g) \arrow[dl, "Ff_x"'] \arrow[drr, "Ff_y"] & & & & (Fh, i) \arrow[dll, "Fh_y"'] \arrow[dr, "Fh_z"] & \\
(Fx, c) & & & (Fy, d) & & & (Fz, e)
\end{tikzcd}$$

where we get the right commutativity of this diagram from the coherence conditions of ψ.

There is a canonical projection functor $P_X\colon \int F \to \mathcal{X}$ which is given by forgetting the second entry in both objects and morphisms.

Constructing the functor which is displayed. Let $f\colon x \to y$ in X. The double transformation α in

$$\begin{tikzcd}
X \arrow[dd, "\phi"'] \arrow[dr, "F"] & \\
 \arrow[r, phantom, "\Downarrow \alpha"] & \mathsf{Span} \\
Y \arrow[ur, "G"'] &
\end{tikzcd}$$

has components which look like

$$\begin{tikzcd}
Fx \arrow[d, "\alpha_x"'] & Ff \arrow[l, "Ff_x"'] \arrow[r, "Ff_y"] \arrow[d, "\alpha_f"] & Fy \arrow[d, "\alpha_y"] \\
G\phi x & G\phi f \arrow[l, "G\phi f_x"] \arrow[r, "G\phi f_y"'] & G\phi y
\end{tikzcd}$$

where α_x and α_y are the object components of α, and α_f is the morphism component of α. Commutativity of the squares gives us the following equations.

$$G\phi f_y = \alpha_y \circ Ff_y \circ \alpha_f$$
$$G\phi f_x = \alpha_x \circ Ff_x \circ \alpha_f$$

Given a morphism $(\phi, \alpha)\colon (X, F) \to (Y, G)$ in Disp, we must construct a morphism $[P_X\colon \int F \to X] \to [P_Y\colon \int G \to Y]$ in $\mathsf{Cat}^{\to}$. Such a morphism must be a commutative square of functors of the following form.

$$\begin{tikzcd} \int F \arrow[r, dashed] \arrow[d, "P_X"'] & \int G \arrow[d, "P_Y"] \\ X \arrow[r, "\phi"'] & Y \end{tikzcd}$$

Define the functor

$$\widehat{\alpha}: \int F \to \int G$$

by

$$\widehat{\alpha}(x \in X, c \in Fx) = (\phi x \in Y, \alpha_x c \in G\phi x)$$

on objects, and $\widehat{\alpha}(f,g) = (\phi f, \alpha_f g)$ on morphisms.

$$\begin{tikzcd} & (Ff,g) \arrow[dl, "Ff_x"'] \arrow[dr, "Ff_y"] & & & & (G\phi f, \alpha_f g) \arrow[dl, "G\phi f_x"'] \arrow[dr, "G\phi f_y"] & \\ (Fx,c) & & (Fy,d) & \mapsto & (G\phi x, \alpha_x c) & & (G\phi y, \alpha_y d) \end{tikzcd}$$

Lemma B.1. *$\widehat{\alpha}$ is a functor $\int F \to \int G$.*

Proof. We check that composition is preserved.

$$\begin{aligned} \widehat{\alpha}((h,i) \circ (f,g)) &= \widehat{\alpha}(h \circ f, \psi_?(g,i)) \\ &= (\phi(h \circ f), \alpha_{h \circ f}\psi_?(g,i)) \\ &= (\phi h \circ \phi f, \psi_?(\alpha_f g, \alpha_h i)) \\ &= (\phi h, \alpha_h i) \circ (\phi f, \alpha_f g) \\ &= \widehat{\alpha}(h,i) \circ \widehat{\alpha}(f,g) \end{aligned}$$ □

Lemma B.2. *The following square commutes, so $(\widehat{\alpha}, \phi)$ forms a morphism in $\mathsf{Cat}^{\to}$.*

$$\begin{tikzcd} \int F \arrow[r, "\widehat{\alpha}"] \arrow[d, "P_X"'] & \int G \arrow[d, "P_Y"] \\ X \arrow[r, "\phi"'] & Y \end{tikzcd}$$

Proof. Let $(x,c) \in \int F$. Then

$$P_Y \widehat{\alpha}(x,c) = P_Y(\phi x, \alpha_x(c)) = \phi x = \phi P_X(x,c).$$

For $(f,g) : (x,c) \to (y,d)$ in $\int F$, we get

$$P_Y \widehat{\alpha}(f,g) = P_Y(\phi f, \alpha_f(g)) = \phi f = \phi P_X(f,g).$$ □

Proposition B.3. *$\int : \mathsf{Disp} \to \mathsf{Cat}^{\to}$ is a functor*

Proof. . . . □

The inverse to the total category functor. We now define the inverse to $\int: \mathsf{Disp} \to \mathsf{Cat}^{\to}$, which we call $\mathscr{D}: \mathsf{Cat}^{\to} \to \mathsf{Disp}$. Given a functor $F: C \to X$, define $\mathscr{D}F: X \to \mathsf{Span}$ as follows. For an object x of X, let

$$\mathscr{D}F(x) = F^{-1}(x) = \{c \in C \mid Fc = x\}$$

and for a morphism $f: x \to y$ in X, let $\mathscr{D}F(f)$ be the span

$$\begin{tikzcd}
 & F^{-1}f \arrow[dl, "s_f"'] \arrow[dr, "t_f"] & \\
F^{-1}x & & F^{-1}y
\end{tikzcd}$$

where $F^{-1}f$ is the set of morphisms in C that are assigned to f by F, and s_f and t_f are the appropriate restrictions of the functions which assign source and target objects respectively to each morphism in C.

Given a pair of composable morphisms in C

$$x \xrightarrow{f} y \xrightarrow{g} z$$

F gives us a composable pair of spans.

$$\begin{tikzcd}
 & & F^{-1}f \times_{F^{-1}y} F^{-1}g \arrow[dl] \arrow[dr] & & \\
 & F^{-1}f \arrow[dl, "s_f"'] \arrow[dr, "t_f"] & & F^{-1}g \arrow[dl, "s_g"'] \arrow[dr, "t_g"] & \\
F^{-1}x & & F^{-1}y & & F^{-1}z
\end{tikzcd}$$

The apex of the composite span consists of pairs (ϕ, ψ) of morphisms in C such that $F\phi = f$, $F\psi = g$, and $t_f(\phi) = s_g(\psi)$. So they are composable in C. Since F is a strict functor, then $F(\psi \circ \phi) = g \circ f$, then we can choose the laxator

$$F_{x,y}: F^{-1}f \times_{F^{-1}y} F^{-1}g \to F(g \circ f)$$

to be the function which gives the composite.

The unitor

$$\begin{tikzcd}
 & F^{-1}x \arrow[dl, "1"'] \arrow[dr, "1"] \arrow[dd, "\mathscr{D}F_{1_x}"] & \\
F^{-1}x & & F^{-1}x \\
 & F^{-1}1_x \arrow[ul, "s_{1_x}"] \arrow[ur, "t_{1_x}"'] &
\end{tikzcd}$$

is given by assigning an object $c \in F^{-1}x$ to its identity $1_c \in F^{-1}1_x$. Since F is a strict functor, then $F(1_c) = 1_x$, so this assignment makes sense.

Lemma B.4. *$\mathscr{D}F$ is a lax double functor $X \to \mathsf{Span}$.*

Proof. Probably need to say a bit more about categories being monads in Span before this, or else avoid actually saying this at all. The laxator and unitor come directly from the structure which makes C a monad in Span. The necessary identities follow from axioms of a monad. $\square$

Given a morphism in $\mathsf{Cat}^{\to}$,

$$\begin{tikzcd} C \arrow[r, "\phi_1"] \arrow[d, "F"'] & D \arrow[d, "G"] \\ X \arrow[r, "\phi_2"'] & Y \end{tikzcd}$$

we need to construct the corresponding morphism in Disp.

$$\begin{tikzcd} X \arrow[dr, "\mathscr{D}F"] \arrow[dd, "\phi_2"'] & \\ {} \arrow[r, phantom, "\Downarrow\widehat{\phi}"] & \mathsf{Span} \\ Y \arrow[ur, "\mathscr{D}G"'] & \end{tikzcd}$$

The double transformation $\widehat{\phi}\colon \mathscr{F} \Rightarrow \mathscr{D}G\phi_2$ needs to have components of the following form.

$$\begin{tikzcd} \mathscr{D}Fx \arrow[d, "\widehat{\phi}_x"'] & \mathscr{D}Ff \arrow[l, "s_f"'] \arrow[r, "t_f"] \arrow[d, "\widehat{\phi}_f"] & \mathscr{D}Fy \arrow[d, "\widehat{\phi}y"] \\ \mathscr{D}G\phi_2 x & \mathscr{D}G\phi_2 f \arrow[l, "s_f"] \arrow[r, "t_f"'] & \mathscr{D}G\phi_2 y \end{tikzcd}$$

We abuse the labels s_f and t_f above. If c is an object in C such that $Fc = x$, then by assumption, $\phi_1 c$ is an object in D such that $G(\phi_1 c) = \phi_2 Fc = \phi_2 x$. Thus we define $\widehat{\phi}_f(c) = \phi_1 c$. If k is a morphism in C such that $Fk = f$, then by assumption, $\phi_1 k$ is a morphism in D such that $G(\phi_1 k) = \phi_2 Fk = \phi_2 f$. Thus we define $\widehat{\phi}_f(k) = \phi_1 k$.

Proposition B.5. *$\mathscr{D}$ is a functor* $\mathsf{Cat}^{\to} \to \mathsf{Disp}$.

Proof. Given composable morphisms in $\mathsf{Cat}^{\to}$

$$\begin{tikzcd} C \arrow[r, "\phi_1"] \arrow[d, "F"'] & D \arrow[r, "\psi_1"] \arrow[d, "G"] & E \arrow[d, "H"] \\ X \arrow[r, "\phi_2"'] & Y \arrow[r, "\psi_2"'] & Z \end{tikzcd}$$

we can compose them in $\mathsf{Cat}^{\to}$ and then apply $\mathscr{D}$, giving

$$\begin{tikzcd} C \arrow[r, "\psi_1\phi_1"] \arrow[d, "F"'] & E \arrow[d, "H"] \\ X \arrow[r, "\psi_2\phi_2"'] & Z \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} X \arrow[dr, "\mathscr{D}F"] \arrow[dd, "\psi_2\phi_2"'] & \\ {} \arrow[r, phantom, "\Downarrow\widehat{\psi\phi}"] & \mathsf{Span} \\ Z \arrow[ur, "\mathscr{D}H"'] & \end{tikzcd}$$

or we can first apply $\mathscr{D}$ and then compose them in Disp.

$$\begin{tikzcd} X \arrow[d, "\phi_2"'] \arrow[dr, bend left, "\mathscr{D}F"] \arrow[d, phantom, shift left=4ex, "\Downarrow\widehat{\phi}"] & \\ Y \arrow[r, "\mathscr{D}G"] \arrow[d, "\psi_2"'] \arrow[d, phantom, shift left=4ex, "\Downarrow\widehat{\psi}"] & \mathsf{Span} \\ Z \arrow[ur, bend right, "\mathscr{D}H"'] & \end{tikzcd} \quad \mapsto \quad \begin{tikzcd} X \arrow[dd, "\psi_2\phi_2"'] \arrow[dr, "\mathscr{D}F"] \arrow[d, phantom, shift left=8ex, "(\phi_2 * \widehat{\psi})\widehat{\phi}\Downarrow"] & \\ & \mathsf{Span} \\ Z \arrow[ur, "\mathscr{D}H"'] & \end{tikzcd}$$

We must confirm that the result of these two procedures is the same. The frame is already identical as written. The morphism components of $\widehat{\psi\phi}$ are of the form

$$\widehat{\psi\phi}_f(k) = \psi_1(\phi_1(k)).$$

The morphism components of $(\phi_2 * \widehat{\psi})\widehat{\phi}$ are of the form

$$(\phi_2 * \widehat{\psi})\widehat{\phi}_f(k) = \psi_1(\phi_1(k)). \qquad \square$$

Equivalence. Now we show that $\int$ gives an equivalence by showing that $\mathscr{D}$ is a weak inverse.

Theorem B.6. *$\mathscr{D}$ is a weak inverse to $\int$. In particular,* $\mathsf{Disp} \cong \mathsf{Cat}^{\to}$.

Proof. Let $F: C \to X$ be an object in $\mathsf{Cat}^{\to}$. Define $\varepsilon_F: \int \mathscr{D}F \to C$ by

$$\varepsilon_F(x, c) = c$$
$$\varepsilon_F(f, g) = g.$$

Composition is preserved since the decoration is given by the laxator, and the laxator for $\mathscr{D}$ is given by composition in C. Similar for identities. This is obviously surjective on objects and morphisms. If $\varepsilon_F(x, c) = \varepsilon_F(y, d)$, then $x = F(c) = F(d) = y$. Thus ε_F is injective on objects. Similar for morphisms.

The square

$$\begin{tikzcd} \int \mathscr{D}F \arrow[r, "\varepsilon_F"] \arrow[d, "P_X"'] & C \arrow[d, "F"] \\ X \arrow[r, "1_X"'] & X \end{tikzcd}$$

clearly commutes. We refer to the whole square as ε_F as well. This is an isomorphism in $\mathsf{Cat}^{\to}$.

Given a morphism $\phi: F \to G$ in $\mathsf{Cat}^{\to}$, we see the square

$$\begin{tikzcd} \int \mathscr{D}F \arrow[r, "\varepsilon_F"] \arrow[d, "\widehat{\widehat{\phi}}"'] & C \arrow[d, "\phi_1"] \\ \int \mathscr{D}G \arrow[r, "\varepsilon_G"'] & D \end{tikzcd}$$

commutes because

$$\begin{aligned} \varepsilon_G \widehat{\widehat{\phi}}(x, c) &= \varepsilon_G(\phi_2 x, \widehat{\phi}(c)) \\ &= \varepsilon_G(\phi_2 x, \phi_1(c)) \\ &= \phi_1(c) \\ &= \phi_1 \varepsilon_F(x, c). \end{aligned}$$

Thus, the naturality square commutes.

$$\begin{tikzcd}
\int \mathscr{D}F \arrow[r, "\varepsilon_F"] \arrow[d, swap, "\widehat{\widehat{\phi}}"] & F \arrow[d, "\phi"] \\
\int \mathscr{D}G \arrow[r, swap, "\varepsilon_G"] & G
\end{tikzcd}$$

Define $\eta \colon 1_{\mathsf{Disp}} \to \mathscr{D}\int$

$$\begin{tikzcd}
X \arrow[r, bend left, "F"] \arrow[r, bend right, swap, "\mathscr{D}\int F"] \arrow[r, phantom, "\Downarrow \eta_F"] & \mathsf{Span}
\end{tikzcd}$$

by

$$\begin{aligned}
(\eta_F)_x \colon Fx &\to \mathscr{D}\textstyle\int Fx \\
c &\mapsto (x, c) \\
(\eta_F)_x \colon Ff &\to \mathscr{D}\textstyle\int Ff \\
g &\mapsto (f, g).
\end{aligned}$$

These are clearly isomorphisms of sets.

Check the components are lax natural transformations.

Check it's natural. □

References

[AL19] Benedikt Ahrens and Peter LeFanu Lumsdaine. Displayed categories. *Logical Methods in Computer Science*, 15(1), 2019. Available at arXiv:1705.04296. (Referred to on page 10, 17.)

[BC19] John C. Baez and Kenny Courser. Structured cospans. Available at arxiv:1911.04630, 2019. (Referred to on page 15.)

[Bén00] Jean Bénabou. Distributors at work, 2000. Lecture notes taken by Thomas Streicher available at http://www.mathematik.tu-darmstadt.de/ streicher/FIBR/DiWo.pdf. (Referred to on page 10, 17.)

[BM20] John C. Baez and Jade Master. Open Petri nets. *Mathematical Structures in Computer Science*, 2020. To appear. Available at arXiv:1808.05415. (Referred to on page 4, 15.)

[BMMS01] Roberto Bruni, José Meseguer, Ugo Montanari, and Vladimiro Sassone. Functorial models for Petri nets. *Information and Computation*, 170(2):207–236, 2001. (Referred to on page 3.)

[DMM89] Pierpaolo Degano, José Meseguer, and Ugo Montanari. Axiomatizing net computations and processes. In *Logic in Computer Science, 1989. LICS'89, Proceedings., Fourth Annual Symposium on*, pages 175–185. IEEE, 1989. (Referred to on page 4.)

[DPP10] Robert Dawson, Robert Paré, and Dorette Pronk. The span construction. *Theory and Applications of Categories*, 24(13):302–377, 2010. (Referred to on page 6.)

[DS97] Brian Day and Ross Street. Monoidal bicategories and Hopf algebroids. *Adv. Math.*, 129(1):99–157, 1997. (Referred to on page 15.)

[evi] Principle of equivalence. *web resource*. https://ncatlab.org/nlab/show/principle+of+equivalence. (Referred to on page 3.)

[GLP22] Fabrizio Romano Genovese, Fosco Loregian, and Daniele Palombi. A categorical semantics for bounded petri nets. *Electronic Proceedings in Theoretical Computer Science*, 372:59–71, November 2022. (Referred to on page 3.)

[GR03] Alexander Grothendieck and Michèle Raynaud, editors. *Revêtements étales et groupe fondamental: Séminaire de Géometrie Algébrique du Bois Marie 1960-61, SGA1*. Soc.

Mathématique de France, Paris, 2003. Available at arXiv:0206203. (Referred to on page 10, 17.)

[GS20] Fabrizio Genovese and David I. Spivak. A categorical semantics for guarded Petri nets. In Fabio Gadducci and Timo Kehrer, editors, *Graph Transformation*, pages 57–74, Cham, 2020. Springer International Publishing. Available as arXiv:2002.02762. (Referred to on page 2, 3.)

[Jen81] Kurt Jensen. Coloured Petri nets and the invariant-method. *Theoretical Computer Science*, 14(3):317–336, 1981. (Referred to on page 3, 5.)

[Jen13] Kurt Jensen. *Coloured Petri nets: basic concepts, analysis methods and practical use*, volume 1. Springer Science & Business Media, 2013. (Referred to on page 1, 2, 13.)

[KLPA06] Fabrice Kordon, Alban Linard, and Emmanuel Paviot-Adet. Optimized colored nets unfolding. In *International Conference on Formal Techniques for Networked and Distributed Systems*, pages 339–355. Springer, 2006. (Referred to on page 13.)

[Lak97] Charles Lakos. On the abstraction of coloured Petri nets. In *International Conference on Application and Theory of Petri Nets*, pages 42–61. Springer, 1997. (Referred to on page 1, 5, 9, 10.)

[Lak00] Charles Lakos. Composing abstractions of coloured Petri nets. In *International Conference on Application and Theory of Petri Nets*, pages 323–342. Springer, 2000. (Referred to on page 1.)

[LHY12] Fei Liu, Monika Heiner, and Ming Yang. An efficient method for unfolding colored Petri nets. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–12. IEEE, 2012. (Referred to on page 13.)

[Liu12] Fei Liu. *Colored Petri Nets for Systems Biology*. PhD thesis, Brandenburgischen Technischen Universität Cottbus, 2012. (Referred to on page 13.)

[Mas20] Jade Master. Petri nets based on lawvere theories. *Mathematical Structures in Computer Science*, 30(7):833–864, 2020. (Referred to on page 2, 4, 5.)

[MM90] José Meseguer and Ugo Montanari. Petri nets are monoids. *Information and computation*, 88(2):105–155, 1990. (Referred to on page 1, 2, 4.)

[PA97] D. Pavlović and S. Abramsky. Specifying interaction categories. In *Category Theory and Computer Science*, volume 1290 of *Lecture Notes in Computer Science*, pages 147–158. Springer, Berlin, Heidelberg, 1997. (Referred to on page 11, 17.)

[Sas95] Vladimiro Sassone. On the category of Petri net computations. In *Colloquium on Trees in Algebra and Programming*, pages 334–348. Springer, 1995. (Referred to on page 2, 4.)

[Sas96] Vladimiro Sassone. An axiomatization of the algebra of Petri net concatenable processes. *Theoretical Computer Science*, 170(1-2):277–296, 1996. (Referred to on page 4.)

[Shu08] Michael Shulman. Framed bicategories and monoidal fibrations. *Theory Appl. Categ.*, 20:No. 18, 650–738, 2008. (Referred to on page 7.)

Email address: `jmoeller31415@gmail.com, jadeedenstarmaster@gmail.com`