



# A Categorical Semantics for Guarded Petri Nets

Fabrizio Genovese<sup>1</sup>  and David I. Spivak<sup>2</sup> 

<sup>1</sup> Statebox, Amsterdam, The Netherlands

research@statebox.io

<sup>2</sup> MIT, Boston, USA

dspivak@mit.edu

**Abstract.** We build on the correspondence between Petri nets and free symmetric strict monoidal categories already investigated in the literature, and present a categorical semantics for Petri nets with guards. This comes in two flavors: Deterministic and with side-effects. Using the Grothendieck construction, we show how the guard semantics can be internalized in the net itself.

## 1 Introduction

Category theory has been used to study Petri nets at least since the beginning of the nineties [6]. Throughout this time, the main effort in this direction of research consisted in showing how Petri nets can be thought of as presenting various flavors of free monoidal categories [2, 5, 6, 8]. This idea has been very influential, successfully modeling the individual-token philosophy via process semantics.

On the other hand, shortly after Petri's first publications about the nets that carry his name [7] researchers started investigating what happens when nets are enriched with new features. One of the most successful extensions of Petri nets is *guarded (or coloured) nets* [3]. Modulo different flavors of modeling what boils down to be the same concept, a *guarded net* is a Petri net with the following extra properties:

- To each token is attached some “attribute”. The kind of attributes we can attach to tokens depends on the place the token is in;
- Each arc is decorated with an expression, which modifies tokens' attributes as they flow through the net;
- Each transition is decorated with a predicate and only fires on tokens whose attributes satisfy the predicate.

At a fist glance, guarded nets allow for a more expressive form of modeling with respect to their unguarded counterparts, but as we will see shortly, this is not necessarily the case. Indeed, depending on the underlying theory from which properties, expressions, and predicates are drawn the gain in expressive power with respect to undecorated nets may be nil: With a wise choice of underlying

theory, coloured nets amount to be nothing more than syntactic sugar for standard nets, though of course the availability of such syntactic sugar can greatly simplify the modeling of complex processes using the Petri net formalism.

Recently there has been renewed interest in employing Petri nets as the basis for a programming language [10]. In this setting, the categorical correspondence between nets and symmetric monoidal categories has been of the utmost importance, single-token philosophy being considered necessary to make the programming language usable [9]. Clearly, extending nets with new features such as guards or timings is desirable to make the language more expressive.

In this work we try to unify these two longstanding directions of research – the categorical approach to Petri nets and the study of guarded nets – by showing how guarded nets can be modeled as ordinary Petri nets with a particular flavor of semantics in the style of [1].

Importantly, we are able to define both a *deterministic semantics* and a *non-deterministic semantics* in our formalism. The first models the traditional notion of guards deterministically modifying data attached to tokens, while the second describes a setting where token data is modified depending on side effects.

Using the Grothendieck construction, we show how the guard semantics can be *internalized* in the net itself, providing a categorical proof that in our model, guarded nets do not increase expressivity, as compared to traditional nets. This is a desired feature, since it means that many nice properties of nets such as termination or decidability of the reachability relation are preserved. It also shows that the core mathematical abstraction in computer implementations of Petri nets need not be modified when offering users the flexibility of guarded nets.

We save all proofs for the appendix, which starts on page 14.

## 2 Guarded Nets

Having given an intuitive version of what a guarded net is, we now start modeling the concept formally. We will use the formalism developed in [1], of which we recall some core concepts.

We denote by  $\mathfrak{F}(N)$  the free symmetric strict monoidal category associated to a Petri net  $N$ , and with  $\mathfrak{U}(\mathcal{C})$  the Petri net associated to the free symmetric strict monoidal category  $\mathcal{C}$ . We denote composition in diagrammatic order; i.e. given  $f: c \rightarrow d$  and  $g: d \rightarrow e$ , we denote their composite by  $(f \circ g): c \rightarrow e$ .

**Definition 1.** *Given a strict monoidal category  $\mathcal{S}$ , a Petri net with  $\mathcal{S}$ -semantics is a pair  $(N, N^\sharp)$ , consisting of a Petri net  $N$  and a strict monoidal functor*

$$N^\sharp: \mathfrak{F}(N) \rightarrow \mathcal{S}.$$

*A morphism  $F: (M, M^\sharp) \rightarrow (N, N^\sharp)$  is just a strict monoidal functor  $F: \mathfrak{F}(M) \rightarrow \mathfrak{F}(N)$  such that  $M^\sharp = F \circ N^\sharp$ .*

*Nets equipped with  $\mathcal{S}$ -semantics and their morphisms form a monoidal category denoted  $\mathbf{Petri}^{\mathcal{S}}$ , with the monoidal structure arising from the product in  $\mathbf{Cat}$ .*

**Definition 2.** We denote by  $\mathbf{Set}_*$  the category of sets and partial functions, and by  $\mathbf{Span}$  the 1-category of sets and spans, where isomorphic spans are identified. Both these categories are symmetric monoidal. From now on, we will work with the strictified version of  $\mathbf{Set}_*$  and  $\mathbf{Span}$ , respectively.

*Example 1.* Let  $\mathbf{1}$  denote the terminal symmetric monoidal category. A Petri net with  $\mathbf{1}$ -semantics is just a Petri net. Petri nets are in bijective correspondence with free symmetric strict monoidal categories, so  $\mathbf{Petri}^{\mathbf{1}}$  denotes the usual category of free symmetric strict monoidal categories and strict monoidal functors between them.

**Notation 1.** Recall that a morphism  $A \rightarrow B$  in  $\mathbf{Span}$  consists of a set  $S$  and a pair of functions  $A \leftarrow S \rightarrow B$ . When we need to notationally extract this data from  $f$ , we write

$$A \xleftarrow{f_1} S_f \xrightarrow{f_2} B$$

We sometimes consider the span as a function  $f: S_f \rightarrow A \times B$ , thus we may write  $f(s) = (a, b)$  for  $s \in S_f$  with  $f_1(s) = a$  and  $f_2(s) = b$ .

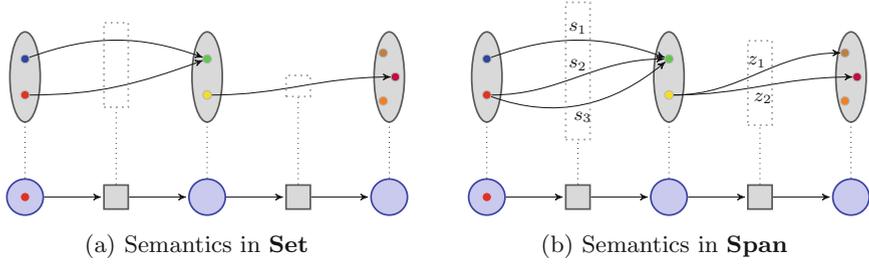
Perhaps unsurprisingly,  $\mathbf{Set}_*$  and  $\mathbf{Span}$  will be the target semantics corresponding to two different flavors for our guards, with  $\mathbf{Span}$  allowing for some form of nondeterminism – expressed as the action of side-effects – whereas  $\mathbf{Set}_*$  models a purely deterministic semantics. Expressing things formally:

**Definition 3.** A guarded net is an object of  $\mathbf{Petri}^{\mathbf{Set}_*}$ . A guarded net with side effects is an object of  $\mathbf{Petri}^{\mathbf{Span}}$ . A morphism of guarded nets (with side effects) is a morphism in  $\mathbf{Petri}^{\mathbf{Set}_*}$  (resp. in  $\mathbf{Petri}^{\mathbf{Span}}$ ).

*Remark 1.* Although it doesn't affect our formalism by any means, in practice the choice of semantics, both for  $\mathbf{Set}_*$  and  $\mathbf{Span}$ , is limited by computational requirements: the places in a net are usually sent to *finite* sets, while transitions are usually sent to computable functions and spans<sup>1</sup>, respectively. Such restrictions are necessary to make sure the net is executable and to keep model checking decidable.

Let us unroll the cryptic Definition 3, starting from the case  $\mathbf{Petri}^{\mathbf{Set}_*}$ . An object in  $\mathbf{Petri}^{\mathbf{Set}_*}$  is a net  $N$  together with a strict monoidal functor  $N^\sharp: \mathfrak{F}(N) \rightarrow \mathbf{Set}_*$ . It assigns to each place  $p$  of  $N$  – corresponding to a generating object of  $\mathfrak{F}(N)$  – a set  $N^\sharp(p)$ , representing all the possible colours a token in  $p$  can assume. A transition  $f: p \rightarrow p'$  – corresponding to a generating morphism of  $\mathfrak{F}(N)$  – gets sent to a partial function  $N^\sharp(f): N^\sharp(p) \rightarrow N^\sharp(p')$ , representing how token colours are transformed during firing. Importantly, the fact that the functions in the semantics are *partial* means that a transition may not be defined for tokens of certain colors. An example of this is the net in Fig. 1a, which is shown together with its semantics. Although reachability in the base net seems quite straightforward, we see that a token in the leftmost place will never reach the rightmost place, since the rightmost transition is not defined on the tokens output by the leftmost one.

<sup>1</sup> A *computable span* is one for which both legs are computable functions.



**Fig. 1.** The same net (below), equipped with a partial function and span semantics, respectively (above).

In the case of  $\mathbf{Petri}^{\mathbf{Span}}$  the intuition is similar. Objects are sent to sets, exactly as in  $\mathbf{Petri}^{\mathbf{Set}_*}$ , but transitions are mapped to spans. Spans can be understood as *relations with witnesses*, provided by elements in the apex of the span. Practically, this means that each path from the span domain to its codomain is indexed by some element of the span apex, as it is shown in Fig. 1b. The presence of witnesses allows to consider different paths between the same elements. Moreover, an element in the domain can be sent to different elements in the codomain via different paths. We interpret this as *non-determinism*: The firing of the transition is not only a matter of the tokens input and output, it also includes the path chosen, which we interpret as having side-effects that are interpreted outside of our model. As one can see, in both Figs. 1a and 1b the composition of paths is the empty function (resp. span). Seeing things from a reachability point of view, the process given by firing the left transition and then the right will never occur. Let us make this intuition precise:

**Definition 4.** Given a guarded Petri net (with side effects)  $(N, N^\sharp)$ , a marking for  $(N, N^\sharp)$  is a pair  $(X, x)$  where  $X$  is an object of  $\mathfrak{F}(N)$  and  $x \in N^\sharp X$ . We say that a marking  $(Y, y)$  is reachable from  $(X, x)$  if there is a morphism  $f : X \rightarrow Y$  in  $\mathfrak{F}(N)$  such that  $N^\sharp f(x) = y$ .

The goal we will pursue in the next section will be to internalize the guard semantics in the free category  $\mathfrak{F}(N)$  associated to a net.

### 3 Internalizing Guards

By “internalizing the semantics of a guarded net  $N$  in  $\mathfrak{F}(N)$ ” we mean *obtaining an unguarded net  $M$  such that  $\mathfrak{F}(M)$  represents all the possible runs of  $N$* . For readers familiar with coloured Petri nets, this corresponds to the claim that reachability in a coloured net is equivalent to reachability in a suitably constructed “standard” net [3].

Since our point of view is process-theoretic, and we are working with symmetric strict monoidal categories and functors, such internalization must be built categorically. The main tool we will use is the *Grothendieck construction* [4], which in our context we will specialize to functors to  $\mathbf{Set}_*$  and  $\mathbf{Span}$ , respectively.

**Definition 5.** Let  $(M, M^\sharp) \in \mathbf{Petri}^{\mathit{Set}_*}$  be a guarded net. We define its internalization, denoted  $\int M^\sharp$ , as the following category:

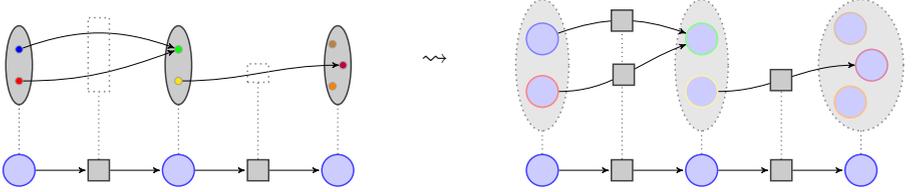
- The objects of  $\int M^\sharp$  are pairs  $(X, x)$ , where  $X$  is an object of  $\mathfrak{F}(M)$  and  $x$  is an element of  $M^\sharp X$ . Concisely:

$$\text{Obj } \int M^\sharp := \{(X, x) \mid (X \in \text{Obj } \mathfrak{F}(M)) \wedge (x \in M^\sharp X)\}.$$

- A morphism from  $(X, x)$  to  $(Y, y)$  in  $\int M^\sharp$  is a morphism  $f: X \rightarrow Y$  in  $\mathfrak{F}(M)$  such that  $x$  is sent to  $y$  via  $M^\sharp f$ . Concisely:

$$\text{Hom } \int M^\sharp [(X, x), (Y, y)] := \{f \mid (f \in \text{Hom}_{\mathfrak{F}(M)} [X, Y]) \wedge (M^\sharp f(x) = y)\}.$$

It is worth giving some intuition of what the Grothendieck construction does in our context. It basically makes a place for each element of the set we send a place to, and makes a transition for each path between these elements, as shown below:



An equivalent definition exists when the semantics is taken to be in **Span**, which is the following:

**Definition 6.** Let  $(M, M^\sharp) \in \mathbf{Petri}^{\mathit{Span}}$  be a guarded net with side effects. We define the internalization of  $(M, M^\sharp)$ , denoted with  $\int M^\sharp$ , as the following category:

- The objects of  $\int M^\sharp$  are pairs  $(X, x)$ , where  $X$  is an object of  $\mathfrak{F}(M)$  and  $x$  is an element of  $M^\sharp X$ . Concisely:

$$\text{Obj } \int M^\sharp := \{(X, x) \mid (X \in \text{Obj } \mathfrak{F}(M)) \wedge (x \in M^\sharp X)\}.$$

- A morphism from  $(X, x)$  to  $(Y, y)$  in  $\int M^\sharp$  is a pair  $(f, s)$  where  $f: X \rightarrow Y$  in  $\mathfrak{F}(M)$  and  $s \in S_{M^\sharp f}$  in the apex of the corresponding span connects  $x$  to  $y$ . Concisely:

$$\text{Hom } \int M^\sharp [(X, x), (Y, y)] := \{(f, s) \mid (f \in \text{Hom}_{\mathfrak{F}(M)} [X, Y]) \wedge (s \in S_{M^\sharp f}) \wedge (M^\sharp f(s) = (x, y))\}.$$

The intuition in the span case is exactly as for partial functions, and we don't deem it useful to draw the same picture again. Looking at the example, though, a couple of things become clear. The first is that to justify the idea of the Grothendieck construction turning an assignment of semantics into a net we have to prove that the resulting category is symmetric strict monoidal and free. The second is that the net thus built is fibered over the base net, and there should be an opposite construction sending  $\int M^\sharp$  to  $M$ . Both of these claims are true, as we now prove:

**Lemma 1.** *In the case of both  $\mathbf{Set}_*$  and  $\mathbf{Span}$ , the category  $\int M^\sharp$  has a strict symmetric monoidal structure.*

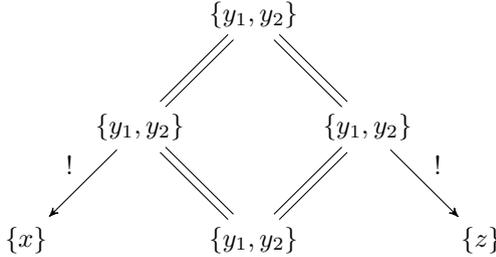
**Theorem 1.** *In both the case of  $\mathbf{Set}_*$  and of  $\mathbf{Span}$  the strict symmetric monoidal category  $\int M^\sharp$  is free.*

**Counterexample 1** (Relations). *Theorem 1 does not hold – the Grothendieck construction does not yield a free symmetric strict monoidal category – if we replace  $\mathbf{Set}_*$  or  $\mathbf{Span}$  with  $\mathbf{Rel}$ . To see this, consider  $\int M^\sharp$  in the case that  $M^\sharp: \mathfrak{F}(M) \rightarrow \mathbf{Rel}$ . Let  $M$  be the Petri net consisting of three places  $X, Y, Z$  and two transitions  $f: X \rightarrow Y$  and  $g: Y \rightarrow Z$ . Let  $M^\sharp$  send  $X$  to  $\{x\}$ ,  $Y$  to  $\{y_1, y_2\}$ , and  $Z$  to  $\{z\}$ . On morphisms, let  $M^\sharp$  send  $f$  to the maximal relation on  $\{x\} \times \{y_1, y_2\}$  and  $g$  to the maximal relation on  $\{y_1, y_2\} \times \{z\}$ . Then we have the following four generating morphisms in  $\int M^\sharp$ :*

$$\begin{aligned} f_1: (X, x) &\rightarrow (Y, y_1) & f_2: (X, x) &\rightarrow (Y, y_2) \\ g_1: (Y, y_1) &\rightarrow (Z, z) & g_2: (Y, y_2) &\rightarrow (Z, z) \end{aligned}$$

There is an equality  $f_1 \circ g_1 = f_2 \circ g_2$  as morphisms  $(X, x) \rightarrow (Z, z)$  in  $\int M^\sharp$ , proving  $\int M^\sharp$  is not free.

The reason that Theorem 1 holds in the span case is that spans keep track of different paths between elements, whereas relations do not. To see this, consider the span composition:



It is clear that in this composition the two paths from  $x$  to  $z$  are considered as separated in the  $\mathbf{Span}$  case, and witnessed by  $y_1, y_2$  respectively, while in the case of  $\mathbf{Rel}$  they would have been conflated to one. The result is that these paths correspond to the same morphism in the relational case of  $\int M^\sharp$ , introducing new equations and breaking freeness, while they stay separated in the span case.

**Lemma 2.** *In the case of both  $\mathbf{Set}_*$  and  $\mathbf{Span}$ , there is a strict monoidal functor  $\pi_M: \int M^\sharp \rightarrow \mathfrak{F}(M)$  sending  $(X, x)$  to  $X$  and  $f: (X, x) \rightarrow (Y, y)$  to  $f: X \rightarrow Y$  (resp.  $(f, s): (X, x) \rightarrow (Y, y)$  to  $f: X \rightarrow Y$ ).*

*Remark 2.* In general,  $\pi_M$  is not an opfibration. This is because our target categories  $\mathbf{Set}_*$  and  $\mathbf{Span}$  allow for partial functions. Indeed, if  $f: X \rightarrow Y$  in  $M$  is sent by  $M^\sharp$  to a partial function that is not defined on  $x \in M^\sharp X$ , then there is no coCartesian lift emanating from  $(X, x)$  for the morphism  $f$ .

We conclude this section by proving that the reachability semantics of a guarded net coincides with the reachability semantics of its internalization.

**Theorem 2.** *Let  $(N, N^\sharp)$  be a guarded Petri net (with side effects).  $(Y, y)$  is reachable from  $(X, x)$  if and only if  $(Y, y)$  is reachable from  $(X, x)$  in the net  $\mathfrak{U}(\int N^\sharp)$ .*

## 4 Properties of Internalizations

The Grothendieck construction provides a way to internalize partial function and span semantics to nets. As such, it acts on objects of the categories  $\mathbf{Petri}^{\mathbf{Set}^*}$  and  $\mathbf{Petri}^{\mathbf{Span}}$ , respectively. It is thus worth asking what happens to morphisms in these categories. The answer is, luckily, easy to find:

**Lemma 3.** *Let  $F: (M, M^\sharp) \rightarrow (N, N^\sharp)$  be a morphism in  $\mathbf{Petri}^{\mathbf{Set}^*}$  (resp. in  $\mathbf{Petri}^{\mathbf{Span}}$ ). Then it lifts to strict monoidal functor  $\bar{F}: \int M^\sharp \rightarrow \int N^\sharp$  (resp.  $\hat{F}: \int M^\sharp \rightarrow \int N^\sharp$ ), such that the following diagram on the left (resp. on the right) commutes:*

$$\begin{array}{ccc}
 \int M^\sharp & \overset{\bar{F}}{\dashrightarrow} & \int N^\sharp \\
 \pi_M \downarrow & & \downarrow \pi_N \\
 \mathfrak{F}(M) & \xrightarrow{F} & \mathfrak{F}(N) \\
 M^\sharp \searrow & & \swarrow N^\sharp \\
 & \mathbf{Set}_* & 
 \end{array}
 \qquad
 \begin{array}{ccc}
 \int M^\sharp & \overset{\hat{F}}{\dashrightarrow} & \int N^\sharp \\
 \pi_M \downarrow & & \downarrow \pi_N \\
 \mathfrak{F}(M) & \xrightarrow{F} & \mathfrak{F}(N) \\
 M^\sharp \searrow & & \swarrow N^\sharp \\
 & \mathbf{Span} & 
 \end{array}$$

**Notation 2.** *The notation for the liftings in Lemma 3 is easy to remember: The arrow over  $\bar{F}$  looks like a stylized function, while the hat over  $\hat{F}$  looks like a stylized span.*

The lifting of Lemma 3 is quite well-behaved. First of all, it is worth stressing how it preserves some relevant categorical properties:

**Lemma 4.** *For any map  $F: (M, M^\sharp) \rightarrow (N, N^\sharp)$  in  $\mathbf{Petri}^{\mathbf{Set}^*}$  (respectively in  $\mathbf{Petri}^{\mathbf{Span}}$ ), the functor  $F$  is faithful if and only if  $\bar{F}$  is faithful (resp.  $\hat{F}$  is faithful). If  $F$  is full, then so is  $\bar{F}$  (resp.  $\hat{F}$ ).*

Having ascertained that “basic” categorical properties are preserved, it is worth asking what happens to particular classes of functors in  $\mathbf{Petri}^{\mathbf{Set}^*}$  and  $\mathbf{Petri}^{\mathbf{Span}}$ , respectively.

Following [1], there are three relevant kinds of morphisms in a category of Petri nets with semantics. On one hand there are transition-preserving functors, which represent morphisms of free monoidal categories arising purely from the topological structure of their underlying net. On the other there are functors

representing glueings of nets, which are themselves divided into synchronizations (defined in terms of addition and erasing of generators, that is, double pushouts) and identifications (defined in terms of pushouts). Let us investigate which ones of these properties are preserved.

**Definition 7.** A strict symmetric monoidal functor  $F$  between FSSMCs  $\mathcal{C}$ ,  $\mathcal{D}$  is said to be transition-preserving when each generating morphism  $f$  of  $\mathcal{C}$  is mapped to  $\sigma \circledast g \circledast \sigma'$  for some generating morphism  $g$  of  $\mathcal{D}$  and symmetries  $\sigma, \sigma'$ .

**Lemma 5.** If  $F$  is transition-preserving, so are  $\vec{F}$  and  $\widehat{F}$ .

**Lemma 6.** If  $F$  is injective on objects, so are  $\vec{F}$  and  $\widehat{F}$ .

Another interesting class of morphisms is *identifications*:

**Definition 8.** A Petri net  $(N, N^\sharp)$  is said to be an identification of  $(M, M^\sharp)$  if there is a morphism  $F : (M, M^\sharp) \rightarrow (N, N^\sharp)$  such that:

- There is a Petri net  $O$ , and a pair of transition-preserving functors  $l, r : \mathfrak{F}(O) \rightarrow \mathfrak{F}(M)$ ;
- $l \circledast M^\sharp = r \circledast M^\sharp$ ; and
- $F$  is the coequalizer of  $l$  and  $r$ .

Identifications are also preserved. The ultimate reason for this is that identifications are defined purely in terms of coequalizers of transition-preserving functors, which are preserved by the Grothendieck construction.

**Lemma 7.** If  $(N, N^\sharp)$  is an identification of  $(M, M^\sharp)$  via  $F$  and witnesses  $O, l, r$ , then  $\int N^\sharp$  is an identification of  $\int M^\sharp$  via  $\vec{F}$  and witnesses  $\mathfrak{U}(\int (l \circledast M^\sharp))$ ,  $\vec{l}$ ,  $\vec{r}$ . The span case is analogous.

Preservation of identifications also entails that addition of generators for a net are preserved by internalizations.

**Definition 9.** A net  $(M, M^\sharp)$  is an addition of generating morphisms to  $(K, K^\sharp)$  via  $W, w$  if:

- There is a net  $W$  together with a strict monoidal functor  $w : \mathfrak{F}(W) \rightarrow \mathfrak{F}(K)$  which sends generating objects to generating objects, is injective on objects and faithful;
- $\mathfrak{F}(M)$  is the pushout of  $\mathfrak{F}(\overline{W}) \hookrightarrow \mathfrak{F}(W) \xrightarrow{w} \mathfrak{F}(K)$  and  $\mathfrak{F}(\overline{W}) \hookrightarrow \mathfrak{F}(W)$ , where  $\overline{W}$  denotes the net with the same places of  $W$  and no transitions; and
- $M^\sharp$  arises from the universal property of the pushout.

**Lemma 8.** *Let  $(M, M^\sharp)$  be an addition of generating morphisms to  $(K, K^\sharp)$  via  $W, w$ . Then  $\int M^\sharp$  is an addition of generating morphisms to  $\int K^\sharp$  via  $\mathfrak{U}(\int (w \circlearrowleft K^\sharp)), \bar{w}$ . The span case is analogous.*

Finally, we investigate what happens when considering erasings of generators from a net. To do this, we first follow [1] to define subnets:

**Definition 10.** *Given Petri nets  $N, N_w$ , we say that  $N_w$  is a subnet of  $N$  if its places and transitions are a subset of places and transitions of  $N$ , and input and output functions on  $N_w$  are restrictions of the input and output functions on  $N$ . If  $N_w$  is a subnet of  $N$ , then there is an obvious identity on objects, identity on morphisms strict monoidal functor,  $\iota : \mathfrak{F}(N_w) \hookrightarrow \mathfrak{F}(N)$  between their associated free symmetric strict monoidal categories. From this, we say that a net  $(N_w, N_w^\sharp)$  is a subnet of  $(N, N^\sharp)$  if  $N_w$  is a subnet of  $N$  and  $N_w^\sharp = \iota \circlearrowleft N^\sharp$ .*

This enables us to define erasings of generators:

**Definition 11.** *Let  $(N_w, (N_w)^\sharp)$  be a subnet of  $(N, N^\sharp)$ . An erasing of generators of  $(N, N^\sharp)$  via  $N_w$  is a net  $(K, K^\sharp)$  such that:*

- $(K, K^\sharp)$  is a subnet of  $(N, N^\sharp)$ ;
- $(\overline{N_w}, \overline{(N_w)^\sharp})$ , where  $\overline{N_w}$  denotes the net with the same places of  $N_w$  and no transitions, is a subnet of  $(K, K^\sharp)$ ; and
- $\mathfrak{F}(N)$  is the pushout of  $\mathfrak{F}(\overline{N_w}) \hookrightarrow \mathfrak{F}(N_w)$ ; and  $\mathfrak{F}(\overline{N_w}) \hookrightarrow \mathfrak{F}(K)$ .

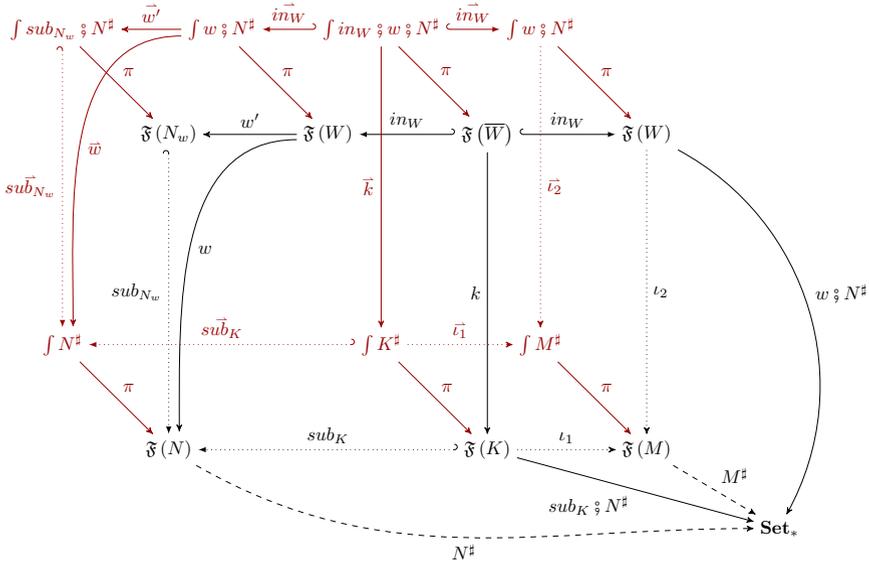
Indeed, erasings of generators are preserved as well by our internalization:

**Lemma 9.** *Let  $(K, K^\sharp)$  be an erasing of generating morphisms from  $(N, N^\sharp)$  via a subnet  $N_w$ . Then  $\int K^\sharp$  is an erasing of generators from  $\int N^\sharp$  via  $\int \text{sub}_{N_w} \circlearrowleft N^\sharp$ . The span case is analogous.*

Surprisingly, even if erasing and addition of generators are preserved by internalizations, synchronizations are not. Indeed, following [1],  $(M, M^\sharp)$  is a synchronization of  $(N, N^\sharp)$  via  $W, w$  when  $\mathfrak{F}(M)$  is defined to be the result of applying the following double pushout rewrite rule to  $\mathfrak{F}(N)$ :

$$\mathfrak{F}(N_w) \xleftarrow{w'} \mathfrak{F}(W) \xleftarrow{in_w} \mathfrak{F}(\overline{W}) \xrightarrow{in_w} \mathfrak{F}(W)$$

Here, we require that  $w$  factorizes through  $w'$ . In internalizing this construction the pushouts are preserved, but the rewrite rule is not! This becomes evident by lifting the definition of synchronization altogether, where in the following diagram we are sticking to the notation developed in [1]:



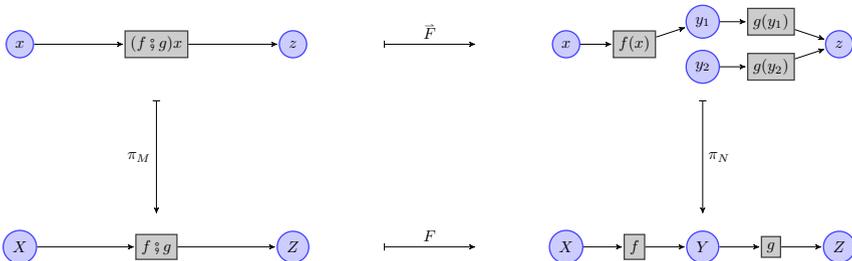
The black arrows are just the definition of synchronization. The dotted arrows denote the pushout arrows, while the dashed arrows arise from the universal property of the pushout. The maroon arrows and objects represent the Grothendieck construction and the lifting of the functors obtained from Lemma 3, while the  $\pi$ s stand for the functors obtained in Lemma 2, where we omitted subscripts to avoid clutter.

As one can see the pushout squares are both preserved, but  $\int M^\sharp$  is not a synchronization of  $\int N^\sharp$  via  $\int in_W \circledast w \circledast N^\sharp$  since

$$\int sub_{N_w} \circledast N^\sharp \neq (\int N^\sharp) \int w \circledast N^\sharp$$

In other words,  $\int sub_{N_w} \circledast N^\sharp$  is too big of a subcategory of  $\int N^\sharp$  to make  $\int M^\sharp$  into a synchronization. An analogous observation holds for spans.

**Counterexample 2** (Synchronizations not preserved). *We provide a practical counterexample of why synchronizations are not preserved by internalizations. Consider the following nets, where we are borrowing the graphical notation developed in [1], decorating net elements with their images in  $\mathbf{Set}_*$ .*



At the base level we have two nets,  $M$  on the left and  $N$  on the right. Eliding the functor  $N^\sharp$ , the places of  $N$  are mapped to sets:

$$X := \{x\} \quad Y := \{y_1, y_2\} \quad Z := \{z\}$$

While transitions are mapped to partial functions  $f: X \rightarrow Y$  and  $g: Y \rightarrow Z$ , defined as follows:

$$f(x) = y_1 \quad g(y_1) = g(y_2) = z$$

$M$  is clearly a synchronization of  $N$  via  $F$ : The generators  $f$  and  $g$  have been erased and a generator corresponding to  $f \circledast g$  has been added. Taking the Grothendieck construction on  $M$  and  $N$  (top left and top right in the figure, respectively), we see how the erasing of generators is problematic: The morphism  $g$  in  $N$  branches into  $g(y_1)$  and  $g(y_2)$  in  $\int N^\sharp$ , of which only  $g(y_1)$  forms a path with  $f(x)$ . In lifting the synchronization  $M$  to  $\int M^\sharp$ , we would expect  $g(y_1)$  and  $f(x)$  to be erased and conflatd into  $f(x) \circledast g(y_1)$ , whereas  $g(y_2)$  stays. But this is not the case, since in  $M$  the generator  $g$  has already been erased “before being allowed to branch”, taking  $g(y_2)$  with it when we take  $\int M^\sharp$ !

As we said before, this ultimately depends on the fact that the internalization of the subnet provided by the synchronization witness contains too many morphisms, and ends up erasing more generators than we would like it to.

## 5 Internalization as a Functor

In this final section, we put together some of the properties we have proved so far about internalizations, and prove that internalization is a functor. The intuitive argument behind the results that are about to follow is this: If  $\int N^\sharp$  internalizes the semantics of  $(N, N^\sharp)$ , in either the case  $N^\sharp: N \rightarrow \mathbf{Set}_*$  or  $N^\sharp: N \rightarrow \mathbf{Span}$ , then  $\int N^\sharp$  should be considered as “just a net”, that is, an object of  $\mathbf{Petri}^1$ ; see Example 1.

Putting together results about lifting of functors obtained in the previous section, we are indeed able to prove this.

**Theorem 3.** *Denote with  $\mathbf{1}$  the terminal category, together with the trivial symmetric monoidal structure on it. There is a faithful, strong monoidal functor  $\mathit{emb}_{\mathbf{Set}_*}: \mathbf{Petri}^{\mathbf{Set}_*} \rightarrow \mathbf{Petri}^{\mathbf{1}}$  defined as follows:*

- On objects, it sends  $(M, M^\sharp)$  to  $(\mathfrak{U}(\int M^\sharp), \mathfrak{U}(\int M^\sharp)^\sharp)$ .
- On morphisms, we send the functor  $F: (M, M^\sharp) \rightarrow (N, N^\sharp)$  to the functor<sup>2</sup>

$$\hat{F}: (\mathfrak{U}(\int M^\sharp), \mathfrak{U}(\int M^\sharp)^\sharp) \rightarrow (\mathfrak{U}(\int N^\sharp), \mathfrak{U}(\int N^\sharp)^\sharp)$$

Similarly, there is a faithful, strong monoidal functor  $\mathit{emb}_{\mathbf{Span}}: \mathbf{Petri}^{\mathbf{Span}} \rightarrow \mathbf{Petri}^{\mathbf{1}}$  defined as follows:

<sup>2</sup> To be absolutely precise, we are referring to the functor  $\mathfrak{F}(\mathfrak{U}(\int M^\sharp)) \simeq \int M^\sharp \xrightarrow{\hat{F}} \int N^\sharp \simeq \mathfrak{F}(\mathfrak{U}(\int N^\sharp))$ .

- On objects, it sends  $(M, M^\#)$  to  $(\mathfrak{U}(f M^\#), \mathfrak{U}(f M^\#)^\#)$ .
- On morphisms, we send the functor  $F: (M, M^\#) \rightarrow (N, N^\#)$  to the functor<sup>3</sup>

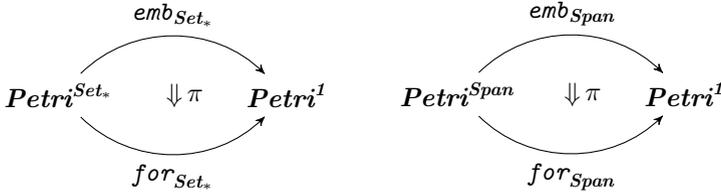
$$\hat{F}: (\mathfrak{U}(f M^\#), \mathfrak{U}(f M^\#)^\#) \rightarrow (\mathfrak{U}(f N^\#), \mathfrak{U}(f N^\#)^\#)$$

Finally, it is worth noting that for each choice of semantics  $\mathcal{S}$  there is another obvious functor from  $\mathbf{Petri}^{\mathcal{S}}$  to  $\mathbf{Petri}^1$ , which just forgets the semantics altogether. It is worth asking how this functor and the ones provided in Theorem 3 are related.

**Proposition 1.** *Denote with*

$$for_{Set_*}: \mathbf{Petri}^{Set_*} \rightarrow \mathbf{Petri}^1 \quad for_{Span}: \mathbf{Petri}^{Span} \rightarrow \mathbf{Petri}^1$$

the “forgetful” functors defined by sending each Petri net  $(M, M^\#)$  to  $(M, M^\#)$ . Then there are natural transformations:



## 6 Conclusion and Future Work

In this work, we described guarded Petri nets as Petri nets endowed with a functorial semantics. We provided two different styles of semantics: a deterministic one, realized using the category of sets and partial functions, and a non-deterministic one that allows for side effects, realized using the category of partial functions and that of spans.

We moreover showed how, using the Grothendieck construction, the guards can be internalized, obtaining a Petri net whose reachability relation is equivalent to the one of the guarded one. We proved that internalizations have nice properties, and the internalization construction is functorial in the choice of the guarded net we start from.

Regarding directions of future work, a pretty straightforward thing to do would be to figure out which semantics, other than  $\mathbf{Set}_*$  and  $\mathbf{Span}$ , are internalizable. That is, if  $F: \mathfrak{F}(N) \rightarrow \mathcal{S}$  is a symmetric monoidal functor, which properties do  $\mathcal{S}$  and  $F$  need to have so that  $\int F$  is a free symmetric strict monoidal category.

---

<sup>3</sup> To be absolutely precise, we are referring to the functor  $\mathfrak{F}(\mathfrak{U}(f M^\#)) \simeq \int M^\# \xrightarrow{\hat{F}} \int N^\# \simeq \mathfrak{F}(\mathfrak{U}(f N^\#))$ .

**Acknowledgements.** David Spivak acknowledges support from Honeywell Inc. as well as from AFOSR grants FA9550-17-1-0058 and FA9550-19-1-0113. Fabrizio Genese wants to thank his fellow team members at Statebox for useful discussion and support.

## Appendix – Proofs

**Lemma 1.** *In the case of both  $\mathbf{Set}_*$  and  $\mathbf{Span}$ , the category  $\int M^\sharp$  has a strict symmetric monoidal structure.*

*Proof.* We start with the case of  $\mathbf{Set}_*$ . Since  $M^\sharp$  is strict monoidal,  $M^\sharp(X \otimes Y) = M^\sharp X \times M^\sharp Y$ . Thus on objects, we can set  $(X, x) \otimes (Y, y) := (X \otimes Y, (x, y))$ . On morphisms, we just use the monoidal product  $f \otimes g$  from  $\mathfrak{F}(M)$ . The monoidal unit is  $(I, *)$ , where  $I$  is the monoidal unit of  $\mathfrak{F}(M)$  and  $*$  is the unique element of the monoidal unit  $\{*\}$  of  $\mathbf{Set}_*$ . The rest of the proof is a straightforward check.

Now we consider the case of  $\mathbf{Span}$ . On objects, we set again  $(X, x) \otimes (Y, y) := (X \otimes Y, (x, y))$ . On morphisms, we set  $(f, s) \otimes (g, t) := (f \otimes g, (s, t))$ , where  $f \otimes g$  is as in  $\mathfrak{F}(M)$  and  $(s, t)$  is the pair of span-apex elements:

$$\begin{array}{ccc} M^\sharp X & \xleftarrow{f_1} S & \xrightarrow{f_2} M^\sharp X' \\ (x, y) & \xleftarrow{(f_1, g_1)} (s, t) & \xrightarrow{(f_2, g_2)} (x', y') \\ M^\sharp Y & \xleftarrow{g_1} T & \xrightarrow{g_2} M^\sharp Y' \end{array}$$

The monoidal unit is  $(I, *)$ , where  $I$  is the monoidal unit of  $\mathfrak{F}(M)$  and  $*$  is the unique element of the monoidal unit  $\{*\}$  of  $\mathbf{Span}$ .

The remainder of the proof is as in the previous case.  $\square$

**Theorem 1.** *In both the case of  $\mathbf{Set}_*$  and of  $\mathbf{Span}$  the strict symmetric monoidal category  $\int M^\sharp$  is free.*

*Proof.* We start with the case of  $\mathbf{Set}_*$ . Consider the free symmetric strict monoidal category  $\hat{M}$  generated as follows:

- Object generators are pairs  $(X, x)$ , with  $X$  object generator in  $\mathfrak{F}(M)$  and  $x \in M^\sharp X$ ;
- A morphism generator  $(X, x) \rightarrow (Y, y)$  is a morphism generator  $f: X \rightarrow Y$  of  $\mathfrak{F}(M)$  such that  $M^\sharp f(x) = y$ .

We want to prove that  $\hat{M}$  and  $\int M^\sharp$  are isomorphic.

First, let  $(X, x)$  be an object in  $\int M^\sharp$ . Then  $X$  is an object of  $\mathfrak{F}(M)$ , which is free, and hence we have  $X = X_1 \otimes \dots \otimes X_n$  for generating objects  $X_1 \dots X_n$  in  $\mathfrak{F}(M)$ . By definition, we have  $x \in M^\sharp X$ . Being  $M^\sharp$  strict, this means:

$$\begin{aligned} x \in M^\sharp X &\Leftrightarrow x \in M^\sharp(X_1 \otimes \dots \otimes X_n) \\ &\Leftrightarrow x \in M^\sharp X_1 \times \dots \times M^\sharp X_n \\ &\Leftrightarrow \exists!(x_1 \in M^\sharp X_1), \dots, \exists!(x_n \in M^\sharp X_n). (x = (x_1, \dots, x_n)) \end{aligned}$$

Hence  $(X, x) = (X_1, x_1) \otimes \dots \otimes (X_n, x_n)$ , and the object generators of  $\int M^\sharp$  are the pairs  $(X, x)$  with  $X$  object generator in  $\mathfrak{F}(M)$  and  $x \in M^\sharp X$ . This means that there is a bijection on objects of  $\hat{M}$  and  $\int M^\sharp$ :

$$(X_1, x_1) \otimes \dots \otimes (X_n, x_n) \mapsto (X_1 \otimes \dots \otimes X_n, (x_1, \dots, x_n))$$

We can then define a symmetric monoidal functor  $T: \hat{M} \rightarrow \int M^\sharp$  that is bijective on objects, and sends a generating morphism  $f: (X, x) \rightarrow (Y, y)$  of  $\hat{M}$  to the morphism  $f: (X, x) \rightarrow (Y, y)$  in  $\int M^\sharp$ .

We want to prove that  $T$  is full and faithful. Faithfulness is obvious; given  $f_1, f_2: (X, x) \rightarrow (Y, y)$  in  $\hat{M}$ , if  $T(f_1) = T(f_2)$  then in particular  $f_1 = f_2$  in  $\mathfrak{F}(M)$ . It follows from the fact that  $M^\sharp$  is at most single-valued – i.e.  $M^\sharp g(x) = x'$  and  $M^\sharp g(x) = x''$  imply  $x' = x''$  – that  $f_1 = f_2$  also in  $\hat{M}$ . For fullness, take a morphism  $f: (X, x) \rightarrow (Z, z)$  in  $\int M^\sharp$ , and notice the following:

- If  $f: (X, x) \rightarrow (Z, z)$  is equal to  $f_1 \circledast f_2$ , where  $f_1: X \rightarrow Y$  and  $f_2: Y \rightarrow Z$ , then we have:

$$z = M^\sharp f(x) = M^\sharp (f_1 \circledast f_2)(x) = (M^\sharp f_1 \circledast M^\sharp f_2)(x)$$

So there is a  $y \in M^\sharp Y$  such that  $M^\sharp f_1(x) = y$  and  $M^\sharp f_2(y) = z$ . This means that  $f_1: (X, x) \rightarrow (Y, y)$  and  $f_2: (Y, y) \rightarrow (Z, z)$  are morphisms in  $\int M^\sharp$ ;

- If  $f: (X_1 \otimes X_2, (x_1, x_2)) \rightarrow (Y_1 \otimes Y_2, (y_1, y_2))$  is equal to  $f_1 \otimes f_2$ , where  $f_1: X_1 \rightarrow Y_1$  and  $f_2: X_2 \rightarrow Y_2$ , then we have:

$$(y_1, y_2) = M^\sharp f(x_1, x_2) = M^\sharp (f_1 \otimes f_2)(x_1, x_2) = (M^\sharp f_1 \times M^\sharp f_2)(x_1, x_2)$$

This means that  $M^\sharp f_1(x_1) = y_1$  and  $M^\sharp f_2(x_2) = y_2$ , and hence that  $f_1: (X_1, x_1) \rightarrow (Y_1, y_1)$  and  $f_2: (X_2, x_2) \rightarrow (Y_2, y_2)$  are morphisms in  $\int M^\sharp$ .

By definition, since  $\mathfrak{F}(M)$  is free, any morphism  $f: X \rightarrow Z$  can be decomposed into a composition of monoidal products of morphism generators, symmetries and identities. The points above prove that  $f: (X, x) \rightarrow (Z, z)$  can be decomposed in the same way, and hence is in the image of  $T$ ; thus it is full.

Our correspondence is bijective on objects and fully faithful, proving that  $\hat{M}$  and  $\int M^\sharp$  are isomorphic as categories. Since  $\hat{M}$  is free so is  $\int M^\sharp$ , completing the proof.

We now consider the case of **Span**. The structure of the proof is similar. Consider the free symmetric strict monoidal category  $\hat{M}$  generated as follows:

- Object generators are pairs  $(X, x)$ , with  $X$  object generator in  $\mathfrak{F}(M)$  and  $x \in M^\sharp X$ ;
- For each morphism generator  $f: X \rightarrow Y$  and  $s \in S$  such that  $M^\sharp f(s) = (x, y)$ , there is a morphism generator  $(f, s): (X, x) \rightarrow (Y, y)$ .

We want to prove that  $\hat{M}$  and  $\int M^\sharp$  are isomorphic.

On objects, the proof of bijectivity is as in the previous case. We can then define a symmetric monoidal functor  $\hat{M} \rightarrow \int M^\sharp$  that is bijective on objects,

and sends a generating morphism  $(f, s): (X, x) \rightarrow (Y, y)$  of  $\hat{M}$  to the morphism  $(f, s): (X, x) \rightarrow (Y, y)$  in  $\int M^\sharp$ .

We want to prove that this functor is full and faithful. Faithfulness is again straightforward. Suppose given  $(f_1, s_1), (f_2, s_2): (X, x) \rightarrow (Y, y)$  in  $\hat{M}$ . By construction,  $(f_1, s_1) = (f_2, s_2)$  in  $\int M^\sharp$  if and only if  $f_1 = f_2$  in  $\mathfrak{F}(M)$  and  $s_1 = s_2$ . But this means that  $(f_1, s_1) = (f_2, s_2)$  also in  $\hat{M}$ .

For fullness, take a morphism  $(f, s): (X, x) \rightarrow (Z, z)$  in  $\int M^\sharp$ , and notice the following:

- Each morphism  $(f, s)$  such that  $f$  is a generator, an identity or a symmetry in  $\mathfrak{F}(M)$  is also in  $\hat{M}$ ;
- If  $(f, s): (X, x) \rightarrow (Z, z)$  is such that  $f = g \circledast h$ , where  $g: X \rightarrow Y$  and  $h: Y \rightarrow Z$ , then by definition of composition in **Span**, we have  $s = (t, u)$  for some  $t, u$  with  $M^\sharp g(t) = (x, y)$  and  $M^\sharp h(u) = (y, z)$ . This means that  $(g, t): (X, x) \rightarrow (Y, y)$  and  $(h, u): (Y, y) \rightarrow (Z, z)$  are morphisms in  $\int M^\sharp$ ;
- If  $(f, s): (X_1 \otimes X_2, (x_1, x_2)) \rightarrow (Y_1 \otimes Y_2, (y_1, y_2))$  is such that  $f = f_1 \otimes f_2$ , where  $f_1: X_1 \rightarrow Y_1$  and  $f_2: X_2 \rightarrow Y_2$ , then  $s = (s_1, s_2)$  for some  $s_1, s_2$ , and we have:

$$((x_1, x_2), (y_1, y_2)) = M^\sharp f(s) = M^\sharp (f_1 \otimes f_2)(s_1, s_2) = (M^\sharp f_1 \times M^\sharp f_2)(s_1, s_2)$$

This means that  $M^\sharp f_1(s_1) = (x_1, y_1)$  and  $M^\sharp f_2(s_2) = (x_2, y_2)$ , and hence that  $(f_1, s_1): (X_1, x_1) \rightarrow (Y_1, y_1)$  and  $(f_2, s_2): (X_2, x_2) \rightarrow (Y_2, y_2)$  are morphisms in  $\int M^\sharp$ .

By definition, since  $\mathfrak{F}(M)$  is free, any morphism  $f$  can be decomposed into a composition of monoidal products of morphism generators, symmetries and identities. The points above prove that each of such morphisms is also in  $\int M^\sharp$ , and hence in  $\hat{M}$ . So  $f$  in  $\int M^\sharp$  is the image of  $f$  in  $\hat{M}$ , and the functor is full.

Since our correspondence is bijective on objects and fully faithful, this proves that  $\hat{M}$  and  $\int M^\sharp$  are isomorphic as categories. Since  $\hat{M}$  is free so is  $\int M^\sharp$ , completing the proof.  $\square$

**Theorem 2.** *Let  $(N, N^\sharp)$  be a guarded Petri net (with side effects).  $(Y, y)$  is reachable from  $(X, x)$  if and only if  $(Y, y)$  is reachable from  $(X, x)$  in the net  $\mathfrak{U}(\int N^\sharp)$ .*

*Proof.* By definition  $(Y, y)$  is reachable from  $(X, x)$  if and only if there is a morphism  $f: X \rightarrow Y$  in  $\mathfrak{F}(N)$  such that  $N^\sharp f(x) = y$  (resp.  $N^\sharp f(s) = (x, y)$  for some  $s \in S_f$ ). Again by definition, this means that  $f: (X, x) \rightarrow (Y, y)$  (resp.  $f_s: (X, x) \rightarrow (Y, y)$ ) is a morphism of  $\int N^\sharp$ . Since  $\int N^\sharp$  is free,  $f$  (resp.  $f_s$ ) can be decomposed as a composition of monoidal products of generating morphisms. But every generating morphism of  $\int N^\sharp$  corresponds to a transition of  $\mathfrak{U}(\int N^\sharp)$ , from which the thesis follows.  $\square$

**Lemma 5.** *If  $F$  is transition-preserving, so are  $\vec{F}$  and  $\widehat{F}$ .*

*Proof.* The proof is obvious considering that, by construction,  $f: (X, x) \rightarrow (Y, y)$  is a generator (resp. a symmetry) in  $\int N^\sharp$  if and only if it is a generator (resp. a symmetry) in  $\mathfrak{F}(N)$ .

An analogous argument holds for  $\widehat{F}$ .  $\square$

**Theorem 3.** *Denote with  $\mathbf{1}$  the terminal category, together with the trivial symmetric monoidal structure on it. There is a faithful, strong monoidal functor  $\mathbf{emb}_{\mathbf{Set}_*} : \mathbf{Petri}^{\mathbf{Set}_*} \rightarrow \mathbf{Petri}^{\mathbf{1}}$  defined as follows:*

- On objects, it sends  $(M, M^\sharp)$  to  $(\mathfrak{U}(\int M^\sharp), \mathfrak{U}(\int M^\sharp)^\sharp)$ .
- On morphisms, we send the functor  $F: (M, M^\sharp) \rightarrow (N, N^\sharp)$  to the functor<sup>4</sup>

$$\vec{F}: \left( \mathfrak{U}(\int M^\sharp), \mathfrak{U}(\int M^\sharp)^\sharp \right) \rightarrow \left( \mathfrak{U}(\int N^\sharp), \mathfrak{U}(\int N^\sharp)^\sharp \right)$$

Similarly, there is a faithful, strong monoidal functor  $\mathbf{emb}_{\mathbf{Span}} : \mathbf{Petri}^{\mathbf{Span}} \rightarrow \mathbf{Petri}^{\mathbf{1}}$  defined as follows:

- On objects, it sends  $(M, M^\sharp)$  to  $(\mathfrak{U}(\int M^\sharp), \mathfrak{U}(\int M^\sharp)^\sharp)$ .
- On morphisms, we send the functor  $F: (M, M^\sharp) \rightarrow (N, N^\sharp)$  to the functor<sup>5</sup>

$$\widehat{F}: \left( \mathfrak{U}(\int M^\sharp), \mathfrak{U}(\int M^\sharp)^\sharp \right) \rightarrow \left( \mathfrak{U}(\int N^\sharp), \mathfrak{U}(\int N^\sharp)^\sharp \right).$$

*Proof.* The proofs for  $\mathbf{Set}_*$  and  $\mathbf{Span}$  are very similar, so we just provide the one for  $\mathbf{Set}_*$ . Clearly if  $F$  is the identity functor  $(M, M^\sharp) \rightarrow (M, M^\sharp)$  then so is  $\vec{F}$ . For composition, consider  $F: (M, M^\sharp) \rightarrow (M, M^\sharp) N^\sharp$  and  $G: (N, N^\sharp) \rightarrow (P, P^\sharp)$ . We have to prove that  $F \vec{\circ} G = \vec{F} \circ \vec{G}$ . On objects,  $F \vec{\circ} G$  sends  $(X, x)$  in  $\int M^\sharp$  to  $((F \circ G)X, x)$  in  $\int P^\sharp$ , so it coincides with  $\vec{F} \circ \vec{G}$ . Now consider a morphism  $f: (X, x) \rightarrow (Y, y)$  in  $\int M^\sharp$ . This is sent by  $\vec{F}$  to  $Ff: (FX, x) \rightarrow (FY, y)$ , and applying  $\vec{G}$  to it one gets  $G(Ff): (G(FX), x) \rightarrow (G(FY), y)$ . Since  $G(F(-))$  is  $(F \circ G)(-)$ , we are done. This proves that  $\mathbf{emb}_{\mathbf{Set}_*}$  is a functor. Faithfulness is trivial.

Now we focus on monoidality. First of all we have to prove that

$$\mathbf{emb}_{\mathbf{Set}_*} \left( (M, M^\sharp) \otimes (N, N^\sharp) \right) \simeq \mathbf{emb}_{\mathbf{Set}_*} (M, M^\sharp) \otimes \mathbf{emb}_{\mathbf{Set}_*} (N, N^\sharp)$$

Remembering from Definition 1 that for each choice of semantics  $\mathcal{S}$  the monoidal structure on  $\mathbf{Petri}^{\mathcal{S}}$  is defined in terms of coproduct of symmetric monoidal

<sup>4</sup> To be absolutely precise, we are referring to the functor  $\mathfrak{F}(\mathfrak{U}(\int M^\sharp)) \simeq \int M^\sharp \xrightarrow{\vec{F}} \int N^\sharp \simeq \mathfrak{F}(\mathfrak{U}(\int N^\sharp))$ .

<sup>5</sup> To be absolutely precise, we are referring to the functor  $\mathfrak{F}(\mathfrak{U}(\int M^\sharp)) \simeq \int M^\sharp \xrightarrow{\widehat{F}} \int N^\sharp \simeq \mathfrak{F}(\mathfrak{U}(\int N^\sharp))$ .

categories, and hence from the coproduct of the underlying nets, this means that:

$$\begin{aligned}
(\mathfrak{U}(f[M^\sharp, N^\sharp]), \mathfrak{U}(f[M^\sharp, N^\sharp])^\sharp) &= \\
&= \mathbf{emb}_{\mathbf{Set}_*}(M + N, [M^\sharp, N^\sharp]) \\
&= \mathbf{emb}_{\mathbf{Set}_*}((M, M^\sharp) \otimes (N, N^\sharp)) \\
&\simeq \mathbf{emb}_{\mathbf{Set}_*}(M, M^\sharp) \otimes \mathbf{emb}_{\mathbf{Set}_*}(N, N^\sharp) \\
&= (\mathfrak{U}(f M^\sharp), \mathfrak{U}(f M^\sharp)^\sharp) \otimes (\mathfrak{U}(f N^\sharp), \mathfrak{U}(f N^\sharp)^\sharp) \\
&= (\mathfrak{U}(f M^\sharp) + \mathfrak{U}(f N^\sharp), [\mathfrak{U}(f M^\sharp)^\sharp, \mathfrak{U}(f N^\sharp)^\sharp])
\end{aligned}$$

Since  $\mathfrak{U}(-)$  preserves isomorphisms and coproducts, it is sufficient to prove:

$$f[M^\sharp, N^\sharp] \simeq f M^\sharp + f N^\sharp$$

- By definition, objects of  $f[M^\sharp, N^\sharp]$  are pairs  $(X, x)$  with  $X \in \mathfrak{F}(M + N) \simeq \mathfrak{F}(M) + \mathfrak{F}(N)$  and  $x \in [M^\sharp, N^\sharp]X$ . This is clearly isomorphic to  $\text{Obj } f M^\sharp \sqcup \text{Obj } f N^\sharp$ .
- Again by definition, we have

$$\begin{aligned}
\text{Hom } f[M^\sharp, N^\sharp] [(X, x), (Y, y)] &:= \\
&:= \{f \in \text{Hom}_{\mathfrak{F}(M) + \mathfrak{F}(N)} [X, Y] \mid [M^\sharp, N^\sharp]f(x) = y\}
\end{aligned}$$

This follows noting that by definition the set of morphisms of  $\mathfrak{F}(M) + \mathfrak{F}(N)$  is the disjoint union of the sets of morphisms of  $\mathfrak{F}(M)$  and  $\mathfrak{F}(N)$ .

Then we have to prove that  $\mathbf{emb}_{\mathbf{Set}_*}(F \otimes G) = \mathbf{emb}_{\mathbf{Set}_*}F \otimes \mathbf{emb}_{\mathbf{Set}_*}G$ . Unrolling definitions this amounts to prove that  $F \overset{\sim}{+} G = \overset{\sim}{F} \overset{\sim}{+} \overset{\sim}{G}$ , which is obvious.

Finally, we need to prove that  $\mathbf{emb}_{\mathbf{Set}_*}$  preserves the monoidal unit. Notice that for each choice of semantics  $\mathcal{S}$  the monoidal unit in  $\mathbf{Petri}^{\mathcal{S}}$  is taken to be  $(\emptyset, \emptyset^\sharp \mathcal{S})$ .  $\mathfrak{F}(\emptyset)$  is the free category consisting of only the monoidal unit  $I$ , and  $\emptyset^\sharp \mathcal{S} : \mathfrak{F}(\emptyset) \rightarrow \mathcal{S}$  sends the monoidal unit to the monoidal unit and its identity to itself. In our case, the monoidal unit of  $\mathbf{Petri}^{\mathbf{Set}_*}$  is  $(\emptyset, \emptyset^\sharp)$ , with  $\emptyset^\sharp$  sending the monoidal unit of  $\mathfrak{F}(\emptyset)$  to the singleton set  $\{*\}$  in  $\mathbf{Set}_*$ . In particular, this means that  $f \emptyset^\sharp \simeq \mathfrak{F}(\emptyset)$ , proving that

$$\mathbf{emb}_{\mathbf{Set}_*}(\emptyset, \emptyset^\sharp) = (\mathfrak{U}(f \emptyset^\sharp), \mathfrak{U}(f \emptyset^\sharp)^\sharp) \simeq (\mathfrak{U}(\mathfrak{F}(\emptyset)), \mathfrak{U}(\mathfrak{F}(\emptyset))^\sharp) \simeq (\emptyset, \emptyset^\sharp)$$

□

**Proposition 1.** *Denote with*

$$\mathit{for}_{\mathbf{Set}_*} : \mathbf{Petri}^{\mathbf{Set}_*} \rightarrow \mathbf{Petri}^1 \quad \mathit{for}_{\text{Span}} : \mathbf{Petri}^{\text{Span}} \rightarrow \mathbf{Petri}^1$$

the functors defined by sending each Petri net  $(M, M^\sharp)$  to  $(M, M^\sharp)$ . Then there are natural transformations:

$$\begin{array}{ccc}
 \text{Petri}^{\text{Set}_*} & \xrightarrow{\text{emb}_{\text{Set}_*}} & \text{Petri}^1 \\
 \Downarrow \pi & & \Downarrow \pi \\
 \text{Petri}^{\text{Span}} & \xrightarrow{\text{emb}_{\text{Span}}} & \text{Petri}^1 \\
 \Downarrow \pi & & \Downarrow \pi \\
 \text{Petri}^{\text{Set}_*} & \xrightarrow{\text{for}_{\text{Set}_*}} & \text{Petri}^1 \\
 \Downarrow \pi & & \Downarrow \pi \\
 \text{Petri}^{\text{Span}} & \xrightarrow{\text{for}_{\text{Span}}} & \text{Petri}^1
 \end{array}$$

*Proof.* For each object  $(M, M^\sharp)$  in  $\text{Petri}^{\text{Set}_*}$ , we set

$$\pi_{(M, M^\sharp)} : (\mathfrak{U}(f M^\sharp), \mathfrak{U}(f M^\sharp)^\sharp) \rightarrow (M, M^\sharp)$$

to be the functor  $\mathfrak{F}(\mathfrak{U}(f M^\sharp)) \simeq f M^\sharp \xrightarrow{\pi_M} \mathfrak{F}(M)$ , where  $\pi_M$  is defined as in Lemma 3. The naturality condition follows from Lemma 3 as well. The span case is analogous.  $\square$

## References

1. Genovese, F.: The essence of petri net gluings. <http://arxiv.org/abs/1909.03518>
2. Genovese, F., Herold, J.: Executions in (semi-)integer petri nets are compact closed categories, vol. 287, pp. 127–144. <https://doi.org/10.4204/EPTCS.287.7>
3. Jensen, K., Kristensen, L.M.: Coloured Petri Nets. Springer, Berlin and Heidelberg (2009). <https://doi.org/10.1007/b95112>. <http://link.springer.com/10.1007/b95112>
4. MacLane, S., Moerdijk, I.: Sheaves in Geometry and Logic, a First Introduction to Topos Theory, Universitext, vol. 13. Springer, Heidelberg (1994). <https://doi.org/10.1007/978-1-4612-0927-0>
5. Master, J.: Generalized petri nets. <http://arxiv.org/abs/1904.09091>
6. Meseguer, J., Montanari, U.: Petri nets are monoids. **88**(2), 105–155. [https://doi.org/10.1016/0890-5401\(90\)90013-8](https://doi.org/10.1016/0890-5401(90)90013-8)
7. Petri, C., Reisig, W.: Petri net. [http://www.scholarpedia.org/article/Petri\\_net](http://www.scholarpedia.org/article/Petri_net)
8. Sassone, V.: On the category of Petri net computations. In: Mosses, P.D., Nielsen, M., Schwartzbach, M.I. (eds.) CAAP 1995. LNCS, vol. 915, pp. 334–348. Springer, Heidelberg (1995). [https://doi.org/10.1007/3-540-59293-8\\_205](https://doi.org/10.1007/3-540-59293-8_205)
9. Statebox Team: The mathematical specification of the statebox language. <http://arxiv.org/abs/1906.07629>
10. Statebox Team: Statebox github page. <https://github.com/statebox/>