The Concurrent Games Abstract Machine

Multi-Token Geometry of Interaction and its Causal Unfolding

SIMON CASTELLAN, Inria,Univ. de Rennes, CNRS, IRISA, France
PIERRE CLAIRAMBAULT, Univ Lyon, EnsL, UCBL, CNRS, LIP, LYON, France

We introduce the *concurrent games abstract machine*: a multi-token machine for *Idealized Parallel Algol (*IPA*)*, a higher-order concurrent programming language with shared state and semaphores. Our abstract machine takes the shape of a compositional interpretation of terms as *Petri structures*, certain coloured Petri nets. For the purely functional fragment, our machine is conceptually close to *Geometry of Interaction* token machines, originating from Linear Logic and presenting higher-order computation as the low-level process of a token walking through a graph (a proof net) representing the term. We pair here these ideas with folklore ideas on the representation of first-order imperative concurrent programs as coloured Petri nets.

To prove our machine correct, we follow game semantics and represent types as certain games specifying dependencies and conflict between computational events. We define *Petri strategies* as those Petri structures obeying the rules of the game. In turn, we show how Petri strategies *unfold* to concurrent strategies in the sense of concurrent games on event structures. This not only entails correctness and adequacy of our machine, but also lets us generate operationally a causal description of the behaviour of programs at higher-order types.

Additional Key Words and Phrases: Geometry of Interaction, Game Semantics, Shared Memory Concurrency, Coloured Petri Nets, Higher-Order Computation

1 INTRODUCTION

Interactive semantics, and in particular *Game Semantics* [Abramsky et al. 2000; Hyland and Ong 2000] or *Geometry of Interaction (GoI)* [Danos et al. 1996; Girard 1989], aim at describing formally the execution of programs as an interactive process. They are particularly relevant in the presence of programming features that impact the geometry of the control flow (*e.g.* concurrency, higher-order, exceptions, *etc*), which pose a significant obstacle to modular reasoning. For example, Geometry of Interaction has long been proposed as a basis for the compilation of functional programs [Ghica 2007; Mackie 1995], while structures from game semantics are behind recent work on compositional certified compilation [Koenig and Shao 2020; Stewart et al. 2015; Xia et al. 2020].

GoI can be presented as an abstract machine: these *GoI token machines* seem to originate in [Danos and Regnier 1996] (the *Interaction Abstract Machine (IAM)*, for the λ-calculus) and [Mackie 1995] (for PCF). Token machines usually present a program as a graph (a *proof net*) and its execution as the walk of a *token* (standing for the control flow) through the graph, as determined by local rules. Token machines were extended in multiple ways: to *multi-token* machines, for Linear Logic [Laurent 2001] (without synchronization) or for functional programs [Dal Lago et al. 2014, 2015] (including in call-by-value) or interaction nets [Dal Lago et al. 2014]. They were redeveloped in a coalgebraic setting supporting a range of algebraic effects (*e.g.* nondeterminism, probability, exception, global states, interactive I/O, etc.) [Hoshino et al. 2014; Muroya et al. 2016]; and for quantitative effects up to quantum primitives [Dal Lago et al. 2017; Hasuo and Hoshino 2017]. Despite this remarkable breadth, many combinations of effects ubiquitous in realistic programming languages are missing, including *shared memory concurrency* (a notable exception is Ghica's *geometry of synthesis* [Ghica 2007], however for a language with a type discipline ensuring that state causes no race). This is

Authors' addresses: Simon Castellan, Inria,Univ. de Rennes, CNRS, IRISA, Rennes, France, simon.castellan@inria.fr; Pierre Clairambault, Univ Lyon, EnsL, UCBL, CNRS, LIP, LYON, F-69364 LYON Cedex 07, France, Pierre.Clairambault@ens-lyon.fr.

2018. 2475-1421/2018/1-ART1 $15.00
https://doi.org/

not due to an inherent restriction of GoI; but the usual correctness arguments (typically via cut elimination or realizability) may not extend well. This leaves open:

How to build a GoI token machine for a concurrent higher-order language with shared memory? (1)

In contrast, *game semantics*, after the initial models of PCF [Abramsky et al. 2000; Hyland and Ong 2000], offered a *wealth* of fully abstract models for a wide range of programming features among which local state [Abramsky and McCusker 1996], control operators [Laird 1997], exceptions [Laird 2001], higher-order state [Abramsky et al. 1998] and many others, including in particular higher-order shared memory concurrency [Ghica and Murawski 2008] – up to realistic languages [Murawski and Tzevelekos 2021]. It seems fair to say that game semantics has historically been more versatile than GoI. On the other hand, game semantics lack the direct operational flavour of GoI. Though game semantics is effective and can be computed and manipulated from the source code, the connection between a program and its semantics is quite remote and obfuscated by the many-layered definition of the interpretation of a program into a denotational model; this is an obstacle to it being used as formal basis in software tools, for validating optimizations or in certified compilation. This led researchers to investigate more operational, direct descriptions of the game semantics [Ghica and Tzevelekos 2012; Jaber 2015; Levy and Staton 2014] (though really, connections between game and operational semantics date back to [Danos et al. 1996]), in essence generating strategies by purely operational means. But such connections lack in generality; in particular they only exist for a few sequential languages. So a second motivating open question is:

How best to link game semantics and operational semantics? (2)

In particular, can we construct such an operational-denotational connection for a language with higher-order shared memory concurrency, say *Idealized Parallel Algol (*IPA*)*? Can we relate not merely to the interleaving model of [Ghica and Murawski 2008] but to the interpretation in the truly concurrent framework of *concurrent games on event structures* [Castellan and Clairambault 2020; Castellan et al. 2017, 2019], yielding an operational handle on causal reasoning?

Contributions. In this paper, we attack birds (1) and (2) with one stone. We give a multi-token abstract machine, called the *concurrent games abstract machine*, for IPA. More precisely, we give an interpretation of IPA transforming a program M into a *coloured Petri net* regarded as the concurrent games abstract machine loaded with M. For PCF, the obtained Petri net is similar to the usual token machine [Mackie 1995]; but our machine goes way beyond PCF.

Our correctness proof departs from usual GoI methods. We show that the multi-token machine for M *unfolds* to the event structure serving as interpretation of M in the truly concurrent games interpretation of [Castellan and Clairambault 2020; Castellan et al. 2019]; in the sense of the well-known unfolding of Petri nets to event structures [Hayman and Winskel 2008b; Nielsen et al. 1981]. This is proved compositionally, by defining an unfolding that preserves all operations used in the interpretation. Besides answering (2) for IPA, this also provides an answer to (1) as the correctness of our token machine then follows from adequacy of the concurrent games model of IPA.

In fact, we regard this as bringing a contribution to a third question:

How can we best state correctness of GoI on higher-order types? (3)

Indeed, for many token machines, correctness is stated for programs of ground type only, which is not ideal if GoI is to be thought of as a framework for compositional reasoning on programs. Another approach is to link GoI with game semantics: Baillot proved [Baillot 1999] that GoI *generates* the corresponding AJM-style strategy for IMELL. Ghica and Smith also prove the correctness of *Geometry of Synthesis* by linking it to (interleaving) game semantics [Ghica and Smith 2010]. Our result generalizes both, showing that our abstract machine generates the same causal structure as

$$\frac{}{\Gamma \vdash \mathsf{skip} : \mathbb{U}} \qquad \frac{}{\Gamma \vdash \mathsf{tt} : \mathbb{B}} \qquad \frac{}{\Gamma \vdash \mathsf{ff} : \mathbb{B}} \qquad \frac{}{\Gamma \vdash n : \mathbb{N}} \qquad \frac{}{\Gamma, x : A, \Delta \vdash x : A} \qquad \frac{\Gamma, x : A, \Delta \vdash M : O}{\Gamma, \Delta \vdash \lambda x^A. M : A \to O}$$

$$\frac{\Gamma \vdash M : A \to O \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : O} \qquad \frac{\Gamma \vdash M : O \to O}{\Gamma \vdash \mathsf{Y}\,M : O} \qquad \frac{\Gamma \vdash M : \mathbb{B} \qquad \Gamma \vdash N_1 : \mathbb{X} \qquad \Gamma \vdash N_2 : \mathbb{X}}{\Gamma \vdash \mathsf{if}\,M\,N_1\,N_2 : \mathbb{X}}$$

$$\frac{\Gamma \vdash M : \mathbb{X} \qquad \Gamma \vdash N : \mathbb{Y}}{\Gamma \vdash \mathsf{f}(M, N) : \mathbb{Z}}\ (\mathsf{f} : \mathbb{X} \times \mathbb{Y} \rightharpoonup \mathbb{Z}) \qquad \frac{\Gamma, x : \mathbb{X}, \Delta \vdash M : \mathbb{Y} \qquad \Gamma, \Delta \vdash N : \mathbb{X}}{\Gamma, \Delta \vdash \mathsf{let}\ x = N\ \mathsf{in}\ M : \mathbb{Y}} \qquad \frac{\Gamma, r : \mathbb{V}, \Delta \vdash M : \mathbb{X}}{\Gamma, \Delta \vdash \mathsf{newref}\ r\ \mathsf{in}\ M : \mathbb{X}}$$

$$\frac{\Gamma \vdash M : \mathbb{V} \qquad \Gamma \vdash N : \mathbb{N}}{\Gamma \vdash M{:=}N : \mathbb{U}} \qquad \frac{\Gamma \vdash M : \mathbb{V}}{\Gamma \vdash !M : \mathbb{N}} \qquad \frac{\Gamma, s : \mathbb{S}, \Delta \vdash M : \mathbb{X}}{\Gamma, \Delta \vdash \mathsf{newsem}\ s\ \mathsf{in}\ M : \mathbb{X}} \qquad \frac{\Gamma \vdash M : \mathbb{S}}{\Gamma \vdash \mathsf{grab}\ M : \mathbb{U}} \qquad \frac{\Gamma \vdash N : \mathbb{S}}{\Gamma \vdash \mathsf{release}\ N : \mathbb{U}}$$

Fig. 1. Typing rules for IPA

described by concurrent strategies (as related work, also note a recent GoI-like evaluation machine generating simple strategies for linear recursion schemes [Clairambault and Murawski 2019]).

As a final contribution, our abstract machine is fully implemented and available here.

Other related work. Though Petri nets are widely regarded as a model for concurrent programs, the literature is sparse on denotational interpretations of programs as Petri nets – a significant exception is [Hayman and Winskel 2008a], which interprets a simple (first-order) concurrent language. At the heart of such an interpretation lies a *composition* operation for Petri nets. In this direction note also the *open Petri nets* of [Baez and Master 2020], and the compositional unfolding of coloured Petri nets of [Chatain and Fabre 2010]. In these works, Petri nets synchronize on locations, whereas our nets synchronize on transitions instead.

Outline. In Section 2, we introduce our variant of IPA and set up the structure we use for its interpretation. In Section 3, we set up the concurrent games abstract machine as an interpretation of terms of IPA as *Petri structures*, certain coloured Petri nets. In Section 4, we briefly recall the target of the unfolding, (a symmetry-free version of) the concurrent game semantics of IPA in [Castellan and Clairambault 2020]. In Section 5, we define *Petri strategies* as those Petri structures that respect the game, and provide the *unfolding*. In Section 6, we briefly describe the implementation accompanying the paper and comment on a few optimizations. Finally, in Section 7 we conclude.

2 IPA AND ITS COMPOSITIONAL INTERPRETATIONS

2.1 The language IPA

IPA is a higher-order call-by-name concurrent language with shared memory and semaphores, serving as paradigmatic language for these features in the game semantics literature [Ghica and Murawski 2008]. Our variant is more expressive in some ways (in particular, it has a let construct); but it also has the restriction that variable and semaphore types should not appear at the right hand side of an arrow. This simplifies the exposition without significantly harming expressiveness.

2.1.1 Types and terms. We first describe the **types** of IPA, generated by

$$A, B, C ::= O \mid \mathbb{V} \mid \mathbb{S} \qquad\qquad O ::= \mathbb{U} \mid \mathbb{B} \mid \mathbb{N} \mid A \to O$$

where types generated by O are called **well-opened**. We have $\mathbb{U}$ a *unit* type, $\mathbb{B}$ and $\mathbb{N}$ respectively types for *booleans* and *natural numbers*, a type $\mathbb{V}$ for *integer references*, and $\mathbb{S}$ for *semaphores*. The split into *standard types* and *well-opened types* implements that $\mathbb{V}$ and $\mathbb{S}$ should not appear on the right of an arrow. We refer to $\mathbb{U}, \mathbb{B}$ and $\mathbb{N}$ as **ground types**, and use $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ to range over those.

We define terms directly via typing rules – throughout this paper, we only consider well-typed terms. **Contexts** are lists of typed variables $x_1 : A_1, \dots, x_n : A_n$, where variables come from a fixed countable set Var. **Typing judgments** have the form $\Gamma \vdash M : A$, with Γ a context and A a type.

The construction $\mathsf{f}(M, N)$ applies to any computable partial function $\mathsf{f} : \mathbb{X} \times \mathbb{Y} \rightharpoonup \mathbb{Z}$ (abusing notations to treat the ground types $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ as the underlying sets), and its two operands are intended to be evaluated in parallel. This covers many usual primitives: for instance, if $\Gamma \vdash M, N : \mathbb{U}$, we define $M \parallel N = \parallel(M, N)$, with $\parallel : \mathbb{U} \times \mathbb{U} \to \mathbb{U}$ the trivial function – in fact, we shall use implicitly $\parallel : \mathbb{U} \times \mathbb{X} \to \mathbb{X}$, so that parallel composition propagates a value. The usual predecessor, successor, zero test primitives of PCF can be obtained similarly as particular cases of this operation. Conditionals eliminate only to ground type, but as usual in call-by-name, a more general conditional can be obtained as syntactic sugar. We refer to constants of ground type as **values**; we use v to range over values of any type, and n, b or c to range over values of respective types $\mathbb{N}, \mathbb{B}$ or $\mathbb{U}$.

2.1.2 Examples. We include a few examples of simple IPA programs. First:

$$\vdash \mathsf{coin} = \mathsf{newref}\ r\ \mathsf{in}\ (r := 1 \parallel \mathsf{iszero}\ !r) : \mathbb{B}$$

is a non-deterministic boolean – by convention newref automatically initializes r to 0, so that coin directly sets up a race. Though we have no primitive for sequential composition, for $\Gamma \vdash M : \mathbb{X}$ and $\Gamma \vdash N : \mathbb{Y}$, we define $M;\ N$ as let $x = M$ in $N : \mathbb{Y}$. Another interesting example is

$$x : \mathbb{U}, y : \mathbb{X} \vdash \mathsf{newsem}\ s\ \mathsf{in}\ \mathsf{grab}\ s;\ (x;\ \mathsf{release}\ s \parallel \mathsf{grab}\ s;\ y) : \mathbb{X}$$

which behaves like sequential composition, using a semaphore for synchronization.

As a final example, IPA allows dynamic creation of references and semaphores: for instance,

$$\vdash (\lambda F.\, F\, (\lambda g^{\mathbb{N}\to\mathbb{N}} n^{\mathbb{N}}.\, g\, 1);\ F\, (\lambda g^{\mathbb{N}\to\mathbb{N}} n^{\mathbb{N}}.\, n))\ (\lambda f^{(\mathbb{N}\to\mathbb{N})\to\mathbb{N}\to\mathbb{N}}.\, \mathsf{newref}\ r\ \mathsf{in}\ f\ (\lambda x.\, r := x;\ 0)\ !r) : \mathbb{N}$$

returns 0, because execution causes the initialization of two independent references. An unbounded number of references can arise in this way if this happens within recursion.

We hope these show that though IPA is a toy language, it is a semantically highly non-trivial one which poses realistic challenges. We chose IPA because of its historical importance in the game semantics community, and because there already is a detailed concurrent games model ready for this language [Castellan and Clairambault 2020]. But unlike in [Ghica and Murawski 2008], we made sure to include a let construct to show that the concurrent games abstract machine does handle an explicit control of the evaluation order more typical of call-by-value languages.

2.1.3 Operational semantics. The reference semantics for IPA is a small-step interleaving operational semantics following closely that of [Ghica and Murawski 2008].

We fix a countable set $\mathcal{L}$ of **memory locations**. A **store** is a partial map $s : \mathcal{L} \rightharpoonup \mathbb{N}$ with finite domain where $\mathbb{N}$ stands, overloading notations, for the set of natural numbers. **Configurations** of the operational semantics are tuples $\langle M, s\rangle$ where s is a store with $\mathrm{dom}(s) = \{\ell_1, \dots, \ell_n\}$ and $\Sigma \vdash M : A$ with $\Sigma = \ell_1 : \mathbb{V}, \dots, \ell_i : \mathbb{V}, \ell_{i+1} : \mathbb{S}, \dots, \ell_n : \mathbb{S}$. Reduction rules have the form

$$\langle M, s\rangle \rightsquigarrow \langle M', s'\rangle$$

where $\mathrm{dom}(s) = \mathrm{dom}(s')$; we write $\rightsquigarrow^*$ for the reflexive transitive closure. If $\vdash M : \mathbb{X}$, we write $M \Downarrow$ if $\langle M, \emptyset\rangle \rightsquigarrow^* \langle v, \emptyset\rangle$ for some value v. Then we say that M **converges**, else it **diverges**.

The detailed reduction rules are essentially as in [Ghica and Murawski 2008]. We postpone them to Appendix A, as they will be referred to only indirectly in this paper (via Theorem 4.12).

2.2 IPA-structures

We describe an abstract structure for the interpretation of IPA. The goal is not a categorical semantics with a general soundness theorem, but merely to structure the compositional interpretation. An IPA-structure is a category with just enough structure so that the interpretation may be defined following the standard lines of the semantics of functional languages into an adequately equipped model of Intuitionistic Linear Logic (say, a Seely category [Melliès 2009]). We call a **precategory** a structure with the data of a category, an associative composition but no identity laws.

Definition 2.1. An IPA-**structure** is a precategory $\mathcal{C}$, with a set $\mathcal{C}_\bullet$ of **well-opened** objects – we use A, B, C to range over the set $\mathcal{C}_0$ of objects and O to range over $\mathcal{C}_\bullet$ – and equipped with:

- **Constructions.** We have $\mathbf{U}, \mathbf{B}, \mathbf{N} \in \mathcal{C}_\bullet$, $\mathbf{V}, \mathbf{S} \in \mathcal{C}_0$, and constructions:

  tensor: for any $A, B \in \mathcal{C}_0$, there is $A \otimes B \in \mathcal{C}_0$,
  product: for any finite family $(A_x)_{x \in V}$ with $V \subseteq_f \mathsf{Var}$, there is $\&_{x \in V} A_x \in \mathcal{C}_0$,
  linear arrow: for any $A \in \mathcal{C}_0$ and $O \in \mathcal{C}_\bullet$, there is $A \multimap O \in \mathcal{C}_\bullet$,
  exponential: for any $A \in \mathcal{C}_0$, there is $!A \in \mathcal{C}_0$,

  where we write $\&\emptyset = \top$ for the product of the empty family.
- **Operations.** We have the following three constructions on morphisms:

$$\begin{array}{rccccc} \textit{tensor:} & \otimes & : & \mathcal{C}(A_1, B_1) \times \mathcal{C}(A_2, B_2) & \to & \mathcal{C}(A_1 \otimes A_2, B_1 \otimes B_2) \\ \textit{currying:} & \Lambda^{\Gamma,\Delta}_{x:A,O} & : & \mathcal{C}(!(\&[\Gamma, x : A, \Delta]), O) & \to & \mathcal{C}(!(\&[\Gamma, \Delta]), !A \multimap O) \\ \textit{promotion:} & (-)^\dagger & : & \mathcal{C}(!\Gamma, O) & \to & \mathcal{C}(!\Gamma, !O) \end{array}$$

  where, if $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ where $A_i \in \mathcal{C}_0$ for all i, we write $[\Gamma]$ for $(A_x)_{x \in \{x_1, \ldots, x_n\}}$;
- **Primitives.** We have the basic morphisms listed in Figure 3, where $\otimes^0 A = 1$, $\otimes^1 A = A$, $\otimes^{n+2} A = A \otimes (\otimes^{n+1} A)$, and writing $\mathbf{w}_A \in \mathcal{C}(!A, 1)$ for $\mathbf{c}^0_A$ and $\mathbf{c}_A \in \mathcal{C}(!A, !A \otimes !A)$ for $\mathbf{c}^2_A$.

We shall soon define an interpretation of IPA in any IPA-structure – again, requiring no equation. Besides structuring the interpretation, it also provides a clean way to relate interpretations:

Definition 2.2. Consider $\mathcal{C}$ and $\mathcal{D}$ two IPA-structures, and $F : \mathcal{C} \to \mathcal{D}$ a functor. Then, F is a **strict** IPA-**functor** iff it preserves all structure on the nose.

2.3 Interpretation of IPA

The exact definition of the interpretation follows the standard lines of the interpretation of call-by-name languages into models of intuitionistic linear logic. Fix $\mathcal{C}$ an IPA-structure.

We interpret the *types* of IPA as objects of $\mathcal{C}$, with $[\![\mathbb{U}]\!] = \mathbf{U}$, $[\![\mathbb{B}]\!] = \mathbf{B}$, $[\![\mathbb{N}]\!] = \mathbf{N}$, $[\![\mathbb{V}]\!] = \mathbf{V}$, $[\![\mathbb{S}]\!] = \mathbf{S}$, and finally, $[\![A \to B]\!] = ![\![A]\!] \multimap [\![B]\!]$. Note that well-opened types are mapped to well-opened objects. *Contexts* are also interpreted as objects of $\mathcal{C}$, with $[\![x_1 : A_1, \ldots, x_n : A_n]\!] = \&_{x_i \in \{x_1, \ldots, x_n\}} [\![A_i]\!]$. To any *typed term* $\Gamma \vdash M : A$ we associate as interpretation a morphism $[\![\Gamma \vdash M : A]\!] \in \mathcal{C}(![\![\Gamma]\!], [\![A]\!])$ sometimes shortened to $[\![M]\!]$, following the clauses of Figure 2. Finally, we have:

Lemma 2.3. *Consider $\mathcal{C}, \mathcal{D}$ two* IPA-*structures, and $F : \mathcal{C} \to \mathcal{D}$ an* IPA-*functor. Then,*

(1) *for all type A and context Γ, $F([\![A]\!]_\mathcal{C}) = [\![A]\!]_\mathcal{D}$ and $F([\![\Gamma]\!]_\mathcal{C}) = [\![\Gamma]\!]_\mathcal{D}$,*
(2) *for all term $\Gamma \vdash M : A$, we have $F([\![M]\!]_\mathcal{C}) = [\![M]\!]_\mathcal{D}$.*

This follows by induction on the definition of the interpretation, applying at each step the preservation property of F – so strict IPA-functors list the proof obligations to relate two interpretations.

3 PETRI STRUCTURES FOR IPA

We start by describing the first and central IPA-structure of this paper, that of *Petri structures*.

$$
\begin{array}{rcl}
[\![\Gamma \vdash \mathsf{skip} : \mathbb{U}]\!] & = & \mathbf{skip} \circ \mathbf{w}_\Gamma \\
[\![\Gamma \vdash \mathsf{tt} : \mathbb{B}]\!] & = & \mathbf{tt} \circ \mathbf{w}_\Gamma \\
[\![\Gamma \vdash \mathsf{ff} : \mathbb{B}]\!] & = & \mathbf{ff} \circ \mathbf{w}_\Gamma \\
[\![\Gamma, x : A, \Delta \vdash x : A]\!] & = & \mathbf{var}^{\Gamma,\Delta}_{x:A} \\
[\![\Gamma, \Delta \vdash \lambda x^A.\, M : A \to O]\!] & = & \Lambda^{\Gamma,\Delta}_{x:A,O}([\![M]\!]) \\
[\![\Gamma \vdash M\, N : O]\!] & = & \mathbf{ev}_{!A,O} \circ ([\![M]\!] \otimes [\![N]\!]^\dagger) \circ \mathbf{c}_\Gamma \\
[\![\Gamma \vdash \mathsf{Y}\, M : O]\!] & = & \mathrm{Y}_O \circ [\![M]\!]^\dagger \\
[\![\Gamma \vdash \mathsf{if}\, M\, N_1\, N_2 : \mathbb{X}]\!] & = & \mathbf{if}_\mathrm{X} \circ ([\![M]\!] \otimes ([\![N_1]\!] \otimes [\![N_2]\!])) \circ \mathbf{c}^3_\Gamma \\
[\![\Gamma \vdash \mathsf{f}(M, N) : \mathbb{Z}]\!] & = & \mathbf{op}(\mathrm{f})^{\mathrm{X,Y}}_\mathrm{Z} \circ ([\![M]\!] \otimes [\![N]\!]) \circ \mathbf{c}_\Gamma \\
[\![\Gamma, \Delta \vdash \mathsf{let}\; x = N\; \mathsf{in}\; M : \mathbb{Y}]\!] & = & \mathbf{let}_\mathrm{X,Y} \circ (\Lambda^{\Gamma,\Delta}_\mathrm{X,Y}([\![M]\!]) \otimes [\![N]\!]) \circ \mathbf{c}_{\Gamma,\Delta} \\
[\![\Gamma, \Delta \vdash \mathsf{newref}\; x\; \mathsf{in}\; M : \mathbb{X}]\!] & = & \mathbf{newref}_\mathrm{X} \circ \Lambda^{\Gamma,\Delta}_\mathrm{V,X}([\![M]\!]) \\
[\![\Gamma \vdash M{:=}N : \mathbb{U}]\!] & = & \mathbf{assign} \circ ([\![M]\!] \otimes [\![N]\!]) \circ \mathbf{c}_\Gamma \\
[\![\Gamma \vdash !M : \mathbb{N}]\!] & = & \mathbf{deref} \circ [\![M]\!] \\
[\![\Gamma, \Delta \vdash \mathsf{newsem}\; x\; \mathsf{in}\; M : \mathbb{X}]\!] & = & \mathbf{newsem}_\mathrm{X} \circ \Lambda^{\Gamma,\Delta}_\mathrm{V,X}([\![M]\!]) \\
[\![\Gamma \vdash \mathsf{grab}\, M : \mathbb{U}]\!] & = & \mathbf{grab} \circ [\![M]\!] \\
[\![\Gamma \vdash \mathsf{release}\, N : \mathbb{U}]\!] & = & \mathbf{release} \circ [\![N]\!]
\end{array}
$$

Fig. 2. Interpretation of IPA in an IPA-structure

$$
\begin{array}{rcl}
\mathbf{var}^{\Gamma,\Delta}_{x:A} & \in & \mathcal{C}(!(\&[\Gamma, x : A, \Delta]), A) \\
\mathbf{ev}_{A,O} & \in & \mathcal{C}((A \multimap O) \otimes A, O) \\
\mathbf{c}^n_\Gamma & \in & \mathcal{C}(!\Gamma, \otimes^n(!\Gamma)) \\
\mathbf{skip} & \in & \mathcal{C}(1, \mathbf{U}) \\
\mathbf{tt} & \in & \mathcal{C}(1, \mathbf{B}) \\
\mathbf{ff} & \in & \mathcal{C}(1, \mathbf{B}) \\
\mathbf{n} & \in & \mathcal{C}(1, \mathbf{N}) \\
\mathrm{Y}_O & \in & \mathcal{C}(!(!O \multimap O), O) \\
\mathbf{if}_\mathrm{X} & \in & \mathcal{C}(\mathbf{B} \otimes (\mathbf{X} \otimes \mathbf{X}), \mathbf{X}) \\
\mathbf{op}(\mathrm{f})^{\mathrm{X,Y}}_\mathrm{Z} & \in & \mathcal{C}(\mathbf{X} \otimes \mathbf{Y}, \mathbf{Z}) \\
\mathbf{let}_\mathrm{X,Y} & \in & \mathcal{C}((!\mathbf{X} \multimap \mathbf{Y}) \otimes \mathbf{X}, \mathbf{Y}) \\
\mathbf{newref}_\mathrm{X} & \in & \mathcal{C}(!\mathbf{V} \multimap \mathbf{X}, \mathbf{X}) \\
\mathbf{newsem}_\mathrm{X} & \in & \mathcal{C}(!\mathbf{S} \multimap \mathbf{X}, \mathbf{X}) \\
\mathbf{assign} & \in & \mathcal{C}(\mathbf{V} \otimes \mathbf{N}, \mathbf{U}) \\
\mathbf{deref} & \in & \mathcal{C}(\mathbf{V}, \mathbf{N}) \\
\mathbf{grab} & \in & \mathcal{C}(\mathbf{S}, \mathbf{U}) \\
\mathbf{release} & \in & \mathcal{C}(\mathbf{S}, \mathbf{U})
\end{array}
$$

Fig. 3. Primitives of IPA-structures

3.1 Definition and Examples

Consider fixed sets Tok and $\mathcal{M}$, respectively called *tokens* (which will serve as *colours*), and *addresses* (which will *label* certain transitions). Each $\mathrm{m} \in \mathcal{M}$ has a *polarity* $\mathrm{pol}(\mathrm{m}) \in \{-, +\}$, specifying whether it is a label for actions by the program $(+)$ or the environment $(-)$. We shall not give just yet the definition of these data, which will not be necessary until later on.

We use $\uplus$ to denote the usual set-theoretic union, when it is known to be disjoint.

Definition 3.1. Consider $M \subseteq \mathcal{M}$ a finite subset of addresses.
A **Petri structure on** M is $\sigma = \langle \mathcal{L}, \mathcal{T} = \mathcal{T}^+ \uplus \mathcal{T}^0 \uplus \mathcal{T}^-, \partial, \mathrm{pre}, \mathrm{post}, \delta \rangle$ where:

- $\mathcal{L}$ is a finite set of **locations**,
- $\mathcal{T}$ is a finite set of **transitions** sorted by *polarity* $+, 0$ or $-$,
- ∂ is a labelling function $\mathcal{T}^+ \uplus \mathcal{T}^- \to M$ such that $t \in \mathcal{T}^{\mathrm{pol}(\partial(t))}$ for all $t \in \mathcal{T}^+ \uplus \mathcal{T}^-$,
- pre is a function $\mathcal{T} \to \mathcal{P}(\mathcal{L})$ of **pre-conditions**,
- post is a function $\mathcal{T} \to \mathcal{P}(\mathcal{L})$ of **post-conditions**,

such that $\mathrm{pre}(t^-) = \emptyset$, $\mathrm{post}(t^+) = \emptyset$ for all transitions with the indicated polarity; and δ assigns to any $t \in \mathcal{T}$ a partial function, the **transition function**, typed according to its polarity:

$$
\begin{array}{rcrcl}
\delta\langle t^0 \rangle & : & \mathrm{cond}(\mathrm{pre}(t)) & \rightharpoonup & \mathrm{cond}(\mathrm{post}(t))\,, \\
\delta\langle t^- \rangle & : & \mathrm{Tok} & \rightharpoonup & \mathrm{cond}(\mathrm{post}(t))\,, \\
\delta\langle t^+ \rangle & : & \mathrm{cond}(\mathrm{pre}(t)) & \rightharpoonup & \mathrm{Tok}\,.
\end{array}
$$

where $\mathrm{cond}(L) = \mathrm{Tok}^L$ is the set of **conditions** with support L – we write cond for all conditions.

In the terminology of coloured Petri nets, the set of colours is Tok, and is independent of the location. *Locations* are internal buffers that may contain one or several tokens. *Neutral transitions* correspond to internal computation: firing t^0 takes one token from each location of $\mathrm{pre}(t)$, puts one token in each location of $\mathrm{post}(t)$, acting on colours as prescribed by $\delta\langle t \rangle$. *Negative* and *Positive* transitions, also called **visible transitions**, interact with the outside world. We shall see later on the rules that govern this interaction. For now we focus on **closed** Petri structures, defined as Petri structures on the set $M = \{\mathtt{Q}^-, \mathtt{A}^+\}$ – here $\mathtt{Q}^-$ *initiates* computation, while $\mathtt{A}^+$ *terminates* it.

For now we shall content ourselves with the informal description of the token game given above, and build up intuition by considering examples of closed Petri structures.

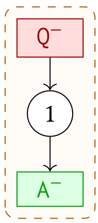
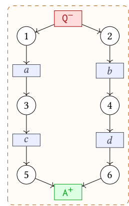
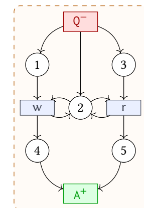


Fig. 4



$$\begin{aligned}
\delta\langle \mathsf{Q}\rangle\,(\bullet) &= \{\bullet^{@1}, \bullet^{@2}\}\\
\delta\langle a\rangle\,(\{\bullet^{@1}\}) &= \{1^{@3}\}\\
\delta\langle b\rangle\,(\{\bullet^{@2}\}) &= \{1^{@4}\}\\
\delta\langle c\rangle\,(\{1^{@3}\}) &= \{1^{@5}\}\\
\delta\langle d\rangle\,(\{1^{@4}\}) &= \{1^{@6}\}\\
\delta\langle \mathsf{A}\rangle\,(\{n^{@5}, m^{@6}\}) &= n+m
\end{aligned}$$

Fig. 5. Closed Petri structure for $1+1 : \mathbb{N}$



$$\begin{aligned}
\delta\langle \mathsf{Q}\rangle\,(\bullet) &= \{1^{@1}, 0^{@2}, \bullet^{@3}\}\\
\delta\langle \mathsf{w}\rangle(\{n^{@1}, p^{@2}\}) &= \{\checkmark^{@4}, n^{@2}\}\\
\delta\langle \mathsf{r}\rangle\,(\{n^{@2}, \bullet^{@3}\}) &= \{n^{@2}, n^{@5}\}\\
\delta\langle \mathsf{A}\rangle\,(\{\checkmark^{@4}, n^{@5}\}) &= \mathsf{iszero}(n)
\end{aligned}$$

Fig. 6. Closed Petri structure for coin : $\mathbb{B}$

3.1.1 *Closed Petri structures.* We temporarily fix Tok $= \{\bullet\} \uplus \mathbb{N}$ – $\bullet$ is a token with no value, while natural numbers stand for the corresponding value (we shall later settle on a more expressive Tok).

We draw a Petri structure σ following standard conventions from Petri nets: *locations* are circles, while *transitions* are boxes. The graph drawn carries the information of $\mathcal{L}$, $\mathcal{T}$, pre and post, while δ is given separately as a transition table. Whenever unambiguous, we use as name of visible transitions their label via ∂. If $L = \{l_1, \ldots, l_n\} \subseteq \mathcal{L}$, a condition $(\mathsf{t}_i)_{l_i \in L} \in \mathsf{cond}(L)$ is written $\{\mathsf{t}_1^{@l_1}, \ldots, \mathsf{t}_n^{@l_n}\}$ where each $l_i \in L$ appears exactly once. An individual $\mathsf{t}^{@l}$, where $\mathsf{t} \in$ Tok and $l \in \mathcal{L}$, is called a **token-in-location**, or **tokil** for short – we write TokIL(σ) for the set of tokils.

We start with the closed Petri structure for the constant 1, in Figure 4. Upon being triggered by Q^-, the net immediately prepares value 1 in location 1 by $\delta\langle \mathsf{Q}^-\rangle(\bullet) = \{1^{@1}\}$. This enables transition A^+, which outputs the value via $\delta\langle \mathsf{A}^+\rangle(\{n^{@1}\}) = n$. Figure 5 presents a parallel evaluation of $1+1$. When triggered the net throws two tokils $\bullet^{@1}$ and $\bullet^{@2}$ corresponding to evaluation requests for the two constants. Both tokils are forwarded to an independent copy of the structure of Figure 4. Upon receiving the two values in locations 5 and 6, the last transition fires and outputs the sum $1+1$. The example – or its closed variant as obtained by the interpretation – may be ran in the implementation here. It illustrates how Petri structures handle basic calling and returning mechanisms, and displays parallel computation. Those simple examples are not particularly original, and follow the usual folklore lines along which coloured Petri nets may be used to represent programs.

Another well-known idea is that Petri nets can represent shared state: Figure 6 shows the closed Petri structure for $\vdash$ coin : $\mathbb{B}$ as defined in Section 2.1.2. Upon initialization, the net throws three tokens: the tokil $0^{@2}$ initializes the variable to value 0; the tokil $1^{@1}$ is a write request for the value 1; and $\bullet^{@3}$ is a read request. There is a race between the read and write requests: if r wins, the value in location 5 ends up being 0, while if w then r reads value 1 instead. The final transition A^+ waits for the write acknowledgment and the result of the read to return the value read. The example – or its close variant as obtained by the interpretation – may be ran in the implementation here.

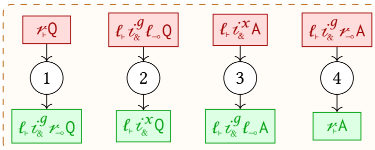
3.1.2 *Open Petri structures.* Petri structures also handle *open* or *higher-order* programs.

For open programs interacting with their execution environment, Petri structures will have a wider range of labels for visible transitions, reflecting the possible avenues of interaction. For instance, Petri structures corresponding to programs typed with $g : \mathbb{U} \to \mathbb{U}, x : \mathbb{U} \vdash \mathbb{U}$ will use

$$\mathcal{M} = \{\ell_{\vdash}\,\iota_{\&}^{g}\,\ell_{\multimap}\mathsf{Q}^-, \ell_{\vdash}\,\iota_{\&}^{g}\,\ell_{\multimap}\mathsf{A}^+, \ell_{\vdash}\,\iota_{\&}^{g}\,r_{\multimap}\mathsf{Q}^+, \ell_{\vdash}\,\iota_{\&}^{g}\,r_{\multimap}\mathsf{A}^-, \ell_{\vdash}\,\iota_{\&}^{x}\mathsf{Q}^+, \ell_{\vdash}\,\iota_{\&}^{x}\mathsf{A}^-, r_{\vdash}\mathsf{Q}^-, r_{\vdash}\mathsf{A}^+\}$$

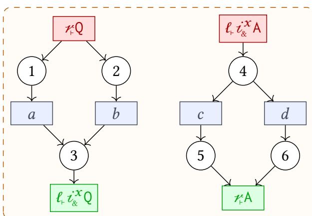
where the *injections* $\ell_{\vdash}$ and $r_{\vdash}$ indicate the two sides of $\vdash$; $\iota_{\&}^{x}$ and $\iota_{\&}^{g}$ point to a variable name; $\ell_{\multimap}$ and $r_{\multimap}$ point to either side of an arrow $\to$; and Q and A stand for "Question" (call) or "Answer" (return). Without Q and A, each of the addresses above corresponds to an occurrence of a base type in $g : \mathbb{U} \to \mathbb{U}, x : \mathbb{U} \vdash \mathbb{U}$, and then each base type occurrence admits a *call* and a *return*.

More generally, we shall use in this paper the following addresses:



$$\begin{array}{lllll}
\delta\langle r_{\vdash}\mathsf{Q}\rangle & (\bullet) & = & \{\bullet^{@1}\} \\
\delta\langle \ell_{\vdash}\iota^{g}_{\&}\ell_{\multimap}\mathsf{Q}\rangle & (\bullet) & = & \{\bullet^{@2}\} \\
\delta\langle \ell_{\vdash}\iota^{x}_{\&}\mathsf{A}\rangle & (\checkmark) & = & \{\checkmark^{@3}\} \\
\delta\langle \ell_{\vdash}\iota^{g}_{\&}r_{\multimap}\mathsf{A}\rangle & (\checkmark) & = & \{\checkmark^{@4}\}
\end{array}
\qquad
\begin{array}{llll}
\delta\langle \ell_{\vdash}\iota^{g}_{\&}r_{\multimap}\mathsf{Q}\rangle & (\{\bullet^{@1}\}) & = & \bullet \\
\delta\langle \ell_{\vdash}\iota^{x}_{\&}\mathsf{Q}\rangle & (\{\bullet^{@2}\}) & = & \bullet \\
\delta\langle \ell_{\vdash}\iota^{g}_{\&}\ell_{\multimap}\mathsf{A}\rangle & (\{\checkmark^{@3}\}) & = & \checkmark \\
\delta\langle r_{\vdash}\mathsf{A}\rangle & (\{\checkmark^{@4}\}) & = & \checkmark
\end{array}$$

Fig. 7. An open Petri structure for $g : \mathbb{U} \longrightarrow \mathbb{U}, x : \mathbb{U} \vdash g\,x : \mathbb{U}$



$$\begin{array}{llll}
\delta\langle r_{\vdash}\mathsf{Q}\rangle & (([\,], \bullet)) & = & \{([\blacklozenge], \bullet)^{@1}, ([\blacklozenge], \bullet)^{@2}\} \\
\delta\langle a\rangle & (\{([\mathrm{e}], \mathrm{d})^{@1}\}) & = & \{([\ell_{!}\mathrm{e}], \mathrm{d})^{@3}\} \\
\delta\langle b\rangle & (\{([\mathrm{e}], \mathrm{d})^{@2}\}) & = & \{([r_{!}\mathrm{e}], \mathrm{d})^{@3}\} \\
\delta\langle \ell_{\vdash}\iota^{x}_{\&}\mathsf{Q}\rangle & (\{(\mathrm{s}, \mathrm{d})^{@3}\}) & = & (\mathrm{s}, \mathrm{d}) \\
\delta\langle \ell_{\vdash}\iota^{x}_{\&}\mathsf{A}\rangle & (\mathrm{s}, \mathrm{d}) & = & \{(\mathrm{s}, \mathrm{d})^{@4}\} \\
\delta\langle c\rangle & (\{([\ell_{!}\blacklozenge], \mathrm{d})^{@4}\}) & = & \{([\,], \mathrm{d})^{@5}\} \\
\delta\langle d\rangle & (\{([r_{!}\blacklozenge], \mathrm{d})^{@4}\}) & = & \{([\,], \mathrm{d})^{@6}\} \\
\delta\langle r_{\vdash}\mathsf{A}\rangle & (\{([\,], \mathrm{d})^{@5}, ([\,], \mathrm{d}')^{@6}\}) & = & ([\,], \mathrm{d} + \mathrm{d}')
\end{array}$$

Fig. 8. Petri structure for $x : \mathbb{N} \vdash x + x : \mathbb{N}$

Definition 3.2. We first define the set $\mathcal{M}$ of **addresses** as

$$\mathcal{M} ::= \ell_{\otimes}\mathcal{M} \mid r_{\otimes}\mathcal{M} \mid \ell_{\vdash}\mathcal{M} \mid r_{\vdash}\mathcal{M} \mid \ell_{\multimap}\mathcal{M} \mid r_{\multimap}\mathcal{M} \mid \iota^{x}_{\&}\mathcal{M} \mid w_{\mathbb{V}}\mathcal{M} \mid r_{\mathbb{V}}\mathcal{M} \mid g_{\mathbb{S}}\mathcal{M} \mid r_{\mathbb{S}}\mathcal{M} \mid \mathsf{Q} \mid \mathsf{A},$$

for $x \in \mathrm{Var}$ – we use m to range over addresses. These have a **polarity** defined as $\mathrm{pol}(\mathsf{Q}) = -$, $\mathrm{pol}(\mathsf{A}) = +$, $\mathrm{pol}(\ell_{\vdash}\mathsf{m}) = -\mathrm{pol}(\mathsf{m})$, $\mathrm{pol}(\ell_{\multimap}\mathsf{m}) = -\mathrm{pol}(\mathsf{m})$, and preserved in all other cases.

In this way, we show in Figure 7 (a close variant of) the open Petri structure interpreting $g : \mathbb{U} \longrightarrow \mathbb{U}, x : \mathbb{U} \vdash g\,x : \mathbb{U}$. Tokens are either $\bullet$ for a data request, or $\checkmark$ for the unique possible value on $\mathbb{U}$. Upon initialization with $r_{\vdash}\mathsf{Q}^{-}$, the net interrogates the return value of g. If g calls its argument with $\ell_{\vdash}\iota^{g}_{\&}\ell_{\multimap}\,\mathsf{Q}^{-}$, the net interrogates the return value of x. If x returns a value with $\ell_{\vdash}\iota^{x}_{\&}\mathsf{A}$, this value is propagated to the argument of g. Finally, if g returns with $\ell_{\vdash}\iota^{g}_{\&}r_{\multimap}\mathsf{A}$, the value is forwarded to the right hand side. Readers familiar with proof nets will recognize the axiom links, readers familiar with game semantics will recognize pairs of Opponent moves and induced Player responses.

The implementation has no preset choice for this example, though one may obtain it by manually typing the program "g x" here – for clarity the implementation displays the hierarchical constraints between calls and returns, even though those are not part of the Petri structure.

3.1.3 Handling duplications. Next we introduce a crucial aspect of Petri structures: the need for *thread indexing*, leading to the exact definition of tokens. More precisely, we set:

Definition 3.3. The sets $\mathcal{E}$ of **exponential signatures** and $\mathcal{D}$ of **data signatures** are:

$$\mathcal{E} ::= \ell_{!}\mathcal{E} \mid r_{!}\mathcal{E} \mid \langle \mathcal{E}, \mathcal{E}\rangle \mid \blacklozenge \qquad\qquad \mathcal{D} ::= n \mid \mathtt{tt} \mid \mathtt{ff} \mid \checkmark \mid \bullet,$$

and we also write $\mathcal{E}^{*}$ for finite lists of exponential signatures, called **exponential stacks**. We use e to range over exponential signatures, s to range over exponential stacks, and d for data signatures.

The set of **tokens**, written Tok, is simply defined as $\mathcal{E}^{*} \times \mathcal{D}$, ranged over by t.

Exponential stacks originate in standard GoI token machines [Danos and Regnier 1996]. An exponential signature uniquely identifies a precise resource occurrence within the net and allows us to route accesses to distinct resources via the same address. Figure 8 shows a net involving duplication. Upon receiving $r_{\vdash}\mathsf{Q}^{-}$, the net throws two tokens corresponding to the evaluations of x. Both tokens are redirected to location 3 enabling $\ell_{\vdash}\iota^{x}_{\&}\mathsf{Q}$, but with distinct exponential stacks. Subsequently, returns $\ell_{\vdash}\iota^{x}_{\&}\mathsf{A}$ are distributed according to their exponential signature. It requires

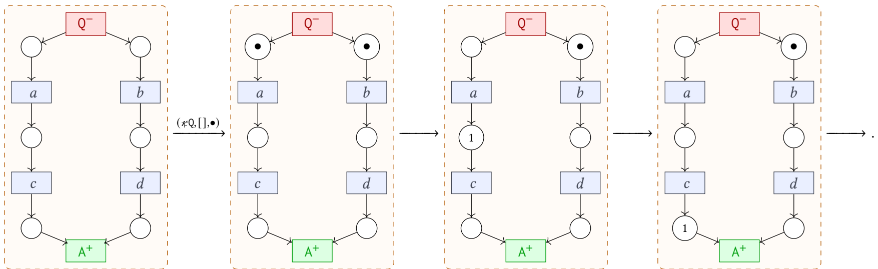


Fig. 9. A run of a Petri structure

two instances of $\ell_{\vdash} \dot{\iota}^{x}_{\&}\mathsf{A}$ with s respectively set to $[\ell_{!}\blacklozenge]$ and $[r_{!}\blacklozenge]$, routed to locations 5 and 6, in order to trigger the final return $r_{\vdash}\mathsf{A}^{+}$. The example may be ran in the implementation here.

3.1.4 The token game. Next, we formalize the token game on Petri structures.

As is familiar from the Petri net literature, a state of a Petri structure is called a *marking*:

Definition 3.4. Consider $\boldsymbol{\sigma}$ a Petri structure. A **marking** on $\boldsymbol{\sigma}$ is a finite subset of TokIL.

The set of markings on $\boldsymbol{\sigma}$ is written $\mathscr{M}(\boldsymbol{\sigma})$, and we use α, β, γ as metavariables for markings.

We use the same notation for markings as for conditions, *i.e.* $\alpha = \{([\,], \bullet)^{@1}, ([\,], \bullet)^{@2}, ([\blacklozenge], \checkmark)^{@2}\}$ is a (non-reachable) marking for the Petri structure of Figure 8. Unlike for conditions, markings allow several tokens on the same location; however we cannot have the *same* token twice on the same location – markings are sets, not multisets. It should be clear from this notation that conditions may be regarded as markings, and we shall do so silently from now on.

We define the token game, *i.e.* execution, as a walk on a labelled transition system on markings. While neutral transitions act on markings only, visible transitions *send* or *receive* tokens on addresses, regarded as *channels*. Together, an address and a token form a **move** – we define Moves $= \mathcal{M} \times \mathcal{E}^* \times \mathcal{D} \simeq \mathcal{M} \times \mathsf{Tok}$. We use m to range over moves – notice the different font from m for addresses. We now define the transition system, in two steps. For $\boldsymbol{\sigma}$ a Petri structure, we set:

Definition 3.5. We define the **instantiated transitions** (**itransitions** for short) as one of:

$$\begin{array}{rclcll}
t^0(\!|\alpha|\!) & : & \alpha & \longmapsto_{\boldsymbol{\sigma}} & \beta & \text{if } \alpha \in \mathsf{cond}(\mathsf{pre}(t)) \text{ and } \delta\langle t\rangle(\alpha) = \beta, \\
t^+(\!|\alpha|\!) & : & \alpha & \overset{m}{\longmapsto}_{\boldsymbol{\sigma}} & \emptyset & \text{if } \alpha \in \mathsf{cond}(\mathsf{pre}(t)),\ \delta\langle t\rangle(\alpha) = (\mathrm{s}, \mathrm{d}) \text{ and } m = (\partial(t), \mathrm{s}, \mathrm{d}), \\
t^-(\!|(\mathrm{s}, \mathrm{d})|\!) & : & \emptyset & \overset{m}{\longmapsto}_{\boldsymbol{\sigma}} & \beta & \text{if } m = (\partial(t), \mathrm{s}, \mathrm{d}) \text{ with } \delta\langle t\rangle(\mathrm{s}, \mathrm{d}) = \beta,
\end{array}$$

where $t \in \mathcal{T}$ has the indicated polarity; we set $\partial(t(\!|-|\!)) = m$ the *move* labelling a visible itransition.

We write $\mathsf{IT}_{\boldsymbol{\sigma}}$ the set of itransitions of $\boldsymbol{\sigma}$, and use $\mathbf{t}$ to range over those.

A visible itransition $\mathbf{t}$ is labelled with a move $m = (\mathrm{m}, \mathrm{s}, \mathrm{d}) = \partial(\mathbf{t})$. For negative itransitions we read this as the token (s, d) being *received* on m, while for positive itransitions, (s, d) is *sent* on m. The actual token game is played by instantiated transitions *in context*:

Definition 3.6. An **instantiated transition in context** (**ictransition** for short) is one of:

$$\begin{array}{rclcll}
\mathbf{t}^0 \uplus \gamma & : & \alpha \uplus \gamma & \longrightarrow_{\boldsymbol{\sigma}} & \beta \uplus \gamma & \text{if } \mathbf{t} : \alpha \longmapsto_{\boldsymbol{\sigma}} \beta, \\
\mathbf{t}^+ \uplus \gamma & : & \alpha \uplus \gamma & \overset{m}{\longrightarrow}_{\boldsymbol{\sigma}} & \gamma & \text{if } \mathbf{t} : \alpha \overset{m}{\longmapsto}_{\boldsymbol{\sigma}} \emptyset, \\
\mathbf{t}^- \uplus \gamma & : & \gamma & \overset{m}{\longrightarrow}_{\boldsymbol{\sigma}} & \gamma \uplus \beta & \text{if } \mathbf{t} : \emptyset \overset{m}{\longmapsto}_{\boldsymbol{\sigma}} \beta,
\end{array}$$

for $\gamma \in \mathscr{M}(\boldsymbol{\sigma})$ with $\gamma \cap \alpha = \gamma \cap \beta = \emptyset$.

We write $\mathsf{ITC}_{\boldsymbol{\sigma}}$ for the set of ictransitions of $\boldsymbol{\sigma}$, ranged over by $\mathfrak{t}$ (note the different font).

Definition 3.7. A **run** in $\boldsymbol{\sigma}$ is a sequence $\rho = \mathsf{t}_1 \dots \mathsf{t}_n$ of ictransitions s.t. $\mathsf{t}_i : \alpha_i \longrightarrow_{\boldsymbol{\sigma}} \alpha_{i+1}$ or $\mathsf{t}_i : \alpha_i \xrightarrow{\mathsf{m}}_{\boldsymbol{\sigma}} \alpha_{i+1}$ with $\alpha_1 = \emptyset$. We write $\rho : \emptyset \longrightarrow\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha_{n+1}$ or $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha_{n+1}$, where $s = \mathsf{m}_1 \dots \mathsf{m}_p$ lists the labels appearing in ρ. We also write $s = \mathsf{play}(\rho)$ and call s the **play** of ρ.

For example, the following is a run ρ for (the variant with full tokens of) Figure 5:

$$
\begin{array}{cccccc}
\emptyset & \xrightarrow{(\mathsf{Q},[],\bullet)} & \{([],\bullet)^{@1},([],\bullet)^{@2}\} & \longrightarrow & \{([],1)^{@3},([],\bullet)^{@2}\} & \longrightarrow \quad \{([],1)^{@5},([],\bullet)^{@2}\} \\
& \longrightarrow & \{([],1)^{@5},([],1)^{@4}\} & \longrightarrow & \{([],1)^{@5},([],1)^{@6}\} & \xrightarrow{(\mathsf{A},[],2)} \quad \emptyset
\end{array}
$$

with $\mathsf{play}(\rho) = (\mathsf{Q},[],\bullet)(\mathsf{A},[],1)$. It may be visualized as tokens walking through the Petri net as in Figure 9, ignoring the exponential stacks (which are always $[]$ in this example).

Among other things, this lets us define a notion of *may-convergence* for closed Petri structures:

Definition 3.8. Consider $\boldsymbol{\sigma}$ a closed Petri structure. We say $\boldsymbol{\sigma}$ **may converge**, written $\boldsymbol{\sigma} \Downarrow$, iff there is a run $\rho : \emptyset \longrightarrow\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha$ such that $\mathsf{play}(\rho) = (\mathsf{Q},[],\bullet)(\mathsf{A},[],\mathsf{d})$ for some $\mathsf{d} \neq \bullet$.

Next, we set toward defining the interpretation of IPA with Petri structures; *i.e.* following Section 2.2, constructing an IPA-structure with Petri structures as morphisms. The conceptual core of this endeavour is the definition of a *precategory* of Petri structures, and in particular their *composition*.

3.2 The Precategory PStruct

We build a precategory PStruct. Its *objects* are finite sets $M \subseteq \mathcal{M}$, considered as *interfaces*. A *morphism* from M to N is a Petri structure on $M +^{\vdash} N = \ell_{\vdash}(M) \uplus r_{\vdash}(N)$, up to *isomorphism*:

Definition 3.9. Consider $\boldsymbol{\sigma}, \boldsymbol{\tau}$ two Petri structures on $M \subseteq \mathcal{M}$. An **isomorphism** $\varphi : \boldsymbol{\sigma} \cong \boldsymbol{\tau}$ consists of bijections $\varphi_{\mathcal{L}} : \mathcal{L}_{\boldsymbol{\sigma}} \simeq \mathcal{L}_{\boldsymbol{\tau}}$ and $\varphi_{\mathcal{T}} : \mathcal{T}_{\boldsymbol{\sigma}} \simeq \mathcal{T}_{\boldsymbol{\tau}}$ compatible with all structure.

3.2.1 Composition of Petri structures. Fix $\boldsymbol{\sigma} \in \mathsf{PStruct}(M, N)$ and $\boldsymbol{\tau} \in \mathsf{PStruct}(N, P)$ two Petri structures; we aim to define $\boldsymbol{\tau} \odot \boldsymbol{\sigma} \in \mathsf{PStruct}(M, P)$, their *composition*. Composing $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ amounts to synchronizing $\boldsymbol{\sigma}$'s visible transitions on the *right* with $\boldsymbol{\tau}$'s visible transitions on the left:

Definition 3.10. Visible transitions $t^{\boldsymbol{\sigma}} \in \mathcal{T}^{+}_{\boldsymbol{\sigma}} \uplus \mathcal{T}^{-}_{\boldsymbol{\sigma}}$ and $t^{\boldsymbol{\tau}} \in \mathcal{T}^{+}_{\boldsymbol{\tau}} \uplus \mathcal{T}^{-}_{\boldsymbol{\tau}}$ are **synchronizable** if they have opposite polarities; and $\partial_{\boldsymbol{\sigma}}(t^{\boldsymbol{\sigma}}) = r_{\vdash}\mathsf{m}$ while $\partial_{\boldsymbol{\tau}}(t^{\boldsymbol{\tau}}) = \ell_{\vdash}\mathsf{m}$ for some $\mathsf{m} \in N$. We define the set

$$
\begin{array}{rcl}
\mathcal{T}_{\boldsymbol{\tau}} \circledast \mathcal{T}_{\boldsymbol{\sigma}} & = & \{\ell^{\odot}(t) \mid t \in \mathcal{T}^{0}_{\boldsymbol{\sigma}} \uplus \mathcal{T}^{p,\ell_{\vdash}}_{\boldsymbol{\sigma}}\} \uplus \{r^{\odot}(t) \mid t \in \mathcal{T}^{0}_{\boldsymbol{\tau}} \uplus \mathcal{T}^{p,r_{\vdash}}_{\boldsymbol{\tau}}\} \uplus \\
& & \{t^{\boldsymbol{\sigma}} \circledast t^{\boldsymbol{\tau}} \mid t^{\boldsymbol{\sigma}} \in \mathcal{T}_{\boldsymbol{\sigma}}, t^{\boldsymbol{\tau}} \in \mathcal{T}_{\boldsymbol{\tau}} \text{ synchronizable.}\}
\end{array}
$$

with $\mathcal{T}^{p,\imath}_{\boldsymbol{\sigma}}$ transitions of polarity $p \in \{-,+\}$ and label $\partial(t) = \imath\mathsf{m}$ for some $\mathsf{m} \in \mathcal{M}$; idem for $\mathcal{T}^{p,\imath}_{\boldsymbol{\tau}}$.

Intuitively, $\mathcal{T}_{\boldsymbol{\tau}} \circledast \mathcal{T}_{\boldsymbol{\sigma}}$ imports an unsynchronized transition t from $\boldsymbol{\sigma}$ as $\ell^{\odot}(t)$, an unsynchronized transition t from $\boldsymbol{\tau}$ as $r^{\odot}(t)$, but also has a new transition $t^{\boldsymbol{\sigma}} \circledast t^{\boldsymbol{\tau}}$ for every synchronizable pair.

We have used $\ell^{\odot}$ and $r^{\odot}$ to keep transitions from $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ disjoint. From now on, if X and Y are sets, we write $X +^{\odot} Y = \ell^{\odot}(X) \uplus r^{\odot}(Y)$. We shall use the same convention with other tags later on – or simply write $X + Y = \ell(X) \uplus r(Y)$. Now, we may finally define the **composition** $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ as:

Definition 3.11. We set $\mathcal{L}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} = \mathcal{L}_{\boldsymbol{\sigma}} +^{\odot} \mathcal{L}_{\boldsymbol{\tau}}$; $\mathcal{T}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} = \mathcal{T}_{\boldsymbol{\tau}} \circledast \mathcal{T}_{\boldsymbol{\sigma}}$; $\mathcal{T}^{p}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} = \mathcal{T}^{p,\ell_{\vdash}}_{\boldsymbol{\sigma}} +^{\odot} \mathcal{T}^{p,r_{\vdash}}_{\boldsymbol{\tau}}$ where $p \in \{+,-\}$; $\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\ell^{\odot}(t)) = \partial_{\boldsymbol{\sigma}}(t)$ and $\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(r^{\odot}(t)) = \partial_{\boldsymbol{\tau}}(t)$. Conditions are in Figure 10, and:

$$
\begin{array}{rclrcl}
\delta_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}\langle \ell^{\odot}(t)\rangle(\ell^{\odot}(\alpha)) & = & \ell^{\odot}(\delta_{\boldsymbol{\sigma}}\langle t\rangle(\alpha)) & \delta_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}\langle t^{+} \circledast t^{-}\rangle(\ell^{\odot}(\alpha)) & = & r^{\odot}((\delta_{\boldsymbol{\tau}}\langle t^{-}\rangle \circ \delta_{\boldsymbol{\sigma}}\langle t^{+}\rangle)(\alpha)) \\
\delta_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}\langle r^{\odot}(t)\rangle(r^{\odot}(\beta)) & = & r^{\odot}(\delta_{\boldsymbol{\tau}}\langle t\rangle(\beta)) & \delta_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}\langle t^{-} \circledast t^{+}\rangle(r^{\odot}(\beta)) & = & \ell^{\odot}((\delta_{\boldsymbol{\sigma}}\langle t^{-}\rangle \circ \delta_{\boldsymbol{\tau}}\langle t^{+}\rangle)(\beta))
\end{array}
$$

where $\ell^{\odot}$ and $r^{\odot}$ are applied to $\alpha \in \mathsf{cond}_{\boldsymbol{\sigma}}$ and $\beta \in \mathsf{cond}_{\boldsymbol{\tau}}$ by retagging locations, *i.e.* with $\ell^{\odot}(\alpha) = \{(\mathsf{s},\mathsf{d})^{@\ell^{\odot}(l)} \mid (\mathsf{s},\mathsf{d})^{@l} \in \alpha\} \in \mathsf{cond}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}$ and $r^{\odot}(\beta) = \{(\mathsf{s},\mathsf{d})^{@r^{\odot}(l)} \mid (\mathsf{s},\mathsf{d})^{@l} \in \beta\} \in \mathsf{cond}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}$.

$$
\begin{array}{rclcrcl}
\mathsf{pre}_{\tau\odot\sigma}(\ell^{\odot}(t)) & = & \ell^{\odot}(\mathsf{pre}_{\sigma}(t)) & \quad & \mathsf{post}_{\tau\odot\sigma}(\ell^{\odot}(t)) & = & \ell^{\odot}(\mathsf{post}_{\sigma}(t)) \\
\mathsf{pre}_{\tau\odot\sigma}(\reflectbox{ℓ}^{\odot}(t)) & = & \reflectbox{ℓ}^{\odot}(\mathsf{pre}_{\tau}(t)) & & \mathsf{post}_{\tau\odot\sigma}(\reflectbox{ℓ}^{\odot}(t)) & = & \reflectbox{ℓ}^{\odot}(\mathsf{post}_{\tau}(t)) \\
\mathsf{pre}_{\tau\odot\sigma}(t^{+} \circledast t^{-}) & = & \ell^{\odot}(\mathsf{pre}_{\sigma}(t^{+})) & & \mathsf{post}_{\tau\odot\sigma}(t^{+} \circledast t^{-}) & = & \reflectbox{ℓ}^{\odot}(\mathsf{post}_{\tau}(t^{-})) \\
\mathsf{pre}_{\tau\odot\sigma}(t^{-} \circledast t^{+}) & = & \reflectbox{ℓ}^{\odot}(\mathsf{pre}_{\tau}(t^{+})) & & \mathsf{post}_{\tau\odot\sigma}(t^{-} \circledast t^{+}) & = & \ell^{\odot}(\mathsf{post}_{\sigma}(t^{-}))
\end{array}
$$

Fig. 10. Pre-conditions and post-conditions for the composition

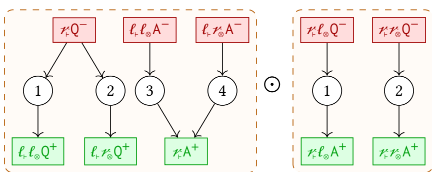


Fig. 11. A composition yielding Figure 5

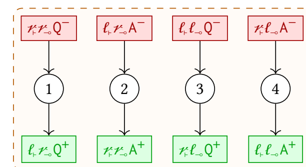


Fig. 12. The Petri structure $\mathbf{cc}_{!\mathbf{N}\multimap\mathbf{N}}$

We show in Figure 11 an example composition, yielding (up to iso) the structure of Figure 5 – we omit the transition tables in order to save space. Notice how the two copies of Figure 4 *glue* together the disconnected components – for *calls* and *returns* – of the left hand side operand.

Intuitively, runs on $\tau \odot \sigma$ happen as follows: tokens first appear via negative transitions on either side. The token game is, at first, played independently in σ and τ. As soon as σ wishes to play a positive transition on the right (resp. τ wishes to play a positive transition on the left), it synchronizes with the matching negative transition on the other side – if it exists – and the resulting (neutral) transition has transition function the composite as for the two compounds. The effect of that synchronization is that tokens "jump" between σ and τ, following the control flow.

3.2.2 The copycat Petri structure. Next, PStruct requires an identity: the *copycat* Petri structure. *Copycat* exchanges tokens between left and right, forwarding negative moves on either side to the matching positive move on the other side, keeping tokens otherwise unchanged.

Definition 3.12. For $M \subseteq \mathcal{M}$ finite, we define the **copycat Petri structure** on M, written $\mathbf{cc}_M$. Its *locations* are $\mathcal{L}_{\mathbf{cc}_M} = M$, its *transitions* are $\mathcal{T}_{\mathbf{cc}_M} = M \times \{\ell, \reflectbox{ℓ}\}$ with polarities as in

$$\mathcal{T}^{+}_{\mathbf{cc}_M} = (M^{+} \times \{\reflectbox{ℓ}\}) \uplus (M^{-} \times \{\ell\}) \qquad \mathcal{T}^{-}_{\mathbf{cc}_M} = (M^{-} \times \{\reflectbox{ℓ}\}) \uplus (M^{+} \times \{\ell\})$$

and no neutral transition. We set $\partial(\mathsf{m}, \ell) = \ell_{\!\cdot}\mathsf{m}$ and $\partial(\mathsf{m}, \reflectbox{ℓ}) = \reflectbox{ℓ}_{\!\cdot}\mathsf{m}$; we set $\mathsf{pre}(\mathsf{m}^{+}, \reflectbox{ℓ})$, $\mathsf{pre}(\mathsf{m}^{-}, \ell)$, $\mathsf{post}(\mathsf{m}^{-}, \reflectbox{ℓ})$ and $\mathsf{post}(\mathsf{m}^{+}, \ell)$ to $\{\mathsf{m}\}$ and pre and post returning $\emptyset$ elsewhere. Finally:

$$
\begin{array}{rclcrcl}
\delta\langle(\mathsf{m}^{+}, \reflectbox{ℓ})\rangle(\{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}) & = & (\mathsf{s}, \mathsf{d}) & \quad & \delta\langle(\mathsf{m}^{-}, \reflectbox{ℓ})\rangle(\mathsf{s}, \mathsf{d}) & = & \{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\} \\
\delta\langle(\mathsf{m}^{-}, \ell)\rangle(\{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}) & = & (\mathsf{s}, \mathsf{d}) & & \delta\langle(\mathsf{m}^{+}, \ell)\rangle(\mathsf{s}, \mathsf{d}) & = & \{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}\,.
\end{array}
$$

As an example, we show in Figure 12 the Petri structure $\mathbf{cc}_{!\mathbf{N}\multimap\mathbf{N}}$, writing $!\mathbf{N} \multimap \mathbf{N}$ for the set $\{\ell_{\multimap}\mathsf{Q}^{+}, \ell_{\multimap}\mathsf{A}^{-}, \reflectbox{ℓ}_{\multimap}\mathsf{Q}^{-}, \reflectbox{ℓ}_{\multimap}\mathsf{A}^{+}\}$ which shall arise as the interpretation of $\mathbb{N} \to \mathbb{N}$.

Associativity of composition up to iso is direct, which makes PStruct a precategory. Note that copycat is *not* neutral for composition up to iso: composing with copycat yields a structure with strictly more nodes – it is in fact the reason why we based IPA-structures on precategories rather than categories. Copycat will be neutral for composition only *w.r.t. unfolding*, see Section 5.2.

3.3 PStruct **as an** IPA**-structure: Constructions, Operations**

Now, we introduce the IPA-structure operations for Petri structures, following Definition 2.1.

For PStruct we fix $\mathsf{PStruct}_{\bullet} = \mathsf{PStruct}_0$: we do not need to distinguish well-opened objects.

$$
\begin{array}{rclcrcl}
\mathsf{pre}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\ell^{\otimes}(t)) & = & \ell^{\otimes}(\mathsf{pre}_{\boldsymbol{\sigma}}(t)) & \qquad & \mathsf{post}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\ell^{\otimes}(t)) & = & \ell^{\otimes}(\mathsf{post}_{\boldsymbol{\sigma}}(t)) \\
\mathsf{pre}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\mathscr{r}^{\otimes}(t)) & = & \mathscr{r}^{\otimes}(\mathsf{pre}_{\boldsymbol{\tau}}(t)) & & \mathsf{post}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\mathscr{r}^{\otimes}(t)) & = & \mathscr{r}^{\otimes}(\mathsf{post}_{\boldsymbol{\tau}}(t))
\end{array}
$$

Fig. 13. Pre-conditions and post-conditions for the tensor operation

3.3.1 Constructions. First of all, the *constructions* of Definition 2.1 are applied to finite sets of addresses simply by applying the corresponding injections from Definition 3.2. In other words we set $M \otimes N = M +^{\otimes} N$; $\&_{x\in V} M_x = \uplus_{x\in V}\, \iota_{\&}^{x}(M_x)$; $M \multimap N = M +^{\multimap} N$ and $!M = M$.

For basic types, we set $\mathbf{U}, \mathbf{B}$ and $\mathbf{N}$ as $\mathbf{G} = \{\mathtt{Q}^-, \mathtt{A}^+\}$. We postpone $\mathbf{V}$ and $\mathbf{S}$ to Section 3.5.

3.3.2 Tensor. Next, we define *tensor* of Petri structures. Fix $\boldsymbol{\sigma}, \boldsymbol{\tau}$ Petri structures:

Definition 3.13. Consider $\boldsymbol{\sigma} \in \mathsf{PStruct}(M_1, N_1)$ and $\boldsymbol{\tau} \in \mathsf{PStruct}(M_2, N_2)$.
We set $\mathcal{L}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} = \mathcal{L}_{\boldsymbol{\sigma}} +^{\otimes} \mathcal{L}_{\boldsymbol{\tau}}$; $\mathcal{T}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} = \mathcal{T}_{\boldsymbol{\sigma}} +^{\otimes} \mathcal{T}_{\boldsymbol{\tau}}$ with $\mathcal{T}^{p}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} = \mathcal{T}^{p}_{\boldsymbol{\sigma}} +^{\otimes} \mathcal{T}^{p}_{\boldsymbol{\tau}}$ for $p \in \{+, -\}$;

$$
\begin{array}{rcll@{\qquad}rcll}
\partial_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\ell^{\otimes}(t)) & = & \ell_{\vdash}\ell_{\otimes}\mathsf{m} & (\partial_{\boldsymbol{\sigma}}(t) = \ell_{\vdash}\mathsf{m}) & \partial_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\mathscr{r}^{\otimes}(t)) & = & \ell_{\vdash}\mathscr{r}_{\otimes}\mathsf{m} & (\partial_{\boldsymbol{\tau}}(t) = \ell_{\vdash}\mathsf{m}) \\
\partial_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\ell^{\otimes}(t)) & = & \mathscr{r}_{\vdash}\ell_{\otimes}\mathsf{m} & (\partial_{\boldsymbol{\sigma}}(t) = \mathscr{r}_{\vdash}\mathsf{m}) & \partial_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\mathscr{r}^{\otimes}(t)) & = & \mathscr{r}_{\vdash}\mathscr{r}_{\otimes}\mathsf{m} & (\partial_{\boldsymbol{\tau}}(t) = \mathscr{r}_{\vdash}\mathsf{m})
\end{array}
$$

pre- and post-conditions in Figure 13, and for the transition table we set:

$$
\delta_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}\langle \ell^{\otimes}(t)\rangle(\ell^{\otimes}(\alpha)) = \ell^{\otimes}(\delta_{\boldsymbol{\sigma}}\langle t\rangle(\alpha)) \qquad \delta_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}\langle \mathscr{r}^{\otimes}(t)\rangle(\mathscr{r}^{\otimes}(\alpha)) = \mathscr{r}^{\otimes}(\delta_{\boldsymbol{\tau}}\langle t\rangle(\alpha))
$$

with $\ell^{\otimes}, \mathscr{r}^{\otimes}$ applied on conditions as in Definition 3.11. This yields $\boldsymbol{\sigma}\otimes\boldsymbol{\tau} \in \mathsf{PStruct}(M_1\otimes M_2, N_1\otimes N_2)$.

This simply puts $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ side by side without interaction. As an example, the Petri structure on the right hand side of the composition symbol in Figure 11 is $\boldsymbol{\sigma} \otimes \boldsymbol{\sigma}$ for $\boldsymbol{\sigma}$ in Figure 4.

3.3.3 Currying. Rather than merely introducing *currying*, we introduce a general operation to *rename* the addresses associated with visible transitions in a Petri structure:

Definition 3.14. Take $\boldsymbol{\sigma}$ a Petri structure on M and $f : \mathcal{M} \rightharpoonup \mathcal{M}$ s.t. $M \subseteq \mathrm{dom}(f), f(M) \subseteq N$.
The **renaming** $\boldsymbol{\sigma}[f]$, a Petri structure on N, is as $\boldsymbol{\sigma}$ except for $\partial_{\boldsymbol{\sigma}[f]}(t) = f(\partial_{\boldsymbol{\sigma}}(t))$.

The net itself is not affected by the change, only the labelling function for visible transition. Now:

Definition 3.15. Consider $\boldsymbol{\sigma} \in \mathsf{PStruct}(!(\&[\Gamma, x : A, \Delta]), O)$. We define the function $\Lambda_x : \mathcal{M}_{\vdash} \rightharpoonup \mathcal{M}_{\vdash}$ by $\Lambda_x(\mathscr{r}_{\vdash}\mathsf{m}) = \mathscr{r}_{\vdash}\mathscr{r}_{\multimap}\mathsf{m}$, $\Lambda_x(\ell_{\vdash}\iota_{\&}^{x}\mathsf{m}) = \mathscr{r}_{\vdash}\ell_{\multimap}\mathsf{m}$, and $\Lambda_x(\mathsf{m}) = \mathsf{m}$ otherwise.
Then, setting $\Lambda^{\Gamma,\Delta}_{x:A,O}(\boldsymbol{\sigma}) = \boldsymbol{\sigma}[\Lambda_x]$, we obtain $\Lambda^{\Gamma,\Delta}_{x:A,O}(\boldsymbol{\sigma}) \in \mathsf{PStruct}(!(\&[\Gamma, \Delta]), !A \multimap O)$.

This reassigns visible transitions corresponding to variable x to the left hand side of $\multimap$ on the right hand side of $\vdash$. Transitions initially assigned to the right hand side must also be relabelled, but the rest are unchanged. The net itself (*i.e.* the graph) remains the same.

3.3.4 Promotion. The final operation on Petri strategies is *promotion*, for a Petri structure $\boldsymbol{\sigma}$:

Definition 3.16. Consider $\boldsymbol{\sigma} \in \mathsf{PStruct}(!M, N)$. We set $\mathcal{L}_{\boldsymbol{\sigma}^{\dagger}} = \mathcal{L}_{\boldsymbol{\sigma}}$, $\mathcal{T}_{\boldsymbol{\sigma}^{\dagger}} = \mathcal{T}_{\boldsymbol{\sigma}}$ with the same polarities, $\partial_{\boldsymbol{\sigma}^{\dagger}} = \partial_{\boldsymbol{\sigma}}$, and pre- and post-conditions are also unchanged. Finally, the *transition table* is:

$$
\begin{array}{rcll}
\delta_{\boldsymbol{\sigma}^{\dagger}}\langle t^{0}\rangle(\mathsf{e} :: \alpha) & = & \mathsf{e} :: \beta & \text{if } \delta_{\boldsymbol{\sigma}}\langle t\rangle(\alpha) = \beta \\
\delta_{\boldsymbol{\sigma}^{\dagger}}\langle t^{+}\rangle(\mathsf{e} :: \alpha) & = & (\mathsf{e} :: \mathsf{s}, \mathsf{d}) & \text{if } \partial_{\boldsymbol{\sigma}}(t) = \mathscr{r}_{\vdash}- \text{ and } \delta_{\boldsymbol{\sigma}}\langle t\rangle(\alpha) = (\mathsf{s}, \mathsf{d}) \\
\delta_{\boldsymbol{\sigma}^{\dagger}}\langle t^{+}\rangle(\mathsf{e} :: \alpha) & = & (\langle \mathsf{e}, \mathsf{e}'\rangle :: \mathsf{s}, \mathsf{d}) & \text{if } \partial_{\boldsymbol{\sigma}}(t) = \ell_{\vdash}- \text{ and } \delta_{\boldsymbol{\sigma}}\langle t\rangle(\alpha) = (\mathsf{e}' :: \mathsf{s}, \mathsf{d}) \\
\delta_{\boldsymbol{\sigma}^{\dagger}}\langle t^{-}\rangle(\mathsf{e} :: \mathsf{s}, \mathsf{d}) & = & \mathsf{e} :: \alpha & \text{if } \partial_{\boldsymbol{\sigma}}(t) = \mathscr{r}_{\vdash}- \text{ and } \delta_{\boldsymbol{\sigma}}\langle t\rangle(\mathsf{s}, \mathsf{d}) = \alpha \\
\delta_{\boldsymbol{\sigma}^{\dagger}}\langle t^{-}\rangle(\langle \mathsf{e}, \mathsf{e}'\rangle :: \mathsf{s}, \mathsf{d}) & = & \mathsf{e} :: \alpha & \text{if } \partial_{\boldsymbol{\sigma}}(t) = \ell_{\vdash}- \text{ and } \delta_{\boldsymbol{\sigma}}\langle t\rangle(\mathsf{e}' :: \mathsf{s}, \mathsf{d}) = \alpha
\end{array}
$$

where $\partial_{\boldsymbol{\sigma}}(t) = \iota-$ means $\partial_{\boldsymbol{\sigma}}(t) = \iota\mathsf{m}$ for some $\mathsf{m} \in \mathcal{M}$, and $\mathsf{e} :: \alpha$ is $\{(\mathsf{e} :: \mathsf{s}_i, \mathsf{d}_i)^{@l_i} \mid (\mathsf{s}_i, \mathsf{d}_i)^{@l_i} \in \alpha\}$.
With this definition, we obtain $\boldsymbol{\sigma}^{\dagger} \in \mathsf{PStruct}(!M, !N)$.

In contrast with currying, promotion *only* affects the transition table, not the other components: promotion only manipulates the exponential signatures, which are only present in the tokens.

Those rules are similar to those dealing with promotion in standard GoI token machines [Danos and Regnier 1996]; their behaviour is somewhat subtle. The central idea is that $\boldsymbol{\sigma}^\dagger$ has – implicitly – countably many copies of $\boldsymbol{\sigma}$ running in parallel. In reality all tokens flow through the same net, but copies are kept apart by the exponential signature added as a new layer of the exponential stack (the rest of the exponential stack concerns promotions deeper within the net). Finally, when interacting with the outside world on the left, this new layer of the exponential stack must be *merged* with the previous top of the stack (an effect known as *digging* in linear logic jargon).

3.4 PStruct as an IPA-structure: Stateless Primitives

Next, we describe all the primitives involved in the interpretation of the λ-calculus and recursion.

3.4.1 Variable, evaluation. First, the *variable* is simply a copycat set to component $\iota^{x}_{\&}$ of the context.

Definition 3.17. Consider $M \subseteq \mathcal{M}$ finite, and $x \in \mathsf{Var}$.

We define $\mathbf{var}_{x:M}$ as $\mathcal{L}_{\mathbf{var}_{x:M}} = \mathcal{L}_{\mathbf{cc}_M}$, $\mathcal{T}_{\mathbf{var}_{x:M}} = \mathcal{T}_{\mathbf{cc}_M}$, with the same pre- and post-conditions as for $\mathbf{cc}_M$. We set $\partial_{\mathbf{var}_{x:M}}(\mathsf{m}, \ell) = \ell_{\vdash}\iota^{x}_{\&}\mathsf{m}$ and $\partial_{\mathbf{var}_{x:M}}(\mathsf{m}, r) = r_{\vdash}\mathsf{m}$. Finally, the transition table is:

$$
\begin{array}{rclcrcl}
\delta\langle(\mathsf{m}^+, r)\rangle(\{(\mathsf{s},\mathsf{d})^{@\mathsf{m}}\}) & = & (\mathsf{s},\mathsf{d}) & \quad & \delta\langle(\mathsf{m}^-, r)\rangle(\mathsf{s},\mathsf{d}) & = & \{(\mathsf{s},\mathsf{d})^{@\mathsf{m}}\} \\
\delta\langle(\mathsf{m}^-, \ell)\rangle(\{(\mathsf{s},\mathsf{d})^{@\mathsf{m}}\}) & = & (\blacklozenge :: \mathsf{s},\mathsf{d}) & & \delta\langle(\mathsf{m}^+, \ell)\rangle(\blacklozenge :: \mathsf{s},\mathsf{d}) & = & \{(\mathsf{s},\mathsf{d})^{@\mathsf{m}}\}\,.
\end{array}
$$

Recall that terms $\Gamma \vdash M : A$ are meant to be interpreted as morphisms $[\![M]\!] \in \mathcal{C}(![\![\Gamma]\!], [\![A]\!])$. The ! explains the $\blacklozenge$ in the transition table, which corresponds to *dereliction* in linear logic terminology.

Note that the top two of these transitions do not affect the token: they only forward it following the structure of the net. From now on, we call **trivial** such transitions and omit them for succinctness.

Evaluation, used for application, is also a copycat:

Definition 3.18. For $M, N \subseteq \mathcal{M}$ finite sets, $\mathbf{ev}_{M,N} = \mathbf{cc}_{M \multimap N}[\Omega]$, where $\Omega : \mathcal{M} \rightharpoonup \mathcal{M}$ is $\Omega(r_{\vdash}r_{\multimap}\mathsf{m}) = r_{\vdash}\mathsf{m}$, $\Omega(r_{\vdash}\ell_{\multimap}\mathsf{m}) = \ell_{\vdash}r_{\otimes}\mathsf{m}$, $\Omega(\ell_{\vdash}r_{\multimap}\mathsf{m}) = \ell_{\vdash}\ell_{\otimes}r_{\multimap}\mathsf{m}$, $\Omega(\ell_{\vdash}\ell_{\multimap}\mathsf{m}) = \ell_{\vdash}\ell_{\otimes}\ell_{\multimap}\mathsf{m}$, and $\Omega(\mathsf{m}) = \mathsf{m}$ otherwise.

3.4.2 Other stateless primitives. For finite $M \subseteq \mathcal{M}$, the (binary) **contraction** $\mathbf{c}_M$ has $\mathcal{L}_{\mathbf{c}_M} = M$, transitions $\mathcal{T}_{\mathbf{c}_M} = M +^{\vdash} (M +^{\otimes} M)$ with polarity as in Definition 3.2. The net (*i.e.* pre- and post-conditions) appears in Figure 15a, and the transition rules in Figure 14. The Petri structure behaves as in Figure 8, using the injections $\ell_!$ and $r_!$ from exponential signatures to record the routing back to the node from which a request originates. The n-ary contraction, defined likewise, is omitted – it may also be obtained by induction, composing binary contractions.

The **fixpoint combinator** Y_M is similar: it has $\mathcal{L}_{\mathrm{Y}_M} = M$, $\mathcal{T}_{\mathrm{Y}_M} = (M +^{\multimap} M) +^{\vdash} M$, net in Figure 15b and transition rules in Figure 14. The net has no loops, but loops may arise by composition.

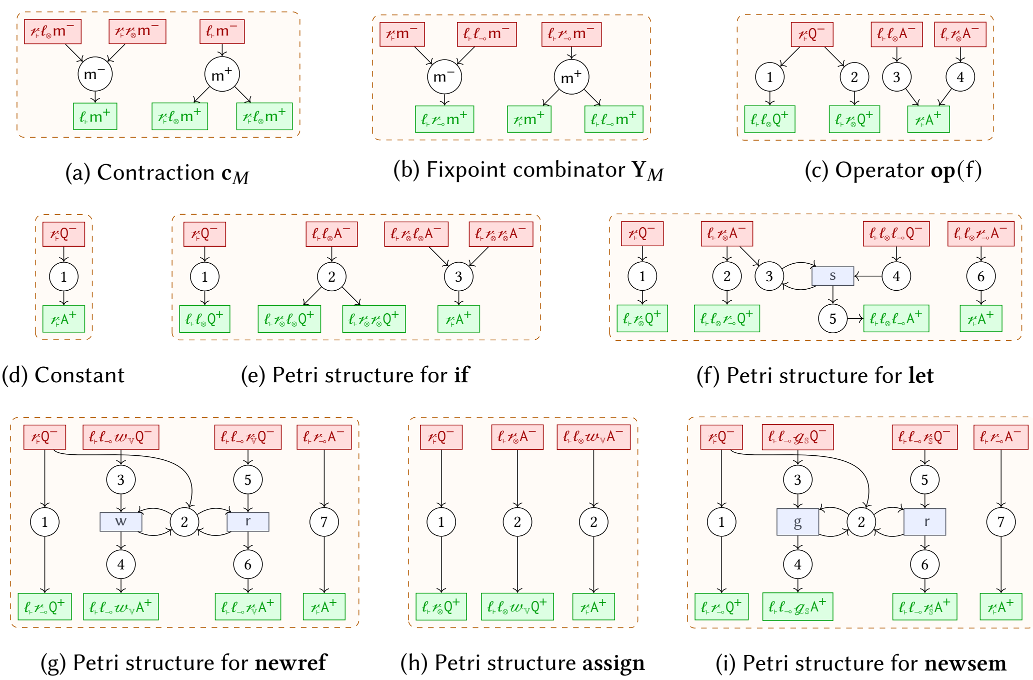
Constants, **conditionals** and **operations** are introduced in Figures 15d, 15e and 15c respectively, with transition rules in Figure 14. Hopefully their behaviour is clear at this point.

Finally, the **let** has net in Figure 15f and transition rules in Figure 14. We explain its behaviour, which is more subtle. Recall that Definition 2.1 requires $\mathbf{let}_{\mathbf{X},\mathbf{Y}} \in \mathcal{C}((!\mathbf{X} \multimap \mathbf{Y}) \otimes \mathbf{X}, \mathbf{Y})$, and in this case $\mathbf{let} \in \mathsf{PStruct}((!\mathbf{G} \multimap \mathbf{G}) \otimes \mathbf{G}, \mathbf{G})$. Upon being called with $r_{\vdash}\mathsf{Q}^-$, **let** triggers the evaluation of its argument at $\ell_{\vdash}r_{\otimes}\mathsf{Q}^+$ – this evaluation is performed *once*, and will never happen again. Upon receiving a value with $\ell_{\vdash}r_{\otimes}\mathsf{A}^-$, **let** does two things: it gives the control to its function argument at $\ell_{\vdash}\ell_{\otimes}r_{\multimap}\mathsf{Q}^+$. In parallel, it *memoizes* the value by storing it into location 3. This value is then read each time the function calls its argument. In essence, **let** is a *read-only memory cell.*

This naturally brings us to our final primitives, dealing with *mutable state* and *semaphores.*

$$
\begin{array}{llll}
\delta_{\mathbf{c}_M}\langle r_{\vdash}\ell_{\otimes}\mathsf{m}^-\rangle(\mathsf{e} :: \mathsf{s}, \mathsf{d}) & = \{(\ell_{!}\mathsf{e} :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^-}\} & \delta_{\mathbf{Y}_M}\langle r_{\vdash}\mathsf{m}^-\rangle(\mathsf{s}, \mathsf{d}) & = \{(\ell_{!}\blacklozenge :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^-}\} \\
\delta_{\mathbf{c}_M}\langle r_{\vdash}r_{\otimes}\mathsf{m}^-\rangle(\mathsf{e} :: \mathsf{s}, \mathsf{d}) & = \{(r_{!}\mathsf{e} :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^-}\} & \delta_{\mathbf{Y}_M}\langle \ell_{\vdash}\ell_{\multimap}\mathsf{m}^-\rangle(\mathsf{e}_1 :: \mathsf{e}_2 :: \mathsf{s}, \mathsf{d}) & = \{(r_{!}\langle \mathsf{e}_1, \mathsf{e}_2\rangle :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^-}\} \\
\delta_{\mathbf{c}_M}\langle r_{\vdash}\ell_{\otimes}\mathsf{m}^+\rangle(\{(\ell_{!}\mathsf{e} :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^+}\}) & = (\mathsf{e} :: \mathsf{s}, \mathsf{d}) & \delta_{\mathbf{Y}_M}\langle r_{\vdash}\mathsf{m}^+\rangle(\{(\ell_{!}\blacklozenge :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^+}\}) & = (\mathsf{s}, \mathsf{d}) \\
\delta_{\mathbf{c}_M}\langle r_{\vdash}r_{\otimes}\mathsf{m}^+\rangle(\{(r_{!}\mathsf{e} :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^+}\}) & = (\mathsf{e} :: \mathsf{s}, \mathsf{d}) & \delta_{\mathbf{Y}_M}\langle \ell_{\vdash}\ell_{\multimap}\mathsf{m}^+\rangle(\{(r_{!}\langle e_1, e_2\rangle :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^+}\}) & = (\mathsf{e}_1 :: \mathsf{e}_2 :: \mathsf{s}, \mathsf{d}) \\
\\
\delta_{\mathbf{if}}\langle \ell_{\vdash}r_{\otimes}\ell_{\otimes}\mathsf{Q}^+\rangle(\{([], \mathsf{tt})^{@2}\}) & = ([], \bullet) & \delta_{\mathbf{op}(\mathsf{f})}\langle r_{\vdash}\mathsf{Q}^-\rangle([], \mathsf{d}) & = \{([], \mathsf{d})^{@1}, ([], \mathsf{d})^{@2}\} \\
\delta_{\mathbf{if}}\langle \ell_{\vdash}r_{\otimes}\ell_{\otimes}\mathsf{Q}^+\rangle(\{([], \mathsf{ff})^{@2}\}) & = ([], \bullet) & \delta_{\mathbf{op}(\mathsf{f})}\langle r_{\vdash}\mathsf{A}^+\rangle(\{([], \mathsf{d})^{@3}, ([], \mathsf{d}')^{@4}) & = ([], \mathsf{f}(\mathsf{d}, \mathsf{d}')) \\
\\
\delta_{\mathbf{newref}}\langle r_{\vdash}\mathsf{Q}^-\rangle([], \bullet) & = \{([], \bullet)^{@1}, ([], 0)^{@2}\} & \delta_{\mathbf{newsem}}\langle r_{\vdash}\mathsf{Q}^-\rangle([], \bullet) & = \{([], \bullet)^{@1}, ([], \mathsf{tt})^{@2}\} \\
\delta_{\mathbf{newref}}\langle \mathsf{w}\rangle(\{([\mathsf{e}], \mathsf{d})^{@3}, (\_, \_)^{@2}\}) & = \{([\mathsf{e}], \mathsf{d})^{@2}, ([\mathsf{e}], \checkmark)^{@4}\} & \delta_{\mathbf{newsem}}\langle \mathsf{g}\rangle(\{([\mathsf{e}], \bullet)^{@3}, (\_, \mathsf{tt})^{@2}\}) & = \{([\mathsf{e}], \mathsf{ff})^{@2}, ([\mathsf{e}], \checkmark)^{@4}\} \\
\delta_{\mathbf{newref}}\langle \mathsf{r}\rangle(\{([\mathsf{e}], \bullet)^{@5}, (\_, \mathsf{d})^{@2}\}) & = \{([\mathsf{e}], \mathsf{d})^{@2}, ([\mathsf{e}], \mathsf{d})^{@6}\} & \delta_{\mathbf{newsem}}\langle \mathsf{r}\rangle(\{([\mathsf{e}], \bullet)^{@3}, (\_, \mathsf{ff})^{@2}\}) & = \{([\mathsf{e}], \mathsf{tt})^{@2}, ([\mathsf{e}], \checkmark)^{@6}\} \\
\\
\delta_{\mathbf{let}}\langle \ell_{\vdash}r_{\otimes}\mathsf{A}^-\rangle([], \mathsf{d}) & = \{([], \mathsf{d})^{@2}, ([], \mathsf{d})^{@3}\} & \delta_{\mathbf{let}}\langle \mathsf{s}\rangle(\{([], \mathsf{d})^{@3}, (\mathsf{s}, \bullet)^{@4}\}) & = \{([], \mathsf{d})^{@3}, (\mathsf{s}, \mathsf{d})^{@5}\}
\end{array}
$$

Fig. 14. Transition tables for IPA-structure primitives



(a) Contraction $\mathbf{c}_M$ (b) Fixpoint combinator $\mathbf{Y}_M$ (c) Operator $\mathbf{op}(\mathsf{f})$

(d) Constant (e) Petri structure for **if** (f) Petri structure for **let**

(g) Petri structure for **newref** (h) Petri structure **assign** (i) Petri structure for **newsem**

Fig. 15. The nets of Petri structures for IPA primitives

3.5 PStruct as an IPA-structure: Stateful Primitives

For types $\mathbb{V}$ and $\mathbb{S}$ we have addresses $\mathbf{V} = \{w_{\mathbb{V}}\mathsf{Q}^-, w_{\mathbb{V}}\mathsf{A}^+, r_{\mathbb{V}}\mathsf{Q}^-, r_{\mathbb{V}}\mathsf{A}^+\}$ and $\mathbf{S} = \{g_{\mathbb{S}}\mathsf{Q}^-, g_{\mathbb{S}}\mathsf{A}^+, r_{\mathbb{S}}\mathsf{Q}^-, r_{\mathbb{S}}\mathsf{A}^+\}$, where $w_{\mathbb{V}}$ is the address for *write* requests, $r_{\mathbb{V}}$ for *read* requests, $g_{\mathbb{S}}$ for *grab*, $r_{\mathbb{S}}$ for *release*.

3.5.1 Reference and semaphore queries. Recall that Definition 2.1 requires primitives

$$\mathbf{assign} \in \mathcal{C}(\mathbf{V} \otimes \mathbf{N}, \mathbf{U}), \quad \mathbf{deref} \in \mathcal{C}(\mathbf{V}, \mathbf{N}), \quad \mathbf{grab} \in \mathcal{C}(\mathbf{S}, \mathbf{U}), \quad \mathbf{release} \in \mathcal{C}(\mathbf{S}, \mathbf{U})$$

for reference and semaphore queries. Among these, we set $\mathbf{deref} = \mathbf{cc}_{\mathbf{N}}[\ell_{\vdash}\mathsf{m} \mapsto \ell_{\vdash}r_{\mathbb{V}}\mathsf{m}, r_{\vdash}\mathsf{m} \mapsto r_{\vdash}\mathsf{m}]$, $\mathbf{grab} = \mathbf{cc}_{\mathbf{U}}[\ell_{\vdash}\mathsf{m} \mapsto \ell_{\vdash}g_{\mathbb{S}}\mathsf{m}, r_{\vdash}\mathsf{m} \mapsto r_{\vdash}\mathsf{m}]$ and $\mathbf{release} = \mathbf{cc}_{\mathbf{U}}[\ell_{\vdash}\mathsf{m} \mapsto \ell_{\vdash}r_{\mathbb{S}}\mathsf{m}, r_{\vdash}\mathsf{m} \mapsto r_{\vdash}\mathsf{m}]$ copycat strategies simply accessing the matching component of the reference or semaphore.

The remaining query **assign** is more elaborate: upon being evaluated, it sends an evaluation request to its integer argument. Upon receiving a value, it sends the write request, and propagates the acknowledgement. The net is in Figure 15h and its transition rules are trivial.

3.5.2 Initialization. The actual stateful behaviour is provided by **newref** $\in \mathcal{C}(!\mathbf{V} \multimap \mathbf{X}, \mathbf{X})$ and **newsem** $\in \mathcal{C}(!\mathbf{S} \multimap \mathbf{X}, \mathbf{X})$ with nets in Figures 15g and 15i, non-trivial transitions in Figure 14.

The Petri structure **newref** takes as an argument $!\mathbf{V} \multimap \mathbf{X}$, *i.e.* a "program of type $\mathbf{X}$" that may query a reference. Location 2 stores the current value. Upon initialization with $r_{\vdash}\mathsf{Q}^-$, **newref**: *(1)* initializes the reference, setting a token with data 0 in location 2; *(2)* in parallel, gives control to the argument via $\ell_{\vdash} r_{\multimap}\mathsf{Q}^+$. *Write* and *read* requests arrive respectively in locations 3 and 5. The neutral transitions w and r perform the memory update and reading. In the normal operation, there is ever at most one token in location 2, forcing reads and writes to be handled in some sequential order.

The net for **newsem** is essentially the same. In normal operation, location 2 contains exactly one token with data either $\mathtt{tt}$ or $\mathtt{ff}$, encoding if the semaphore is free. The transitions g and r grab and release the semaphore, and may only fire if the semaphore currently has the adequate state.

3.6 Wrapping up the Interpretation

With the above, we have completed the construction for:

Corollary 3.19. $\mathsf{PStruct}$ *is an* IPA*-structure.*

Following Section 2.3, we obtain the interpretation of a *type* A of IPA as a finite set $[\![A]\!] \subseteq \mathcal{M}$ of addresses, likewise for a context, and of a term $\Gamma \vdash M : A$ as $[\![M]\!] \in \mathsf{PStruct}(![\![\Gamma]\!], [\![A]\!])$ which we regard as the *concurrent games abstract machine* with M loaded. In particular, a closed program $\vdash M : \mathbb{X}$ is interpreted as a Petri structure $[\![M]\!]$ on set of addresses $\{r_{\vdash}\mathsf{Q}^-, r_{\vdash}\mathsf{A}^+\}$, *i.e.* a closed Petri structure up to the obvious renaming. It will follow from the later results of this paper that:

Theorem 3.20 (Adequacy). *For any closed program* $\vdash M : \mathbb{U}$*, we have* $M \Downarrow$ *iff* $[\![M]\!] \Downarrow$.

This is the standard way of stating that at ground type and with respect to may-convergence, the concurrent games abstract machine is faithful to standard operational semantics. But here adequacy shall follow from a much more powerful result, also giving a proper account of higher-order: we shall prove that the Petri structure $[\![M]\!]$ *unfolds* to the concurrent strategy interpreting M.

4 CONCURRENT STRATEGIES

Before we describe this unfolding, we must recall its target. So we include a brief reminder of the concurrent games model and its adequate semantics of IPA. The model and its interpretation of IPA is first described in [Castellan et al. 2019], with improvements and a detailed adequacy proof in [Castellan and Clairambault 2020]. With respect to this, the present model differs in two ways. Firstly, our moves follow Petri structures and use exponential signatures rather than natural numbers as copy indices. Secondly, unlike in [Castellan and Clairambault 2020], our games and strategies come without symmetry – we discuss this further in Appendix B.4.

4.1 Types and Contexts as Games

The first step is to introduce *games*: a game collects the moves one may encounter during an execution on a given type, and organizes them according to their causal dependencies and conflict.

In concurrent games, both games and strategies are certain *event structures*:

4.1.1 Event structures. Event structures are a well-known model from concurrency theory, presenting a system by specifying causal dependency and conflict between computational events:

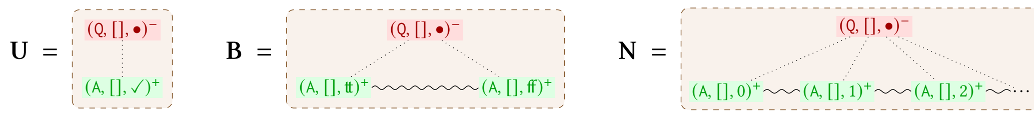


Fig. 16. Interpretation of IPA ground types

Definition 4.1. An **event structure (*es*)** is a triple $E = (|E|, \leq_E, \#_E)$, where $|E|$ is a (countable) set of **events**, $\leq_E$ is a partial order and $\#_E$ is an irreflexive symmetric binary relation, satisfying:

$$\begin{array}{rl} \textit{finite causes}: & \forall e \in |E|,\ [e]_E = \{e' \in |E| \mid e' \leq_E e\} \text{ is finite} \\ \textit{conflict inheritance}: & \forall e_1 \#_E e_2,\ \forall e_2 \leq_E e_2',\ e_1 \#_E e_2', \end{array}$$

and we call $\#_E$ and $\leq_E$ respectively **conflict** and **causal dependency**.

We write $\rightarrowtriangle_E$ for the immediate dependency relation induced by $\leq_E$, *i.e.* $e \rightarrowtriangle_E e'$ iff $e <_E e'$ with no other event strictly in between. An event structure E naturally comes with a notion of *state*, its (finite) **configurations**: those are finite sets $x \subseteq_f |E|$ which are down-closed for $\leq_E$ and compatible, in the sense that if $e, e' \in x$, then $\neg(e \#_E e')$. We write $\mathscr{C}(E)$ the set of configurations of E. If $x \in \mathscr{C}(E)$ and $e \notin x$ satisfies $x \cup \{e\} \in \mathscr{C}(E)$, we say that x **enables** e and write $x \vdash_E e$.

4.1.2 Games and arenas. *Games* are event structures composed of moves:

Definition 4.2. A **game** is an es $A = (|A|, \leq_A, \#_A)$ s.t. $|A| \subseteq \mathsf{Moves}$ and satisfying:

$$\textit{finite addresses:} \quad \text{the set } \mathsf{mult}(A) = \{\mathsf{m} \in \mathcal{M} \mid \exists(\mathsf{m}, \mathsf{s}, \mathsf{d}) \in |A|\} \text{ is finite.}$$

This is the usual notion of concurrent games, with the small difference that moves are taken in Moves. Consequently, polarity is inherited and no longer part of the data – we still display polarities in diagrams. The condition ensures that $\mathsf{mult}(A)$ is an object of PStruct for any A.

In Figure 16, we show the interpretation for ground types. In diagrams for games, the dotted lines indicate immediate causal dependency, flowing from top to bottom. Wiggly lines indicate *conflict* – in the diagram for **N** it is understood that all moves in the second row are in pairwise conflict, some being omitted to alleviate notations. As usual in game semantics for call-by-name languages, these diagrams formalize a protocol where the environment initiates computation by playing the negative move (a *question*), following which Player may respond a value (via an *answer*). All the moves in Figure 16 have a trivial address without injections: non-trivial addresses will arise only with constructors. Likewise, they all have trivial exponential stacks as the types do not contain a !. Finally, the data signature matches the return value whenever relevant, and is • otherwise.

It will be helpful in the sequel to have a tighter grasp on the shape of games arising from types:

Definition 4.3. An **arena** is a game $(A, \leq_A, \#_A)$ satisfying:

$$\begin{array}{rl} \textit{alternating:} & \text{if } a_1 \rightarrowtriangle_A a_2, \text{ then } \mathsf{pol}(a_1) \neq \mathsf{pol}(a_2), \\ \textit{forestial:} & \text{if } a_1 \leq_A a \text{ and } a_2 \leq_A a, \text{ then } a_1 \leq_A a_2 \text{ or } a_2 \leq_A a_1, \\ \textit{locally conflicting:} & \text{if } a_1, a_2 \in |A| \text{ are in minimal conflict, then they are both minimal,} \\ & \text{or have the same (necessarily unique) predecessor.} \\ \textit{negative:} & \text{if } a \in \min(A), \text{ then } \mathsf{pol}(a) = -, \end{array}$$

where $\min(A)$ comprises the minimal events of A. Finally, A is **well-opened** if $\min(A)$ is a singleton.

This used the notion of *minimal conflict*: in an event structure E, $e_1, e_2 \in |E|$ are in **minimal conflict** if $e_1 \#_E e_2$, while if $e_1' \leq_E e_1$ and $e_2' \leq_E e_2$ with at least one of these being strict, $\neg(e_1' \#_E e_2')$ – so the conflict is not inherited. In that case we write $e_1 \sim\!\sim_E e_2$; note that minimal conflict is what is represented throughout this paper in event structure diagrams.

Types and contexts shall be interpreted as arenas, via the constructions introduced next.

$$\begin{array}{rcl}
|A \multimap O| & = & \ell_{\multimap}(|A|) \uplus r_{\multimap}(|O|) \\
\leq_{A\multimap O} & = & \ell_{\multimap}(\leq_A) \uplus r_{\multimap}(\leq_O) \\
 & \uplus & r_{\multimap}(\min(O)) \times \ell_{\multimap}(|A|) \\
\#_{A\multimap O} & = & \ell_{\multimap}(\#_A) \uplus r_{\multimap}(\#_O)
\end{array}$$

(a) Arrow construction

$$\begin{array}{rcl}
|\,\&(A_x)_{x\in V}| & = & \biguplus_{x\in V} \imath^x_{\&}(|A_x|) \\
\leq_{\&(A_x)_{x\in V}} & = & \biguplus_{x\in V} \imath^x_{\&}(\leq_{A_x}) \\
\imath^x_{\&}(a) \# \imath^y_{\&}(a') & \Leftrightarrow & (x = y \wedge a \#_{A_x} a') \\
 & & \vee (x \neq y)
\end{array}$$

(b) Product construction

$$\begin{array}{rcl}
|!A| & = & \biguplus_{\mathsf{e}\in\mathcal{E}} \mathsf{e} :: |A| \\
\leq_{!A} & = & \biguplus_{\mathsf{e}\in\mathcal{E}} \mathsf{e} :: (\leq_A) \\
\#_{!A} & = & \biguplus_{\mathsf{e}\in\mathcal{E}} \mathsf{e} :: (\#_A)
\end{array}$$

(c) Bang construction

Fig. 17. Arena constructions

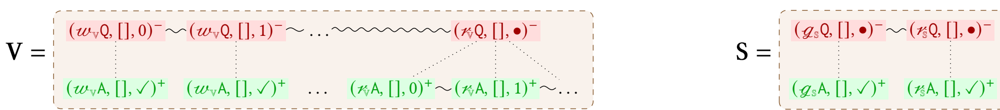


Fig. 18. Arenas for references and semaphores

4.1.3 Basic game constructions. We write either 1 or $\top$ for the empty arena.

If $(\mathsf{m}, \mathsf{s}, \mathsf{d}) \in \mathsf{Moves}$, we write $\ell_{\otimes}((\mathsf{m}, \mathsf{s}, \mathsf{d})) = (\ell_{\otimes}(\mathsf{m}), \mathsf{s}, \mathsf{d})$ and likewise for all injections appearing in the definition of addresses. This extends to sets of moves by direct image. Address injections also apply to binary relations on moves in the obvious way, *e.g.* $\ell_{\otimes}(R) = \{(\ell_{\otimes}(m), \ell_{\otimes}(m')) \mid m\,R\,m'\}$.

Definition 4.4. Consider A, B games. Then we set games $A \otimes B$ and $A \vdash B$ with components:

$$\begin{array}{rclcrcl}
|A \otimes B| & = & \ell_{\otimes}(|A|) \uplus r_{\otimes}(|B|) & \qquad & |A \vdash B| & = & \ell_{\vdash}(|A|) \uplus r_{\vdash}(|B|) \\
\leq_{A\otimes B} & = & \ell_{\otimes}(\leq_A) \uplus r_{\otimes}(\leq_B) & & \leq_{A\vdash B} & = & \ell_{\vdash}(\leq_A) \uplus r_{\vdash}(\leq_B) \\
\#_{A\otimes B} & = & \ell_{\otimes}(\#_A) \uplus r_{\otimes}(\#_B) & & \#_{A\vdash B} & = & \ell_{\vdash}(\#_A) \uplus r_{\vdash}(\#_B)
\end{array}$$

$A \otimes B$ is called the **tensor**, and $A \vdash B$ is called the **hom-game**.

If A, B are arenas, so is $A \otimes B$. In contrast, $A \vdash B$ is never an arena unless A is empty. Formally, both constructions are defined in the same way, but with a distinct injection – remember from Definition 3.2 that $\ell_{\vdash}$ inverts polarity, so that in $A \vdash B$ the polarity is inverted in A.

From the definition, *configurations* of $A \otimes B$ have a restricted shape: any $x \in \mathscr{C}(A \otimes B)$ decomposes uniquely as $x = \ell_{\otimes}(x_A) \uplus r_{\otimes}(x_B)$ with $x_A \in \mathscr{C}(A)$ and $x_B \in \mathscr{C}(B)$ – we write $x = x_A \otimes x_B$. Likewise, any configuration $x \in \mathscr{C}(A \vdash B)$ decomposes as $x = x_A \vdash x_B = \ell_{\vdash}(x_A) \uplus r_{\vdash}(x_B)$.

4.1.4 Further arena constructions. We give the remaining constructions required by Definition 2.1.

First a new notation: for $m = (\mathsf{m}, \mathsf{s}, \mathsf{d}) \in \mathsf{Moves}$ and $\mathsf{e} \in \mathcal{E}$, we write $\mathsf{e} :: m = (\mathsf{m}, \mathsf{e} :: \mathsf{s}, \mathsf{d})$, *i.e.* the exponential signature implicitly applies to the exponential stack of m. This is an injection, and accordingly we apply it to sets and relations as with the injections from addresses.

Definition 4.5. We define three other constructions on arenas:

linear arrow: for A, O arenas with O well-opened, we define $A \multimap O$ well-opened in Figure 17a.
product: for $(A_x)_{x\in V}$ a family of arenas with $V \subseteq \mathsf{Var}$, we define $\&_{x\in V} A_x$ in Figure 17b.
bang: for A an arena, we define $!A$ in Figure 17c.

It remains to give the arenas for references and semaphores, in Figure 18.

The arrow $A \multimap O$ enforces that an argument cannot be called before the function has been called. The bang $!A$ creates countably many independent copies of A, one for each exponential signature, for *thread indexing* (for that, [Castellan and Clairambault 2020] uses *integers*).

A more elaborate example of arena is in Figure 19. The representation is symbolic: from the exponentials in the construction, the full arena is infinite and comprises moves as in the diagram for all exponential signatures $\mathsf{e}, \mathsf{e}' \in \mathcal{E}$. However, we have *e.g.* $(\ell_{\multimap} r_{\multimap}\mathsf{Q}, [e], \bullet) \leq (\ell_{\multimap}\ell_{\multimap}\mathsf{Q}, [\mathsf{e}_1, \mathsf{e}_2], \bullet)$ only when $e = e_1$ – in an exponential stack, the first element corresponds to the outermost $!(-)$.

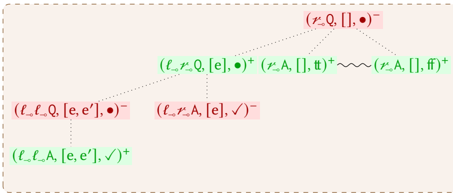


Fig. 19. Arena $!(!\mathbb{U} \multimap \mathbb{U}) \multimap \mathbb{B}$

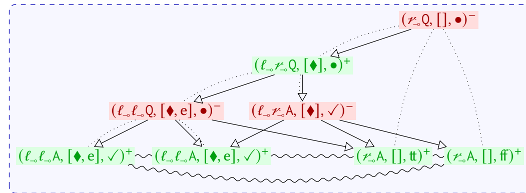


Fig. 20. Part of a concurrent strategy

4.2 The Category of Arenas and Strategies

4.2.1 Definition. In concurrent games, strategies are also event structures, labelled by the game:

Definition 4.6. A **prestrategy** $\sigma : A$ on game A comprises an es $(|\sigma|, \leq_\sigma, \#_\sigma)$ with $\partial : |\sigma| \to |A|$ a function called the **display map**, subject to the following conditions:

$$\begin{array}{rl} \textit{rule-abiding:} & \text{for all } x \in \mathscr{C}(\sigma),\ \partial(x) \in \mathscr{C}(A), \\ \textit{locally injective:} & \text{for all } s_1, s_2 \in x \in \mathscr{C}(\sigma), \text{ if } \partial(s_1) = \partial(s_2) \text{ then } s_1 = s_2. \end{array}$$

We say that σ is a **strategy** if it satisfies the further two conditions:

$$\begin{array}{rl} \textit{courteous:} & \text{for all } s_1 \rightarrowtriangle_\sigma s_2, \text{ if } \mathrm{pol}(s_1) = + \text{ or } \mathrm{pol}(s_2) = - \text{ then } \partial(s_1) \rightarrowtriangle_A \partial(s_2), \\ \textit{receptive:} & \text{for all } x \in \mathscr{C}(\sigma), \text{ for all } \partial(x) \vdash_A a^-, \\ & \text{there is a unique } x \vdash_\sigma s \in \mathscr{C}(\sigma) \text{ such that } \partial(s) = a, \end{array}$$

and additionally it is **negative** if for all $s \in |\sigma|$, if s is minimal then s is negative.

Rule-abiding and *locally injective* together amount to $\partial : \sigma \to A$ being a **map of event structure**. Events of σ inherit a polarity from $\mathrm{pol}_\sigma(s) = \mathrm{pol}(\partial(s))$ – a definition used implicitly from now on. Again, this definition is as in [Castellan and Clairambault 2020] without the component and conditions – unnecessary for this paper – pertaining to symmetry (see Appendix B.4 for more).

The event structure σ presents observable computational events along with their causal dependencies and conflicts. Events of σ are *not* moves of the game, but they do correspond to moves via the action of ∂. This permits an explicit representation of non-deterministic branching: several events of σ may correspond to the same move if they are conflicting and so belong to separate branches of the execution. The strategy keeps them separate, even if they cannot be distinguished.

We show in Figure 20 (part of) a concurrent strategy playing on the game in Figure 19. In such diagrams we represent the strategy and the explored part of the arena in a single picture. Nodes correspond to events of the strategy, drawn directly as their display through ∂ – this means that two conflicting nodes may have the same label, as happens in Figure 20. Arrows $\rightarrowtriangle$ correspond to the immediate causal dependency in σ, while dotted lines correspond as before to the causal dependency from the game. The diagram in Figure 20 is a part of the strategy for:

Example 4.7. We introduce the term $\vdash$ **strictness** : $(\mathbb{U} \to \mathbb{U}) \to \mathbb{B}$, defined as

$$\vdash \lambda f^{\mathbb{U}\to\mathbb{U}}.\, \mathsf{newref}\, x\, \mathsf{in}\, f\ (x := 1);\ \mathsf{not}\ (\mathsf{iszero}\ !x) : (\mathbb{U} \to \mathbb{U}) \to \mathbb{B}\,,$$

using encapsulated state to test if a function f calls its argument (run it here).

Figure 20 reads as follows: Opponent initiates computation with $(\mathscr{r}_\multimap \mathsf{Q}, [], \bullet)^-$. This lets Player call his argument with $(\ell_\multimap \mathscr{r}_\multimap \mathsf{Q}, [\blacklozenge], \bullet)^+$. This lets Opponent return the call with $(\ell_\multimap \mathscr{r}_\multimap \mathsf{A}, [\blacklozenge], \checkmark)^-$, or call its argument with $(\ell_\multimap \ell_\multimap \mathsf{Q}, [\blacklozenge, \mathsf{e}])^-$ for any $\mathsf{e} \in \mathcal{E}$ (in fact Opponent may call this argument arbitrarily many times with distinct exponential signatures, but Figure 20 only represents one call). But these events are not incompatible: though this behaviour is not realizable within IPA, our model lets Opponent call its argument *and* return *concurrently*! In that case both $x := 1$ and $!x$ are running, so there is a *race*. The conflicts in Figure 20 allow two outcomes, depending on who wins.

It remains to introduce the two conditions *courteous* and *receptive*: the former simply states that a strategy has no control over Opponent moves, and must acknowledge each Opponent move A uniquely. *Courtesy* entails that with respect to the immediate causal links of the game, a strategy can only add new dependencies from negative to positive moves. This formalizes the idea that strategies interact in an *asynchronous* environment, where *e.g.* causal links between positive moves may not be preserved by propagation of moves through buffers – or through a Petri structure.

4.2.2 *Isomorphic strategies.*

Strict equality of strategies is too strict to be useful; instead we use:

Definition 4.8. Consider $\sigma, \tau : A$ two (pre)strategies on game A.
An **isomorphism** $\varphi : \sigma \cong \tau$ is an invertible map of es $\varphi : \sigma \to \tau$ such that $\partial_\tau \circ \varphi = \partial_\sigma$.

We write $\sigma \cong \tau$ to mean that σ and τ are isomorphic, leaving the isomorphism unspecified. Clearly, this is an equivalence relation. It is a basic fact from the theory of event structures that isomorphisms between σ and τ are in one-to-one correspondence with order-isos $\varphi : \mathscr{C}(\sigma) \cong \mathscr{C}(\tau)$ such that $\partial_\tau \circ \varphi = \partial_\sigma$, *i.e.* any such order-iso is given by a unique isomorphism of strategies – this is convenient as it is often easier to construct a bijection between configurations rather than events.

For *strategies*, this can be simplified by ignoring trailing Opponent moves. A $x \in \mathscr{C}(\sigma)$ is **+-covered** if any $m \in x$ maximal in x is positive; we write $\mathscr{C}^+(\sigma)$ for the +-covered configurations of σ. The action of an iso φ on +-covered configurations suffices to completely describe it:

Lemma 4.9. *Consider $\sigma, \tau : A$ two strategies, and $\varphi : \mathscr{C}^+(\sigma) \cong \mathscr{C}^+(\tau)$ an order-iso s.t. $\partial_\tau \circ \varphi = \partial_\sigma$. Then, there is a unique $\hat{\varphi} : \sigma \cong \tau$ such that $\hat{\varphi}(x) = \varphi(x)$ for all $x \in \mathscr{C}^+(\sigma)$.*

This is an application in the trivial case without symmetry of Lemma 5.11 from [Castellan and Clairambault 2020], which will be very helpful in constructing isomorphisms in this paper.

4.2.3 *Composition.*

Working towards an IPA-structure, we must first define composition.

A strategy σ **from game** A **to game** B is defined as a strategy $\sigma : A \vdash B$. Let us fix for now games A, B and C and strategies $\sigma : A \vdash B$ and $\tau : B \vdash C$ that we wish to compose to $\tau \odot \sigma : A \vdash C$.

Lemma 4.9 puts the emphasis on +-covered configurations, so we investigate what should be the +-covered configurations of $\tau \odot \sigma$. It turns out that they correspond to pairs $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ whose synchronization of x^σ and x^τ through B is "sound", which we must now define.

We fix the convention that if $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$, we write $\partial_\sigma x^\sigma = x^\sigma_A \vdash x^\sigma_B$ and $\partial_\tau x^\tau = x^\tau_B \vdash x^\tau_C$ where $x^\sigma_A \in \mathscr{C}(A)$, $x^\sigma_B, x^\tau_B \in \mathscr{C}(B)$, and $x^\tau_C \in \mathscr{C}(C)$. We say $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$ are **matching** if $x^\sigma_B = x^\tau_B$, in which case it is unambiguous to write $\partial_\sigma(x^\sigma) = x_A \vdash x_B$ and $\partial_\tau(x^\tau) = x_B \vdash x_C$. So x^σ and x^τ reach the same state on B, but it remains to see if they do it with compatible causal constraints. For that, we set $x_A \parallel x_B \parallel x_C = \ell(x_A) \uplus m(x_B) \uplus r(x_C)$, and

$$
\begin{array}{llllll}
\partial^\ell_\sigma & : & x^\sigma & \to & x_A \parallel x_B \parallel x_C & \\
 & & m & \mapsto & \ell(a) & \text{if } \partial_\sigma(m) = \ell_\vdash(a), \\
 & & m & \mapsto & m(b) & \text{if } \partial_\sigma(m) = r_\vdash(b),
\end{array}
\qquad
\begin{array}{llllll}
\partial^r_\tau & : & x^\tau & \to & x_A \parallel x_B \parallel x_C & \\
 & & n & \mapsto & m(b) & \text{if } \partial_\tau(n) = \ell_\vdash(b), \\
 & & n & \mapsto & r(c) & \text{if } \partial_\tau(n) = r_\vdash(c).
\end{array}
$$

are variants of the display maps set to embed x^σ and x^τ in the common space $x_A \parallel x_B \parallel x_C$.

This lets us check the presence of deadlocks by importing all causal constraints to $x_A \parallel x_B \parallel x_C$:

Definition 4.10. Consider $x^\sigma \in \mathscr{C}(\sigma)$ and $x^\tau \in \mathscr{C}(\tau)$ matching configurations.
They are **causally compatible** if the relation $\lhd = \lhd_\sigma \uplus \lhd_\tau$ on $x_A \parallel x_B \parallel x_C$ set with:

$$
\begin{array}{rclcl}
\partial^\ell_\sigma(m) & \lhd_\sigma & \partial^\ell_\sigma(m') & \qquad & \text{for } m <_\sigma m' \\
\partial^r_\tau(n) & \lhd_\tau & \partial^r_\tau(n') & & \text{for } n <_\tau n'
\end{array}
$$

is acyclic. We also say that the pair x^σ, x^τ is **secured**.

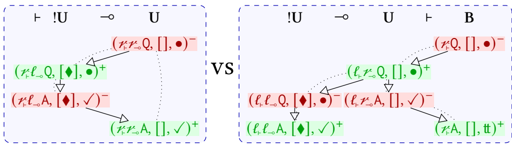


Fig. 21. Matching, secured configurations

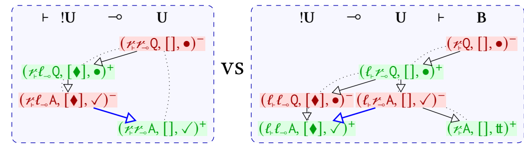


Fig. 22. Matching, non-secured configurations

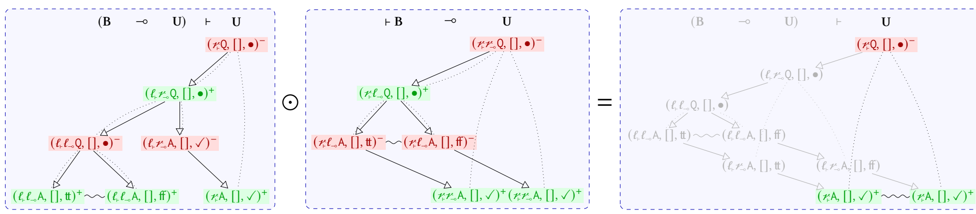


Fig. 23. Example of a composition

We show in Figures 21 and 22 examples of matching secured and non-secured pairs involved in computing the composition of the strategy of Figure 20 with $\lambda x^{\mathbb{U}}. x$. In Figure 22, a deadlock directly arises from opposite causal constraints (highlighted in blue). This entails that the only result arising from this composition will be $\mathtt{tt}$ from Figure 21, as expected since $\lambda x. x$ is strict.

Then $\tau \odot \sigma$ is the *unique* strategy with as +-covered configurations the causally compatible pairs:

Proposition 4.11. *Consider A, B, C games, and $\sigma : A \vdash B$ and $\tau : B \vdash C$ strategies. Then there is a strategy $\tau \odot \sigma : A \vdash C$, unique up to iso, s.t. there are order-isos:*

$$(- \odot -) \quad : \quad \{(x^\tau, x^\sigma) \in \mathscr{C}^+(\tau) \times \mathscr{C}^+(\sigma) \mid \textit{causally compatible}\} \quad \simeq \quad \mathscr{C}^+(\tau \odot \sigma)$$

such that for any $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ causally compatible, $\partial_{\tau\odot\sigma}(x^\tau \odot x^\sigma) = x^\sigma_A \vdash x^\tau_C$.

This is a simplification of Proposition 3.3.1 from [Castellan and Clairambault 2020].

Concretely, composition is performed by parallel interaction (via the synchronizing product of es used by Winskel to model CCS [Winskel 1982]); followed by *hiding* (Lemma 5.11) which keeps the visible events only. In this paper, the above characterization suffices for our purposes.

Figure 23 shows an example composition which, ignoring exponentials, corresponds to:

$$[\![f : \mathbb{B} \to \mathbb{U} \vdash f\ \mathsf{coin}]\!] \odot [\![\vdash \lambda x^{\mathbb{B}}. \mathsf{if}\ x\ \mathsf{skip}\ \mathsf{skip}]\!] : 1 \vdash \mathbb{U}\,.$$

Observe the resulting strategy has two distinct ways to converge, even though the two occurrences of $(\not\vdash \mathsf{A}, [], \checkmark)^+$ correspond to the same event of the left hand side strategy. Each event of the composition carries its whole causal history, including the exact synchronizations that lead to it.

4.3 An IPA-structure

The remaining structure required for the interpretation of IPA follows [Castellan and Clairambault 2020; Castellan et al. 2019]. We must omit the details for lack of space, but they can be found in Appendix B. Altogether, and with Theorem 4.40 of [Castellan and Clairambault 2020]:

Theorem 4.12. *The category* Strat *forms an* IPA*-structure. Moreover, for any closed program $\vdash M : \mathbb{U}$, $[\![M]\!]_{\mathsf{Strat}} \Downarrow$ iff $M \Downarrow$.*

In the statement above, we say that $\sigma : \mathbf{U}$ **converges**, written $\sigma \Downarrow$, iff it has a positive move.

Of course, Strat is most interesting not for programs of ground types, but for higher-order programs: there, it gives a complete account of the underlying *causal* structure behind observable computational actions – along with the non-deterministic branching information.

In the remainder of this paper we show that though $[\![M]\!]_{\mathsf{Strat}}$ is computed by definition *denotationally* by induction on terms following the corresponding operations on strategies, it may also be computed from the execution by *unfolding* the Petri structure interpreting M.

5 PETRI STRATEGIES AND UNFOLDING

Whereas concurrent strategies follow the rules of a game, unrestricted Petri structures are too liberal; they may play moves that are not accessible yet, or even meaningless (*e.g.* a boolean on a unit type). Before we unfold, we must capture, among Petri structures, those that respect the rules.

5.1 Strategic Petri Structures

5.1.1 Definition. Fix A a game, and $\boldsymbol{\sigma}$ a Petri structure on $\mathsf{mult}(A)$. Intuitively, $\boldsymbol{\sigma}$ is a *Petri strategy* on A if its *runs* follow the rules specified by A, *i.e.* form valid plays on A:

Definition 5.1. Consider A a game. A sequence $s_1 \dots s_n \in \mathsf{Moves}^*$ is a **play on** A if

$$\begin{array}{ll} \textit{valid:} & \text{for all } 1 \le i \le n,\ |s_1 \dots s_i| = \{s_1, \dots, s_i\} \in \mathscr{C}(A), \\ \textit{non-repetitive:} & \text{for all } 1 \le i \le j \le n, \text{ if } s_i = s_j \text{ then } i = j, \end{array}$$

we write $\mathsf{Plays}(A)$ for the set of plays on game A.

Recall that any *run* $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha$ on $\boldsymbol{\sigma}$ generates a sequence of moves $\mathsf{play}(\rho)$ obtained by collecting the labels of visible moves in ρ. Requiring that $\mathsf{play}(\rho) \in \mathsf{Plays}(A)$ for every run ρ is too brutal; we shall ask only that $\boldsymbol{\sigma}$ behaves as prescribed by the game as long as Opponent does. But we shall also need a *safety* condition, required for the forthcoming unfolding. For that, we set:

Definition 5.2. Consider $\mathbf{t} : \alpha \longmapsto_{\boldsymbol{\sigma}} \beta$ an itransition of $\boldsymbol{\sigma}$. We call $\mathsf{pre}(\mathbf{t}) = \alpha$ the **pre-condition** of $\mathbf{t}$, and $\mathsf{post}(\mathbf{t}) = \beta$ the **post-condition** of $\mathbf{t}$. The set $\mathsf{new}(\mathbf{t}) = \beta \setminus \alpha$ contains the tokils **produced** by $\mathbf{t}$; and $\mathsf{eat}(\mathbf{t}) = \alpha \setminus \beta$ those **consumed**. Those extend to ictransitions in the obvious way.

The distinction between $\mathsf{post}(\mathbf{t})$ and $\mathsf{new}(\mathbf{t})$ matters for **let**: its transition in Figure 14 requires $([], \mathrm{d})^{@3}$ to fire, but leaves it unchanged. So $([], \mathrm{d})^{@3}$ is both a *pre-condition* and a *post-condition*, but is not *produced* by the transition. Other than for **let**, pre- and post-conditions are always disjoint.

Our safety constraint uses the notion of *collection* of the tokils encountered in a run:

Definition 5.3. Consider $\rho = \mathbf{t}_1 \dots \mathbf{t}_n : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha_{i+1}$, a run of $\boldsymbol{\sigma}$, with $\mathbf{t}_i : \alpha_i \longrightarrow \alpha_{i+1}$.
The **collection** of ρ is $\mathsf{Coll}(\rho) = \bigcup_{1 \le i \le n+1} \alpha_i$. We say $\alpha \in \mathsf{cond}_{\boldsymbol{\sigma}}$ is **fresh in** ρ iff $\alpha \cap \mathsf{Coll}(\rho) = \emptyset$.

This lets us finally give the definition of *Petri strategies* on a game A:

Definition 5.4. We say that $\boldsymbol{\sigma}$ is a **Petri strategy on** A, written $\boldsymbol{\sigma} : A$, if for all $\rho : \emptyset \overset{s}{\longrightarrow\!\!\!\!\!\rightarrow}_{\boldsymbol{\sigma}} \alpha$:

$$\begin{array}{rl} \textit{valid:} & \text{if } s \in \mathsf{Plays}(A) \text{ and } \mathbf{t}^+ : \alpha \xrightarrow{a} \beta, \text{ then } sa \in \mathsf{Plays}(A), \\ \textit{receptive:} & \forall sa^- \in \mathsf{Plays}(A), \exists! \mathbf{t}^- \in \mathsf{IT}_{\boldsymbol{\sigma}} \text{ s.t. } \mathbf{t}^- : \emptyset \overset{a^-}{\longmapsto} \beta \text{ for some } \beta \cap \alpha = \emptyset, \\ \textit{strongly safe:} & \text{if } s \in \mathsf{Plays}(A) \text{ and } \mathbf{t} : \alpha \longrightarrow \beta, \text{ or } \mathbf{t} : \alpha \xrightarrow{a} \beta \text{ with } sa \in \mathsf{Plays}(A), \\ & \text{then } \mathsf{new}(\mathbf{t}) \text{ is fresh in } \rho, \end{array}$$

and additionally it is **negative** iff for all $\mathbf{t}^0 : \alpha \longmapsto \beta$ or $\mathbf{t}^+ : \alpha \overset{a}{\longmapsto} \emptyset$, we have $\alpha \neq \emptyset$.

Strong safety entails that in a given execution, a tokil can occur at most once: it cannot appear, then be consumed, only to reappear later. The main consequence of this is that tokils in a run of a Petri strategy have a canonical *causal* partial ordering, which will be central in the *unfolding* of Petri strategies to actual strategies – we direct to Section 5.2 for more on this.

Examples of Petri strategies abound in this paper so far – since it shall follow from this section that for all term $\Gamma \vdash M : A$, the Petri structure $[\![M]\!]$ is a Petri strategy on $![\![\Gamma]\!] \vdash [\![A]\!]$. In contrast, the Petri structure 1 of Section 3.1.1 does not satisfy $1 : \mathbf{U}$, as there is a clear failure of condition *valid*.

5.1.2 An IPA*-structure.* This lets us refine PStruct as follows:

Proposition 5.5. *There is an* IPA*-structure* PStrat*, with objects arenas (with the associated constructions), and morphisms from A to B the negative Petri strategies $\boldsymbol{\sigma} : A \vdash B$ up to iso.*

There is a strict F : PStrat $\to$ PStruct *sending an arena A to* $\mathsf{mult}(A)$ *and preserving morphisms.*

Proof. We must show that all primitives are Petri strategies, and that all operations preserve those. The main technical challenge is composition: there we show any $\rho : \emptyset \longrightarrow\!\!\!\!\!\!\rightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ projects to

$$\rho_{\boldsymbol{\sigma}} : \emptyset \longrightarrow\!\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}}\,, \qquad\qquad \rho_{\boldsymbol{\tau}} : \emptyset \longrightarrow\!\!\!\!\!\!\rightarrow_{\boldsymbol{\tau}} \alpha_{\boldsymbol{\tau}}$$

with $\alpha = \alpha_{\boldsymbol{\sigma}} +^{\odot} \alpha_{\boldsymbol{\tau}}$. If $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ and if additionally in $\mathsf{play}(\rho)$, the external Opponent respects the game, none of Opponent, $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ can be the first to break the rules; and we get by induction $\mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash C)$, $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$ and $\mathsf{play}(\rho_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C)$. This is the crux from which all conditions follow. See details in Appendix C. □

5.2 The Unfolding of a Petri Strategy

It is well-known that standard Petri nets unfold to event structures [Hayman and Winskel 2008b; Nielsen et al. 1981]. Usually, unfolding is an elaborate inductive unwinding process, yielding first an *occurrence net* which is then converted to an event structure. Here instead we leverage the presence of colours and in particular our *strong safety* condition to give a much more direct definition.

A strategy is a global object, putting together all possible executions with explicit causal and branching information. In this paper, we approach the construction of a strategy more locally: we first perform a causal analysis of individual runs, obtaining a structure called a *rigid family*. We then apply an already existing construction to get an event structure from this rigid family.

5.2.1 On rigid families. Rigid families seem to have remained in the concurrency theory folklore for a while and to have first appeared in published form in [Castellan et al. 2014a,b; Hayman 2014].

Consider $\mathscr{q} = (|\mathscr{q}|, \leq_{\mathscr{q}})$ and $\mathscr{p} = (|\mathscr{p}|, \leq_{\mathscr{p}})$ finite partial orders. We say that $\mathscr{q}$ is **rigidly included** in $\mathscr{p}$, written $\mathscr{q} \hookrightarrow \mathscr{p}$, if $|\mathscr{q}| \subseteq |\mathscr{p}|$, and if that inclusion: *(1)* preserves down-closed sets, *i.e.* $\mathscr{C}(\mathscr{q}) \subseteq \mathscr{C}(\mathscr{p})$; and *(2)* preserves causality, *i.e.* for all $e, e' \in |\mathscr{q}|$, if $e \leq_{\mathscr{q}} e'$ then $e \leq_{\mathscr{p}} e'$ as well.

Definition 5.6. A **rigid family** $\mathcal{F}$ is a non-empty set of finite partial orders which is:

$$\begin{array}{rl} \textit{rigid-closed:} & \text{if } \mathscr{p} \in \mathcal{F} \text{ and } \mathscr{q} \hookrightarrow \mathscr{p}, \text{then } \mathscr{q} \in \mathcal{F}, \\ \textit{binary-compatible:} & \text{if } X \subseteq_f \mathcal{F}, \text{then } X\!\uparrow \text{ iff for all } \mathscr{q}, \mathscr{p} \in X, \{\mathscr{q}, \mathscr{p}\}\!\uparrow. \end{array}$$

where $X\!\uparrow$ means that there is $\mathscr{r} \in \mathcal{F}$ such that for all $\mathscr{q} \in X$, $\mathscr{q} \hookrightarrow \mathscr{r}$.

We added *binary-compatible* to the definition, to match our event structures with binary conflict.

A rigid family $\mathcal{F}$ collects causal executions. Particularly interesting are the **primes** of $\mathcal{F}$, *i.e.* those $\mathscr{q} \in \mathcal{F}$ with a top element $\mathsf{top}(\mathscr{q}) = e$: those can be thought of as a single event e, with a causal history leading to e. Indeed, the reconstructed event structure will have the primes as events:

Proposition 5.7. *For $\mathcal{F}$ a rigid family, the data $\mathrm{Pr}(\mathcal{F}) = (|\mathrm{Pr}(\mathcal{F})|, \leq_{\mathrm{Pr}(\mathcal{F})}, \#_{\mathrm{Pr}(\mathcal{F})})$ defined by:*

$$\begin{array}{rcl} |\mathrm{Pr}(\mathcal{F})| & = & \{\mathscr{q} \in \mathcal{F} \mid \mathscr{q} \textit{ prime}\} \\ \mathscr{q} \leq_{\mathrm{Pr}(\mathcal{F})} \mathscr{p} & \Leftrightarrow & \mathscr{q} \subseteq \mathscr{p} \\ \neg(\mathscr{q} \;\#_{\mathrm{Pr}(\mathcal{F})}\; \mathscr{p}) & \Leftrightarrow & \{\mathscr{q}, \mathscr{p}\}\!\uparrow, \end{array}$$

is an event structure with $\chi_{\mathcal{F}} : \mathscr{C}(\mathrm{Pr}(\mathcal{F})) \cong \mathcal{F}$ an order-isomorphism.

The proof is routine – $\chi_{\mathcal{F}}$ takes $x \in \mathscr{C}(\mathrm{Pr}(\mathcal{F}))$ to its sup $\vee x \in \mathcal{F}$ obtained as the (necessarily) compatible union of all partial orders in x, while $\chi_{\mathcal{F}}^{-1}$ takes $\mathscr{q} \in \mathcal{F}$ to the set of primes below $\mathscr{q}$.

5.2.2 *Causal analysis of runs.* Now, fix a Petri strategy $\boldsymbol{\sigma} : A \vdash B$. From $\boldsymbol{\sigma}$ we shall extract a rigid family with one partial order for each run generating a valid play. Our first definition is:

Definition 5.8. Consider $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \gamma$ a **valid run on** $A \vdash B$, *i.e.* s.t. $\mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash B)$. Write IT_ρ the set of $\mathbf{t} \in \mathsf{IT}_{\boldsymbol{\sigma}}$ s.t. $\mathbf{t} \uplus \gamma$ appears in ρ for some γ. We set binary relations on IT_ρ:

$$\begin{array}{lcl} \mathbf{t} \lhd_A \mathbf{t}' & \Leftrightarrow & \mathbf{t} : \alpha \xmapsto{a} \beta,\ \mathbf{t}' : \alpha' \xmapsto{a'} \beta',\ a \rightarrowtriangle_A a', \\ \mathbf{t} \lhd_{\boldsymbol{\sigma}} \mathbf{t}' & \Leftrightarrow & \mathsf{new}(\mathbf{t}) \cap \mathsf{pre}(\mathbf{t}') \neq \emptyset\ \vee\ \mathsf{post}(\mathbf{t}) \cap \mathsf{eat}(\mathbf{t}') \neq \emptyset \end{array}$$

Then, we set $\lhd_\rho = \lhd_A \cup \lhd_{\boldsymbol{\sigma}}$.

It is obvious that $\lhd_\rho$ is acyclic, because if $\mathbf{t} \lhd_\rho \mathbf{t}'$, by definition $\mathbf{t}$ must appear before $\mathbf{t}'$ in ρ. Hence its reflexive transitive closure is a partial order, yielding a poset $\mathscr{T}(\rho) = (\mathsf{IT}_\rho, \leq_\rho)$.

The poset $\mathscr{T}(\rho)$ is generated by two kinds of basic causal dependencies. Firstly, $\lhd_A$ imports the "static" immediate causal links enforced by the game. Secondly, $\lhd_{\boldsymbol{\sigma}}$ carries the "dynamic" immediate causal links imposed by the token game itself: $\mathbf{t} \lhd_{\boldsymbol{\sigma}} \mathbf{t}'$ if $\mathbf{t}'$ expects some tokils *produced* by $\mathbf{t}$; or if $\mathbf{t}$ requires the presence of some tokils later *destroyed* by $\mathbf{t}'$.

We show that the set of all $\mathscr{T}(\rho)$ forms a rigid family – this rests on a few observations. First:

Lemma 5.9. *Consider ρ, ρ' two valid runs on $\boldsymbol{\sigma}$. If $\mathsf{IT}_\rho = \mathsf{IT}_{\rho'}$, then $\mathscr{T}(\rho) = \mathscr{T}(\rho')$.*

This is immediate since Definition 5.8 does not depend on the order of transitions in ρ.

This lemma draws interest to the sets of itransitions arising from valid runs. If $\mathsf{x} \subseteq_f \mathsf{IT}_{\boldsymbol{\sigma}}$, x is a **history** of $\boldsymbol{\sigma}$, written $\mathsf{x} \in \mathsf{Hist}(\boldsymbol{\sigma})$, if there is $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha$ valid s.t. $\mathsf{x} = \mathsf{IT}_\rho$ (note the change of fonts, to distinguish x from a configuration). By Lemma 5.9, Definition 5.8 yields a poset $\mathscr{T}(\mathsf{x}) = (\mathsf{x}, \leq_{\mathsf{x}})$ for all $\mathsf{x} \in \mathsf{Hist}(\boldsymbol{\sigma})$ – as $\leq_{\mathsf{x}}$ is fully induced by x, we also refer to $\mathscr{T}(\mathsf{x})$ as the **history**.

Proposition 5.10. *The set comprising all $\mathscr{T}(\mathsf{x})$ for $\mathsf{x} \in \mathsf{Hist}(\boldsymbol{\sigma})$, is a rigid family written $\mathscr{T}(\boldsymbol{\sigma})$. Moreover, $\mathscr{T}(\boldsymbol{\sigma})$ (ordered by rigid inclusion) is order-isomorphic to $\mathsf{Hist}(\boldsymbol{\sigma})$ (ordered by inclusion).*

See Appendix D.1. Propositions 5.7 and 5.10 give $\mathrm{Pr}(\mathscr{T}(\boldsymbol{\sigma}))$, with $\mathscr{C}(\mathrm{Pr}(\mathscr{T}(\boldsymbol{\sigma}))) \cong \mathsf{Hist}(\boldsymbol{\sigma})$.

5.2.3 *Construction of a strategy.* The above gives us an event structure $\mathrm{Pr}(\mathscr{T}(\boldsymbol{\sigma}))$ but not quite the right one for a strategy: its events correspond to *all* itransitions, but that includes neutral itransitions not accounted for by strategies. To remove them, we use *projection*:

Lemma 5.11. *Consider E an event structure, and $V \subseteq |E|$ any set of events.*
*Then the **projection** $E \downarrow V$ is an event structure, with components:* events, $|E \downarrow V| = V$; causality, $e \leq_{E\downarrow V} e'$ *iff* $e \leq_E e'$; conflict, $e \#_{E\downarrow V} e'$ *iff* $e \#_E e'$. *Moreover, we have an order-isomorphism*

$$\mathscr{C}(E \downarrow V) \cong \mathscr{C}^V(E)$$

*where $\mathscr{C}^V(E)$ is the **maximally visible** configurations,* i.e. *the $x \in \mathscr{C}(E)$ with maximal events in V.*

Proof. Direct; the order-iso sends $x \in \mathscr{C}(E \downarrow V)$ to $[x]_E = \{e' \in |E| \mid \exists e \in x,\ e' \leq_E e\}$. □

Say $\mathsf{x} \in \mathscr{T}(\boldsymbol{\sigma})$ is **visible** if its top element is a visible itransition written $\mathbf{t} : \alpha \xmapsto{m} \beta$ – in that case we write $\mathbf{t} = \mathsf{top}(\mathsf{x})$ (and recall $\partial_{\boldsymbol{\sigma}}(\mathbf{t}) = m$). We write $\mathscr{V}_{\boldsymbol{\sigma}}$ the set of visible $\mathsf{x} \in \mathscr{T}(\boldsymbol{\sigma})$.

We are finally in position to unfold a Petri strategy $\boldsymbol{\sigma} : A \vdash B$:

Proposition 5.12. *The event structure $\mathscr{U}(\boldsymbol{\sigma}) = \mathrm{Pr}(\mathscr{T}(\boldsymbol{\sigma})) \downarrow \mathscr{V}_{\boldsymbol{\sigma}}$, equipped with the display map*

$$\begin{array}{rcrcl} \partial_{\mathscr{U}(\boldsymbol{\sigma})} & : & |\mathscr{U}(\boldsymbol{\sigma})| & \to & |A \vdash B| \\ & & \mathscr{q} & \mapsto & \partial_{\boldsymbol{\sigma}}(\mathsf{top}(\mathscr{q})) \end{array}$$

is a strategy in the sense of Definition 4.6. Moreover, $\mathscr{U}(\boldsymbol{\sigma})$ is negative if $\boldsymbol{\sigma}$ is.

Proof. From the order-isos $\mathscr{C}(\mathscr{U}(\boldsymbol{\sigma})) \cong \mathscr{C}^{\mathscr{V}_\sigma}(\mathrm{Pr}(\mathscr{T}(\boldsymbol{\sigma})))$ and $\mathscr{C}(\mathrm{Pr}(\mathscr{T}(\boldsymbol{\sigma}))) \cong \mathscr{T}(\boldsymbol{\sigma})$, we get

$$K_{\boldsymbol{\sigma}} : \mathscr{C}(\mathscr{U}(\boldsymbol{\sigma})) \cong \mathscr{T}^V(\boldsymbol{\sigma}) \tag{4}$$

an order-iso with $\mathscr{T}^V(\boldsymbol{\sigma})$ the set of histories whose maximal elements are visible. It sends $x \in \mathscr{C}(\mathscr{U}(\boldsymbol{\sigma}))$ to $\vee x \in \mathscr{T}^V(\boldsymbol{\sigma})$, and its inverse sends $\mathscr{q} \in \mathscr{T}^V(\boldsymbol{\sigma})$ to $\{\mathscr{p} \hookrightarrow \mathscr{q} \mid \mathscr{p} \in \mathscr{V}_{\boldsymbol{\sigma}} \text{ prime}\}$.

Rule-abiding. For $x \in \mathscr{C}(\mathscr{U}(\boldsymbol{\sigma}))$, $\partial_{\mathscr{U}(\boldsymbol{\sigma})}(x)$ is easily characterized as the set of labels of visible itransitions in $K_{\boldsymbol{\sigma}}(x) \in \mathscr{T}(\boldsymbol{\sigma})$ – a configuration of $A \vdash B$, since histories originate in valid runs.

Locally injective. Likewise, through $K_{\boldsymbol{\sigma}}$ this boils down to the fact that no two visible itransitions of $K_{\boldsymbol{\sigma}}(x)$ may have the same label, as that would contradict condition *non-repetitive* of plays.

Courteous, receptive, negative. See Appendix D.1 for details. □

Finally, to prove its functoriality, the following straightforward lemma will be helpful – where $\mathscr{T}^+(\boldsymbol{\sigma})$, the **+-covered histories** of $\boldsymbol{\sigma}$, are those histories whose maximal transitions are positive.

Lemma 5.13. *The order-isomorphism (4) of Lemma 5.12 specializes to $K^+_{\boldsymbol{\sigma}} : \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma})) \cong \mathscr{T}^+(\boldsymbol{\sigma})$ s.t. for all $x \in \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma}))$, we have $\partial_{\mathscr{U}(\boldsymbol{\sigma})}(x) = \partial_{\boldsymbol{\sigma}}(K^+_{\boldsymbol{\sigma}}(x))$ the labels of visible transitions in $K^+_{\boldsymbol{\sigma}}(x)$.*

5.3 Unfolding as an IPA-functor

We show unfolding preserves the interpretation of IPA – the main challenge is composition.

5.3.1 Preservation of composition. Recall from Proposition 4.11 that given strategies $\sigma : A \vdash B$ and $\tau : B \vdash C$, +-covered configurations of $\tau \odot \sigma$ exactly correspond to *causally compatible pairs* of $x^\sigma \in \mathscr{C}^+(\sigma)$ and $x^\tau \in \mathscr{C}^+(\tau)$ such that x^σ and x^τ match on B (through their respective display maps), and such that the induced synchronization is *secured*, *i.e.* deadlock-free. The crux of preservation of composition is a corresponding statement for *histories* of the composition of Petri strategies.

For Petri strategies $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$, we repeat the constructions of Section 4.2.3. Given $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}^+(\boldsymbol{\sigma})$, write $\partial_{\boldsymbol{\sigma}}(\mathsf{x}^{\boldsymbol{\sigma}}) = \mathsf{x}^{\boldsymbol{\sigma}}_A \vdash \mathsf{x}^{\boldsymbol{\sigma}}_B$. Histories $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}^+(\boldsymbol{\sigma})$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}^+(\boldsymbol{\tau})$ are **matching** if $\mathsf{x}^{\boldsymbol{\sigma}}_B = \mathsf{x}^{\boldsymbol{\tau}}_B$; so it is unambiguous to write $\partial_{\boldsymbol{\sigma}}(\mathsf{x}^{\boldsymbol{\sigma}}) = x_A \vdash x_B$ and $\partial_{\boldsymbol{\tau}}(\mathsf{x}^{\boldsymbol{\tau}}) = x_B \vdash x_C$. We set

$$\begin{array}{lllll} \partial^\ell_{\boldsymbol{\sigma}} : & \mathsf{x}^{\boldsymbol{\sigma}} & \rightharpoonup & x_A \parallel x_B \parallel x_C & \\ & \mathbf{t} & \mapsto & \ell(a) & \text{if } \partial_{\boldsymbol{\sigma}}(\mathbf{t}) = \ell_{+}(a), \\ & \mathbf{t} & \mapsto & \mathscr{m}(b) & \text{if } \partial_{\boldsymbol{\sigma}}(\mathbf{t}) = \mathscr{r}_{+}(b), \end{array} \qquad \begin{array}{lllll} \partial^{\mathscr{r}}_{\boldsymbol{\tau}} : & \mathsf{x}^{\boldsymbol{\tau}} & \rightharpoonup & x_A \parallel x_B \parallel x_C & \\ & \mathbf{t} & \mapsto & \mathscr{m}(b) & \text{if } \partial_{\boldsymbol{\tau}}(\mathbf{t}) = \ell_{+}(b), \\ & \mathbf{t} & \mapsto & \mathscr{r}(c) & \text{if } \partial_{\boldsymbol{\tau}}(\mathbf{t}) = \mathscr{r}_{+}(c). \end{array}$$

and undefined otherwise (*i.e.* for neutral itransitions). We define:

Definition 5.14. Consider $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}^+(\boldsymbol{\sigma})$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}^+(\boldsymbol{\tau})$ matching.
They are **causally compatible** if the relation $\lhd = \lhd_{\boldsymbol{\sigma}} \uplus \lhd_{\boldsymbol{\tau}}$ on $x_A \parallel x_B \parallel x_C$ set with:

$$\begin{array}{lllll} \partial^\ell_{\boldsymbol{\sigma}}(\mathbf{t}) & \lhd_{\boldsymbol{\sigma}} & \partial^\ell_{\boldsymbol{\sigma}}(\mathbf{t}') & \qquad & \text{for } \mathbf{t} <_{\mathsf{x}^{\boldsymbol{\sigma}}} \mathbf{t}' \\ \partial^{\mathscr{r}}_{\boldsymbol{\tau}}(\mathbf{t}) & \lhd_{\boldsymbol{\tau}} & \partial^{\mathscr{r}}_{\boldsymbol{\tau}}(\mathbf{t}') & & \text{for } \mathbf{t} <_{\mathsf{x}^{\boldsymbol{\tau}}} \mathbf{t}' \end{array}$$

is acyclic. We also say that the pair $\mathsf{x}^\sigma, \mathsf{x}^\tau$ is **secured**.

This is by design almost a copy of Definition 4.10. As for strategies, we have (see Appendix D.2):

Proposition 5.15. *Consider $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ Petri strategies. Then, there is an order-iso:*

$$(- \odot -) : \{(\mathsf{x}^{\boldsymbol{\tau}}, \mathsf{x}^{\boldsymbol{\sigma}}) \in \mathscr{T}^+(\boldsymbol{\tau}) \times \mathscr{T}^+(\boldsymbol{\sigma}) \mid \textit{causally compatible}\} \cong \mathscr{T}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$$

such that for $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}^+(\boldsymbol{\sigma}), \mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}^+(\boldsymbol{\tau})$ causally compatible, $\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}) = \mathsf{x}^{\boldsymbol{\sigma}}_A \vdash \mathsf{x}^{\boldsymbol{\tau}}_C$.

Proposition 5.16. *For $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ Petri strategies, $\mathscr{U}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \cong \mathscr{U}(\boldsymbol{\tau}) \odot \mathscr{U}(\boldsymbol{\sigma})$.*

Proof. We may deduce preservation of composition simply by manipulating isos:

$$
\begin{array}{rcl}
\mathscr{C}^{+}(\mathscr{U}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})) & \cong & \mathscr{T}^{+}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \\
& \cong & \{(\mathsf{x}^{\boldsymbol{\sigma}}, \mathsf{x}^{\boldsymbol{\tau}}) \in \mathscr{T}^{+}(\boldsymbol{\sigma}) \times \mathscr{T}^{+}(\boldsymbol{\tau}) \mid \text{causally compatible}\} \\
& \cong & \{(x^{\mathscr{U}(\boldsymbol{\sigma})}, x^{\mathscr{U}(\boldsymbol{\tau})}) \in \mathscr{C}^{+}(\mathscr{U}(\boldsymbol{\sigma})) \times \mathscr{C}^{+}(\mathscr{U}(\boldsymbol{\tau})) \mid \text{causally compatible}\} \\
& \cong & \mathscr{C}^{+}(\mathscr{U}(\boldsymbol{\tau}) \odot \mathscr{U}(\boldsymbol{\sigma}))
\end{array}
$$

via Lemma 5.13, Proposition 5.15, and Lemma 5.13 – preservation of causal compatibility follows directly from the order-isomorphism. All these steps preserve displays to the game. By Proposition 4.11, it follows that $\mathscr{U}(\boldsymbol{\tau} \odot \boldsymbol{\sigma}) \cong \mathscr{U}(\boldsymbol{\tau}) \odot \mathscr{U}(\boldsymbol{\sigma})$ as required. □

5.3.2 Wrapping up. For other operations (tensor, currying, promotion), their preservation by the unfolding is much more direct; primitives are shown to unfold to the adequate strategy via a characterization of their +-covered histories – details appear in Appendices D.3 and D.4. Altogether:

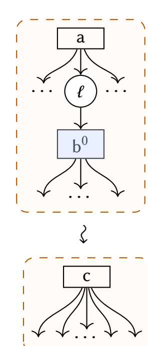
Theorem 5.17. *Unfolding yields an* IPA*-functor* $\mathscr{U}$: PStrat → Strat.

Following [Castellan and Clairambault 2020; Castellan et al. 2019], a formal description of the causal behaviour of any $\Gamma \vdash M : A$ as an event structure can be obtained by its interpretation $[\![M]\!]_{\mathsf{Strat}} \in \mathsf{Strat}(![\![\Gamma]\!], [\![A]\!])$. Theorem 5.17 shows that it can also be obtained more directly by purely operational means, through the unfolding of the abstract machine initialized on M, *i.e.* $[\![M]\!]_{\mathsf{PStrat}}$.

Finally, Theorem 3.20 follows from Theorems 4.12 and 5.17, Proposition 5.5, and Lemma 2.3.

6 IMPLEMENTATION

We implemented our translation from IPA to Petri strategies in an interactive web application available here with some documentation here. The interface lets the user enter a IPA program (or choose from a list of examples) and then displays the Petri strategy. The user can then simulate runs by firing available transitions and see the tokens flow through the net. The implementation only displays the data component of a token, but the exponential stack can be obtained by hovering the mouse above the token. Likewise, transition rules are displayed by hovering the mouse above transitions, though written in an undocumented stack language.



Our translation, following the categorical semantics, tends to generate large Petri strategies. To keep the nets at a reasonable size, we have implemented several optimisations. The first optimisation is to eliminate locations and transitions that are unreachable from a negative transition, or that never reach a positive transition. Such "dead code" can occur during the composition. Moreover, when we have several transitions occurring in a simple sequence, we combine them into one by composing their transition functions and eliminate the intermediate transitions and locations. One example of such optimisation is represented on the right, where we merge a and b^0 and remove the location ℓ. The new transition c has $\mathsf{pre}(\mathsf{c}) = \mathsf{pre}(\mathsf{a})$ and $\mathsf{post}(\mathsf{c}) = \mathsf{post}(\mathsf{a}) \setminus \{\ell\} \cup \mathsf{post}(\mathsf{c})$ and transition function: $\delta\langle\mathsf{c}\rangle(\alpha) = \delta\langle\mathsf{a}\rangle(\alpha) \setminus \{\mathsf{t}\} \cup \delta\langle\mathsf{b}\rangle(\{\mathsf{t}\})$, where t denotes the tokil at ℓ in $\delta\langle\mathsf{a}\rangle(\alpha)$.

There are minor inconsistencies between examples given in the paper and the implementation: firstly, optimization choices are not unique. In the paper, they are chosen so as to make transition functions more intuitive, which sometimes leads to different choices (compare *e.g.* Figure 6 with this). Secondly, in some examples we simplify the exponential signatures used by a strategy rather than using directly those arising from the interpretation (compare *e.g.* Figure 20 with this).

7 CONCLUSION

Though this is a theoretical contribution, we believe it is worth exploring applications to the compilation and analysis of higher-order, concurrent, effectful programming languages.

Firstly, our translation cleanly confines the infinity to tokens (*data* and *exponential* signatures). Forgetting colours, we immediately obtain a finitary over-approximation of the behaviour of programs, a Petri net in the usual sense, that may be used to prove *e.g.* safety properties. This may be refined by, rather than getting entirely rid of the infinite behaviour, handling it symbolically or over-approximating via abstract interpretation – perhaps offering a new truly concurrent basis for the static analysis of higher-order concurrent programs.

Secondly, GoI has been proposed in the past as an approach for the compilation of functional languages, suggested in particular by connections with Lamping's *optimal reduction* for the λ-calculus [Asperti et al. 1994; Gonthier et al. 1992; Lamping 1990]. Perhaps this work suggests a way to reap the rewards of this connection beyond purely functional languages.

Thirdly, another merit in using Petri strategies as an intermediate language is that the unfolding theorem (Theorem 5.17) provides clean and formal semantics which may serve as guide or semantic justification for optimizations, including those that rely on causal information.

IPA is of course hardly a realistic programming language, but we do not foresee any fundamental obstacle in generalizing this approach. In earlier work on GoI, handling call-by-value has sometimes proved a challenge, *e.g.* requiring repeating computation [Dal Lago et al. 2015], which breaks the expected cost model – this motivated the recent *dynamic GoI* of [Muroya and Ghica 2019]. In contrast, Petri strategies support a controlled evaluation order using memoization without repetition, as already illustrated in the *let* binding included in our version of IPA.

ACKNOWLEDGMENTS

We are grateful to Olivier Laurent for enlightening discussions on the Geometry of Interaction.

Work supported by the ANR project DyVerSe (ANR-19-CE48-0010-01); and by the Labex MiLyon (ANR-10-LABX-0070) of Université de Lyon, within the program “Investissements d'Avenir” (ANR-11-IDEX-0007), operated by the French National Research Agency (ANR).

REFERENCES

Samson Abramsky, Kohei Honda, and Guy McCusker. 1998. A Fully Abstract Game Semantics for General References. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science, Indianapolis, Indiana, USA, June 21-24, 1998*. IEEE Computer Society, 334–344. https://doi.org/10.1109/LICS.1998.705669

Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. 2000. Full Abstraction for PCF. *Inf. Comput.* 163, 2 (2000), 409–470. https://doi.org/10.1006/inco.2000.2930

Samson Abramsky and Guy McCusker. 1996. Linearity, Sharing and State: a fully abstract game semantics for Idealized Algol with active expressions. *Electron. Notes Theor. Comput. Sci.* 3 (1996), 2–14. https://doi.org/10.1016/S1571-0661(05)80398-6

Andrea Asperti, Vincent Danos, Cosimo Laneve, and Laurent Regnier. 1994. Paths in the lambda-calculus. In *LICS*. IEEE Computer Society, 426–436.

John C. Baez and Jade Master. 2020. Open Petri nets. *Math. Struct. Comput. Sci.* 30, 3 (2020), 314–341.

Patrick Baillot. 1999. *Approches dynamiques en sémantique de la logique linéaire: jeux et géométrie de l'interaction.* Ph.D. Dissertation. Aix-Marseille 2.

Simon Castellan and Pierre Clairambault. 2020. Disentangling Parallelism and Interference in Game Semantics. (2020). https://arxiv.org/abs/2103.15453 Submitted.

Simon Castellan, Pierre Clairambault, Silvain Rideau, and Glynn Winskel. 2017. Games and Strategies as Event Structures. *Logical Methods in Computer Science* 13, 3 (2017). https://doi.org/10.23638/LMCS-13(3:35)2017

Simon Castellan, Pierre Clairambault, and Glynn Winskel. 2014a. Symmetry in concurrent games. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, Thomas A. Henzinger and Dale Miller (Eds.). ACM, 28:1–28:10. https://doi.org/10.1145/2603088.2603141

Simon Castellan, Pierre Clairambault, and Glynn Winskel. 2019. Thin Games with Symmetry and Concurrent Hyland-Ong Games. *Log. Methods Comput. Sci.* 15, 1 (2019). https://doi.org/10.23638/LMCS-15(1:18)2019

Simon Castellan, Jonathan Hayman, Marc Lasson, and Glynn Winskel. 2014b. Strategies as Concurrent Processes. In *Proceedings of the 30th Conference on the Mathematical Foundations of Programming Semantics, MFPS 2014, Ithaca, NY, USA, June 12-15, 2014 (Electronic Notes in Theoretical Computer Science)*, Bart Jacobs, Alexandra Silva, and Sam Staton (Eds.), Vol. 308. Elsevier, 87–107. https://doi.org/10.1016/j.entcs.2014.10.006

Thomas Chatain and Eric Fabre. 2010. Factorization Properties of Symbolic Unfoldings of Colored Petri Nets. In *Applications and Theory of Petri Nets, 31st International Conference, PETRI NETS 2010, Braga, Portugal, June 21-25, 2010. Proceedings (Lecture Notes in Computer Science)*, Johan Lilius and Wojciech Penczek (Eds.), Vol. 6128. Springer, 165–184. https://doi.org/10.1007/978-3-642-13675-7_11

Pierre Clairambault and Andrzej S. Murawski. 2019. On the Expressivity of Linear Recursion Schemes. In *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany (LIPIcs)*, Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen (Eds.), Vol. 138. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 50:1–50:14. https://doi.org/10.4230/LIPIcs.MFCS.2019.50

Ugo Dal Lago, Claudia Faggian, Ichiro Hasuo, and Akira Yoshimizu. 2014. The geometry of synchronization. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, Thomas A. Henzinger and Dale Miller (Eds.). ACM, 35:1–35:10. https://doi.org/10.1145/2603088.2603154

Ugo Dal Lago, Claudia Faggian, Benoît Valiron, and Akira Yoshimizu. 2015. Parallelism and Synchronization in an Infinitary Context. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*. IEEE Computer Society, 559–572. https://doi.org/10.1109/LICS.2015.58

Ugo Dal Lago, Claudia Faggian, Benoît Valiron, and Akira Yoshimizu. 2017. The geometry of parallelism: classical, probabilistic, and quantum effects. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, Giuseppe Castagna and Andrew D. Gordon (Eds.). ACM, 833–845. http://dl.acm.org/citation.cfm?id=3009859

Vincent Danos, Hugo Herbelin, and Laurent Regnier. 1996. Game Semantics & Abstract Machines. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*. IEEE Computer Society, 394–405. https://doi.org/10.1109/LICS.1996.561456

Vincent Danos and Laurent Regnier. 1996. Reversible, Irreversible and Optimal Lambda-machines. *Electron. Notes Theor. Comput. Sci.* 3 (1996), 40–60.

Dan R. Ghica. 2007. Geometry of synthesis: a structured approach to VLSI design. In *Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2007, Nice, France, January 17-19, 2007*, Martin Hofmann and Matthias Felleisen (Eds.). ACM, 363–375. https://doi.org/10.1145/1190216.1190269

Dan R. Ghica and Andrzej S. Murawski. 2008. Angelic semantics of fine-grained concurrency. *Ann. Pure Appl. Logic* 151, 2-3 (2008), 89–114.

Dan R. Ghica and Alex I. Smith. 2010. Geometry of Synthesis II: From Games to Delay-Insensitive Circuits. In *MFPS (Electronic Notes in Theoretical Computer Science)*, Vol. 265. Elsevier, 301–324.

Dan R. Ghica and Nikos Tzevelekos. 2012. A System-Level Game Semantics. In *Proceedings of the 28th Conference on the Mathematical Foundations of Programming Semantics, MFPS 2012, Bath, UK, June 6-9, 2012 (Electronic Notes in Theoretical Computer Science)*, Ulrich Berger and Michael W. Mislove (Eds.), Vol. 286. Elsevier, 191–211. https://doi.org/10.1016/j.entcs.2012.08.013

Jean-Yves Girard. 1989. Geometry of interaction 1: Interpretation of System F. In *Studies in Logic and the Foundations of Mathematics*. Vol. 127. Elsevier, 221–260.

Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. 1992. The Geometry of Optimal Lambda Reduction. In *POPL*. ACM Press, 15–26.

Ichiro Hasuo and Naohiko Hoshino. 2017. Semantics of higher-order quantum computation via geometry of interaction. *Ann. Pure Appl. Log.* 168, 2 (2017), 404–469. https://doi.org/10.1016/j.apal.2016.10.010

Jonathan Hayman. 2014. Interaction and Causality in Digital Signature Exchange Protocols. In *Trustworthy Global Computing - 9th International Symposium, TGC 2014, Rome, Italy, September 5-6, 2014. Revised Selected Papers (Lecture Notes in Computer Science)*, Matteo Maffei and Emilio Tuosto (Eds.), Vol. 8902. Springer, 128–143. https://doi.org/10.1007/978-3-662-45917-1_9

Jonathan Hayman and Glynn Winskel. 2008a. Independence and Concurrent Separation Logic. *Log. Methods Comput. Sci.* 4, 1 (2008). https://doi.org/10.2168/LMCS-4(1:6)2008

Jonathan Hayman and Glynn Winskel. 2008b. The unfolding of general Petri nets. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2008, December 9-11, 2008, Bangalore, India (LIPIcs)*, Ramesh Hariharan, Madhavan Mukund, and V. Vinay (Eds.), Vol. 2. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 223–234. https://doi.org/10.4230/LIPIcs.FSTTCS.2008.1755

Naohiko Hoshino, Koko Muroya, and Ichiro Hasuo. 2014. Memoryful geometry of interaction: from coalgebraic components to algebraic effects. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, Thomas A. Henzinger and Dale Miller (Eds.). ACM, 52:1–52:10. https://doi.org/10.1145/2603088.2603124

J. M. E. Hyland and C.-H. Luke Ong. 2000. On Full Abstraction for PCF: I, II, and III. *Inf. Comput.* 163, 2 (2000), 285–408. https://doi.org/10.1006/inco.2000.2917

Guilhem Jaber. 2015. Operational Nominal Game Semantics. In *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings (Lecture Notes in Computer Science)*, Andrew M. Pitts (Ed.), Vol. 9034. Springer, 264–278. https://doi.org/10.1007/978-3-662-46678-0_17

Jérémie Koenig and Zhong Shao. 2020. Refinement-Based Game Semantics for Certified Abstraction Layers. In *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller (Eds.). ACM, 633–647. https://doi.org/10.1145/3373718.3394799

James Laird. 1997. Full Abstraction for Functional Languages with Control. In *Proceedings, 12th Annual IEEE Symposium on Logic in Computer Science, Warsaw, Poland, June 29 - July 2, 1997*. IEEE Computer Society, 58–67. https://doi.org/10.1109/LICS.1997.614931

James Laird. 2001. A Fully Abstract Game Semantics of Local Exceptions. In *16th Annual IEEE Symposium on Logic in Computer Science, Boston, Massachusetts, USA, June 16-19, 2001, Proceedings*. IEEE Computer Society, 105–114. https://doi.org/10.1109/LICS.2001.932487

John Lamping. 1990. An Algorithm for Optimal Lambda Calculus Reduction. In *POPL*. ACM Press, 16–30.

Olivier Laurent. 2001. A Token Machine for Full Geometry of Interaction. In *Typed Lambda Calculi and Applications, 5th International Conference, TLCA 2001, Krakow, Poland, May 2-5, 2001, Proceedings (Lecture Notes in Computer Science)*, Samson Abramsky (Ed.), Vol. 2044. Springer, 283–297. https://doi.org/10.1007/3-540-45413-6_23

Paul Blain Levy and Sam Staton. 2014. Transition systems over games. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, Thomas A. Henzinger and Dale Miller (Eds.). ACM, 64:1–64:10. https://doi.org/10.1145/2603088.2603150

Ian Mackie. 1995. The Geometry of Interaction Machine. In *Conference Record of POPL'95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Francisco, California, USA, January 23-25, 1995*, Ron K. Cytron and Peter Lee (Eds.). ACM Press, 198–208. https://doi.org/10.1145/199448.199483

Paul-André Melliès. 2009. Categorical semantics of linear logic. Panoramas et Synthèses (2009).

Andrzej S. Murawski and Nikos Tzevelekos. 2021. Game Semantics for Interface Middleweight Java. *J. ACM* 68, 1 (2021), 4:1–4:51. https://doi.org/10.1145/3428676

Koko Muroya and Dan R. Ghica. 2019. The Dynamic Geometry of Interaction Machine: A Token-Guided Graph Rewriter. *Log. Methods Comput. Sci.* 15, 4 (2019).

Koko Muroya, Naohiko Hoshino, and Ichiro Hasuo. 2016. Memoryful geometry of interaction II: recursion and adequacy. In *POPL*. ACM, 748–760.

Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. 1981. Petri Nets, Event Structures and Domains, Part I. *Theor. Comput. Sci.* 13 (1981), 85–108. https://doi.org/10.1016/0304-3975(81)90112-2

Gordon Stewart, Lennart Beringer, Santiago Cuellar, and Andrew W. Appel. 2015. Compositional CompCert. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015*, Sriram K. Rajamani and David Walker (Eds.). ACM, 275–287. https://doi.org/10.1145/2676726.2676985

Glynn Winskel. 1982. Event Structure Semantics for CCS and Related Languages. In *Automata, Languages and Programming, 9th Colloquium, Aarhus, Denmark, July 12-16, 1982, Proceedings (Lecture Notes in Computer Science)*, Mogens Nielsen and Erik Meineche Schmidt (Eds.), Vol. 140. Springer, 561–576. https://doi.org/10.1007/BFb0012800

Li-yao Xia, Yannick Zakowski, Paul He, Chung-Kil Hur, Gregory Malecha, Benjamin C. Pierce, and Steve Zdancewic. 2020. Interaction trees: representing recursive and impure programs in Coq. *Proc. ACM Program. Lang.* 4, POPL (2020), 51:1–51:32. https://doi.org/10.1145/3371119

Basic red. for PCF

$$
\begin{array}{rcl}
(\lambda x^A.\, M)\, N & \rightsquigarrow & M[N/x] \\
\mathsf{if}\; b\; N_{\mathtt{tt}}\; N_{\mathtt{ff}} & \rightsquigarrow & N_b \\
\mathsf{Y}\, M & \rightsquigarrow & M\,(\mathsf{Y}\, M) \\
\mathsf{let}\; x = v \;\mathsf{in}\; M & \rightsquigarrow & M[v/x] \\
\mathsf{f}(v_1, v_2) & \rightsquigarrow & v \\
(\mathsf{if}\; \mathsf{f}(v_1, v_2) & = & v)
\end{array}
$$

Basic reductions for references and semaphores

$$
\begin{array}{rcl}
\mathsf{newref}\; x \;\mathsf{in}\; v & \rightsquigarrow & v \\
\mathsf{newsem}\; x \;\mathsf{in}\; v & \rightsquigarrow & v
\end{array}
$$

Stateful reductions

$$
\begin{array}{rcl}
\langle !\ell, s \uplus \{\ell \mapsto n\}\rangle & \rightsquigarrow & \langle n, s \uplus \{\ell \mapsto n\}\rangle \\
\langle \ell{:=}n, s \uplus \{\ell \mapsto \_\}\rangle & \rightsquigarrow & \langle \mathsf{skip}, s \uplus \{\ell \mapsto n\}\rangle \\
\langle \mathsf{grab}(\ell), s \uplus \{\ell \mapsto 0\}\rangle & \rightsquigarrow & \langle \mathsf{skip}, s \uplus \{\ell \mapsto 1\}\rangle \\
\langle \mathsf{release}(\ell), s \uplus \{\ell \mapsto n\}\rangle & \rightsquigarrow & \langle \mathsf{skip}, s \uplus \{\ell \mapsto 0\}\rangle \; (n > 0)
\end{array}
$$

Stateless context rules

$$
\frac{M \rightsquigarrow M'}{M\,N \rightsquigarrow M'\,N}
\qquad
\frac{M \rightsquigarrow M'}{\mathsf{if}\; M\; N_1\; N_2 \rightsquigarrow \mathsf{if}\; M'\; N_1\; N_2}
\qquad
\frac{M \rightsquigarrow M'}{!M \rightsquigarrow\, !M'}
\qquad
\frac{N \rightsquigarrow N'}{M{:=}N \rightsquigarrow M{:=}N'}
$$

$$
\frac{M \rightsquigarrow M'}{\mathsf{grab}(M) \rightsquigarrow \mathsf{grab}(M')}
\qquad
\frac{M \rightsquigarrow M'}{\mathsf{release}(M) \rightsquigarrow \mathsf{release}(M')}
\qquad
\frac{M \rightsquigarrow M'}{M{:=}v \rightsquigarrow M'{:=}v}
$$

$$
\frac{N \rightsquigarrow N'}{\mathsf{let}\; x = N \;\mathsf{in}\; M \rightsquigarrow \mathsf{let}\; x = N' \;\mathsf{in}\; M}
\qquad
\frac{M_1 \rightsquigarrow M_1'}{\mathsf{f}(M_1, M_2) \rightsquigarrow \mathsf{f}(M_1', M_2)}
\qquad
\frac{M_2 \rightsquigarrow M_2'}{\mathsf{f}(M_1, M_2) \rightsquigarrow \mathsf{f}(M_1, M_2')}
$$

Stateful context rules

$$
\frac{\langle M[\ell/x], s \uplus \{\ell \mapsto n\}\rangle \rightsquigarrow \langle M'[\ell/x], s' \uplus \{\ell \mapsto n'\}\rangle}{\langle \mathsf{newref}\; x{:=}n \;\mathsf{in}\; M, s\rangle \rightsquigarrow \langle \mathsf{newref}\; x{:=}n' \;\mathsf{in}\; M', s'\rangle}\;(\ell \in \mathcal{L}\text{ fresh})
$$

$$
\frac{\langle M[\ell/x], s \uplus \{\ell \mapsto n\}\rangle \rightsquigarrow \langle M'[\ell/x], s' \uplus \{\ell \mapsto n'\}\rangle}{\langle \mathsf{newsem}\; x{:=}n \;\mathsf{in}\; M, s\rangle \rightsquigarrow \langle \mathsf{newsem}\; x{:=}n' \;\mathsf{in}\; M', s'\rangle}\;(\ell \in \mathcal{L}\text{ fresh})
$$

Fig. 24. Operational semantics of IPA

A OPERATIONAL SEMANTICS

We include in Figure 24 the operational semantics of IPA. Here, $M \rightsquigarrow N$ stands for

$$\langle M, s\rangle \rightsquigarrow \langle N, s\rangle\,,$$

i.e. all rules operate on configurations $\langle M, s\rangle$, but only the relevant part of the tuple is shown.

B THE IPA-STRUCTURE Strat

B.1 Categorical structure

Section 4.2 already contains the data of Strat, and composition. It remains to define:

B.1.1 Copycat. Copycat may be defined on any game, but it is slightly simpler on arenas:

Definition B.1. For each arena A, the **copycat strategy** $\mathrm{cc}_A : A \vdash A$ is defined as having:

$$
\begin{array}{rcl}
|\mathrm{cc}_A| & = & |A \vdash A| \\
\partial_{\mathrm{cc}_A}(m) & = & m \\
\imath(a) \leq_{\mathrm{cc}_A} \imath'(a) & \Leftrightarrow & a <_A a';\text{ or } a = a',\ \mathrm{pol}(\imath(a)) = - \text{ and } \mathrm{pol}(\imath'(a)) = + \\
\imath(a)\ \#_{\mathrm{cc}_A}\ \imath'(a') & \Leftrightarrow & a\ \#_A\ a'\,,
\end{array}
$$

where $\imath, \imath' \in \{\ell_\vdash, r_\vdash\}$.

Copycat acts as an asynchronous forwarder, simply receptive to all Opponent moves and prepared to forward them to the other side as soon as they become available. This means that its +-covered configurations, where all moves have been successfully forwarded, have a particularly simple shape:

Lemma B.2. *Consider A any arena. Then, we have $\mathscr{C}^+(\text{c\!c}_A) = \{x_A \vdash x_A \in \mathscr{C}(A \vdash A) \mid x_A \in \mathscr{C}(A)\}$.*

From Lemma 4.9 this characterizes $\text{c\!c}_A$ uniquely up to iso, just like Proposition 4.11 for composition. It follows from these two facts that composition preserves isomorphisms, that it is associative and that identities are neutral for composition up to isomorphism, see [Castellan et al. 2017] for details.

Corollary B.3. *There is* Strat*, a category having arenas as objects, as morphisms from A to B the negative strategies on the game $A \vdash B$ up to isomorphism, and copycat strategies as identities.*

Unlike PStruct or PStrat, Strat *is* a category satisfying the identity laws – though this fact will not be directly useful for us in this paper.

B.2 Strat as an IPA-structure: Operations

We detail the different operations involved in the IPA-structure.

B.2.1 Tensor. Fix $\sigma_1 \in \mathsf{Strat}(A_1, B_1)$ and $\sigma_2 \in \mathsf{Strat}(A_2, B_2)$. We define:

Definition B.4. We define $\sigma_1 \otimes \sigma_2 \in \mathsf{Strat}(A_1 \otimes A_2, B_1 \otimes B_2)$ with:

$$\begin{array}{rcl} |\sigma_1 \otimes \sigma_2| &=& |\sigma_1| + |\sigma_2| \\ \leq_{\sigma_1 \otimes \sigma_2} &=& \leq_{\sigma_1} + \leq_{\sigma_2} \\ \#_{\sigma_1 \otimes \sigma_2} &=& \#_{\sigma_1} + \#_{\sigma_2} \\ \partial_{\sigma_1 \otimes \sigma_2}(\ell(m)) &=& \partial_{\sigma_1}(m) \\ \partial_{\sigma_1 \otimes \sigma_2}(r(m)) &=& \partial_{\sigma_2}(m) \end{array}$$

The conditions for a strategy are straightforward, and so is:

Proposition B.5. *The strategy $\sigma_1 \otimes \sigma_2 \in \mathsf{Strat}(A_1 \otimes A_2, B_1 \otimes B_2)$ satisfies:*

$$(- \otimes -) : \mathscr{C}^+(\sigma_1) \times \mathscr{C}^+(\sigma_2) \to \mathscr{C}^+(\sigma_1 \otimes \sigma_2)$$

such that $\partial_{\sigma_1 \otimes \sigma_2}(x^{\sigma_1} \otimes x^{\sigma_2}) = (x^{\sigma_1}_{A_1} \otimes x^{\sigma_2}_{A_2}) \vdash (x^{\sigma_1}_{B_1} \otimes x^{\sigma_2}_{B_2})$ for all $x^{\sigma_1} \otimes x^{\sigma_2} \in \mathscr{C}^+(\sigma_1 \otimes \sigma_2)$.
Moreover, $\sigma_1 \otimes \sigma_2$ is the unique *strategy on $A_1 \otimes A_2 \vdash B_1 \otimes B_2$ satisfying this property.*

Proof. The property is a direct verification, and uniqueness follows from Lemma 4.9. □

B.2.2 Currying. As for Petri structures, we start with *renaming*.

Definition B.6. Consider σ a strategy on game A, and $f : |A| \to |B|$.
Then, we define the **renaming** to be as σ except $\partial_{\sigma[f]}(m) = f(\partial_\sigma(m))$.

Without additional conditions, there is no reason why $\sigma[f]$ would be a strategy in general. A convenient situation is when f preserves sufficiently rigidly the rules of the game:

Proposition B.7. *We say that $f : |A| \to |B|$ is **valid** if it is a map of es, additionally satisfying hypotheses* receptive *and* courteous *from Definition 4.6*
If $\sigma : A$ and f is valid, then $\sigma[f]$ is a strategy on B.

Proof. Straightforward. □

However, we cannot use this directly for currying, because the function

$$\begin{array}{rcrcl} \Lambda^{\Gamma,\Delta}_{x:A,O} & : & |\,!\,\&[\Gamma, x : A, \Delta] \vdash O| & \to & |\,!\,\&[\Gamma, \Delta] \vdash !A \multimap O| \\ & & (\mathsf{m}, \mathsf{s}, \mathsf{d}) & \mapsto & (\Lambda^x(\mathsf{m}), \mathsf{s}, \mathsf{d}) \end{array}$$

using Λ_x from Definition 3.15, is not valid (a singleton configuration in A on the left hand side is indeed sent to a non-configuration on the right hand side). However, we do have:

Proposition B.8. *Consider $\sigma \in \mathsf{Strat}(!\,\&[\Gamma, x : A, \Delta], O)$.*
Then, there exists a unique $\Lambda(\sigma) \in \mathsf{Strat}(!\,\&[\Gamma, x : A, \Delta] \vdash O)$ such that

$$\varphi : \mathscr{C}^+(\sigma) \cong \mathscr{C}^+(\Lambda(\sigma))$$

and satisfying that $\partial_{\Lambda(\sigma)}(\varphi(x^\sigma)) = \Lambda^{\Gamma,\Delta}_{x:A,O}(\partial_\sigma(x^\sigma))$ for all $x^\sigma \in \mathscr{C}^+(\sigma)$.

Proof. *Existence.* We set $\Lambda(\sigma)$ as $\sigma[\Lambda^{\Gamma,\Delta}_{x:A,O}]$ even though $\Lambda^{\Gamma,\Delta}_{x:A,O}$ is not valid; that this is still well-defined follows directly from σ negative (see [Castellan and Clairambault 2020, Lemma 4.25]).
Uniqueness. Direct from Lemma 4.9. □

B.2.3 Promotion. Next we define the promotion of $\sigma \in \mathsf{Strat}(!A, B)$.
First, for any arena A, we define the function

$$\begin{array}{rccl} \mathsf{dig}_A & : & !!A & \to & !A \\ & & \mathsf{e}_1 :: (\mathsf{e}_2 :: m) & \mapsto & \langle \mathsf{e}_1, \mathsf{e}_2 \rangle :: m \end{array}$$

yielding a map of event structures. If σ is a strategy, we write $\mathscr{C}^{+,\neq\emptyset}(\sigma)$ the set of +-covered, non-empty configurations of σ. Finally, for X a set we write $\mathsf{Fam}(X)$ for the set of families $(x_i)_{i\in I}$ where $x_i \in X$ and $I \subseteq \mathcal{E}$ is a finite subset of exponential signatures.
With these notations in place, we have:

Proposition B.9. *There is a strategy $\sigma^\dagger \in \mathsf{Strat}(!A, !B)$, unique up to iso, such that there is*

$$[-] : \mathsf{Fam}(\mathscr{C}^{+,\neq\emptyset}(\sigma)) \cong \mathscr{C}^+(\sigma^\dagger)$$

satisfying that for all $(x^{\mathsf{e}})_{\mathsf{e}\in E} \in \mathsf{Fam}(\mathscr{C}^{+,\neq\emptyset}(\sigma))$, we have

$$\partial_{\sigma^\dagger}([(x^{\mathsf{e}})_{\mathsf{e}\in E}]) = \mathsf{dig}\left(\biguplus_{\mathsf{e}\in E} \mathsf{e} :: x^{\mathsf{e}}_{!A}\right) \vdash \left(\biguplus_{\mathsf{e}\in E} \mathsf{e} :: x^{\mathsf{e}}_{B}\right)$$

writing $\partial_\sigma(x^{\mathsf{e}}) = x^{\mathsf{e}}_{!A} \vdash x^{\mathsf{e}}_B$ for all $\mathsf{e} \in E$.

Proof. *Existence.* Straightforward from [Castellan and Clairambault 2020, Definition 4.27] and renaming following dig.
Uniqueness. Direct from Lemma 4.9. □

B.3 Strat as an IPA-Structure: Primitives

B.3.1 Copycat strategies. We first address the three primitives arising as copycat-like strategies: variable, evaluation, and contraction.

Definition B.10. Consider $\Gamma, x : A, \Delta$ a semantic context. Then we set $\mathsf{Var}^{\Gamma,\Delta}_{x:A}$ as $\mathrm{œ}_A[\mathsf{Var}_x]$ where

$$\begin{array}{rcrcl} \mathsf{Var}_x & : & (A \vdash A) & \to & (!\,\&[\Gamma, x : A, \Delta] \vdash A) \\ & & (\iota_\vdash^r \mathsf{m}, \mathsf{s}, \mathsf{d}) & \mapsto & (\iota_\vdash^r \mathsf{m}, \mathsf{s}, \mathsf{d}) \\ & & (\ell_\vdash \mathsf{m}, \mathsf{s}, \mathsf{d}) & \mapsto & (\ell_\vdash \iota^x_\& \mathsf{m}, \blacklozenge :: \mathsf{s}, \mathsf{d}) \end{array}$$

Likewise, the evaluation morphism is simply by renaming.

Definition B.11. Consider A, O arenas with O well-opened.
Then we set $\mathsf{ev}_{A,O} = \mathrm{œ}_{A\multimap O}[\Omega]$ where Ω is that of Definition 3.18 canonically extended to moves.

Finally, we define copycat. As in the main text, for simplicity we give the binary case.

Definition B.12. Consider A an arena. Then we set $\mathrm{c}_A = \text{\oe}_{!A\otimes !A}[\mathrm{c}] \in \mathrm{Strat}(!A, !A \otimes !A)$ where

$$\begin{array}{rcrcl} \mathrm{c} & : & (!A \otimes !A \vdash !A \otimes !A) & \rightharpoonup & (!A \vdash !A \otimes !A) \\ & & (\ell_{\vdash}\ell_{\otimes}\mathrm{m}, \mathrm{e} :: \mathrm{s}, \mathrm{d}) & \mapsto & (\ell_{\vdash}\mathrm{m}, (\ell_{!}\mathrm{e}) :: \mathrm{s}, \mathrm{d}) \\ & & (\ell_{\vdash}\mathscr{r}_{\otimes}\mathrm{m}, \mathrm{e} :: \mathrm{s}, \mathrm{d}) & \mapsto & (\ell_{\vdash}\mathrm{m}, (\mathscr{r}_{!}\mathrm{e}) :: \mathrm{s}, \mathrm{d}) \\ & & (\mathscr{r}_{\vdash}\mathrm{m}, \mathrm{s}, \mathrm{d}) & \mapsto & (\mathscr{r}_{\vdash}\mathrm{m}, \mathrm{s}, \mathrm{d}) \end{array}$$

For the unfolding, it will be convenient to have the following characterization:

Proposition B.13. *For any arena A, $\mathscr{C}^+(\mathrm{c}_A) = \{\ell_!(x_{!A}) \uplus \mathscr{r}_!(y_{!A}) \vdash x_{!A} \otimes y_{!A} \mid x_{!A}, y_{!A} \in \mathscr{C}(!A)\}$.*

Proof. Immediate by Lemma B.2 and definition. □

B.3.2 Constants, conditional, queries. The strategies are displayed in Figure 25. Note that some of these diagrams use a symbolic representation; whenever there is a branch starting with a negative move with some data, there actually is a branch for any instance of the data allowed in the game.

B.3.3 Let. We illustrate the strategy **let** in Figure 26. Note that there is a similar call to $!\mathbf{X}$ for all exponential signature $\mathrm{e} \in \mathcal{E}$.

B.3.4 Recursion. In [Castellan and Clairambault 2020; Castellan et al. 2019], the recursion combinator is obtained via the usual recipe in denotational semantics, as the least fixed point of

$$F \mapsto (f : O \to O \vdash F\, f)\,.$$

Let us give a direct description of the strategy obtained (the recursive equation gives a different choice of copy indices, which does not matter up to the equivalence of strategies in [Castellan and Clairambault 2020; Castellan et al. 2019] – the choice we use here allows for a lighter presentation).

Let us start by drawing the strategy on $\mathbf{U}$. We use particular exponential signatures generated by

$$\begin{array}{rcl} () & := & \ell_!\blacklozenge \\ (\mathrm{e}_{n+1}, \mathrm{e}_n, \ldots, \mathrm{e}_1) & := & \mathscr{r}_!\langle \mathrm{e}_{n+1}, (\mathrm{e}_n, \ldots, \mathrm{e}_1)\rangle\,, \end{array}$$

yielding $(\mathrm{e}_n, \ldots, \mathrm{e}_1) \in \mathcal{E}$ for each $\mathrm{e}_1, \ldots, \mathrm{e}_n \in \mathcal{E}$. With this convention, we draw the recursion combinator for $\mathbf{U}$ in Figure 27. Again the representation is symbolic, with similar branches for all $\mathrm{e}_1, \ldots, \mathrm{e}_{n+1} \in \mathcal{E}$. We leave in grey the answers, which always propagate back to the previous call.

We consider Y_O for O well-opened, so the recursion strategy in general has a spine exactly as the black part of Figure 27. The rest of the strategy is simple copycat behaviour; which may be simply described via the following direct characterization of the $+$-covered configurations of Y_O:

Proposition B.14. *For O a well-opened arena, the strategy Y_O has **events** a subset $|\mathrm{Y}_O| \subseteq |!(!O \multimap O) \vdash O|$, and $+$-**covered configurations** the compatible unions of configurations of the form*

$$\begin{array}{rcl} \emptyset \multimap (() :: x) & \vdash & x \\ (\mathrm{e}_{n+1} :: (\mathrm{e}_n, \ldots, \mathrm{e}_1) :: x) \multimap ((\mathrm{e}_{n+1}, \ldots, \mathrm{e}_1) :: x) & \vdash & \emptyset\,, \end{array}$$

*for $n \in \mathbb{N}, \mathrm{e}_1, \ldots, \mathrm{e}_{n+1} \in \mathcal{E}, x \in \mathscr{C}(O)$; **compatible** means that the union is in $\mathscr{C}(!(!O \multimap O) \vdash O)$.*

We slightly reformulate this proposition to give a more explicit description of $+$-covered configurations of Y_O. In the next lemma, we use the injection of $(\cdot) : \mathcal{E}^* \to \mathcal{E}$.

Lemma B.15. *There is an order-isomorphism between $\mathscr{C}^+(\mathrm{Y}_O)$ and tuples $\langle J \subseteq \mathcal{E}^+, z \in \mathscr{C}(O), (x_{\mathrm{s}} \in \mathscr{C}^{\neq\emptyset}(O))_{\mathrm{s}\in J}\rangle$ such that J is suffix-closed, and empty if z is.*[1] *The isomorphism sends $\langle J, z, (x_{\mathrm{s}})\rangle$ to*

$$(\emptyset \multimap () :: z) \uplus (\mathrm{e} :: (\mathrm{s}) :: x_{\mathrm{e}\cdot\mathrm{s}} \multimap (\mathrm{e}\cdot\mathrm{s}) :: x_{\mathrm{e}\cdot\mathrm{s}})_{\mathrm{e}\cdot\mathrm{s}\in J} \vdash z\,.$$

[1] A stack in J represents a call stack: a list $\mathrm{e}_n \cdot \mathrm{e}_{n-1} \cdot \ldots \cdot \mathrm{e}_1$ represents the calls made to the argument: e_1 is the first call made in the execution, and so on.

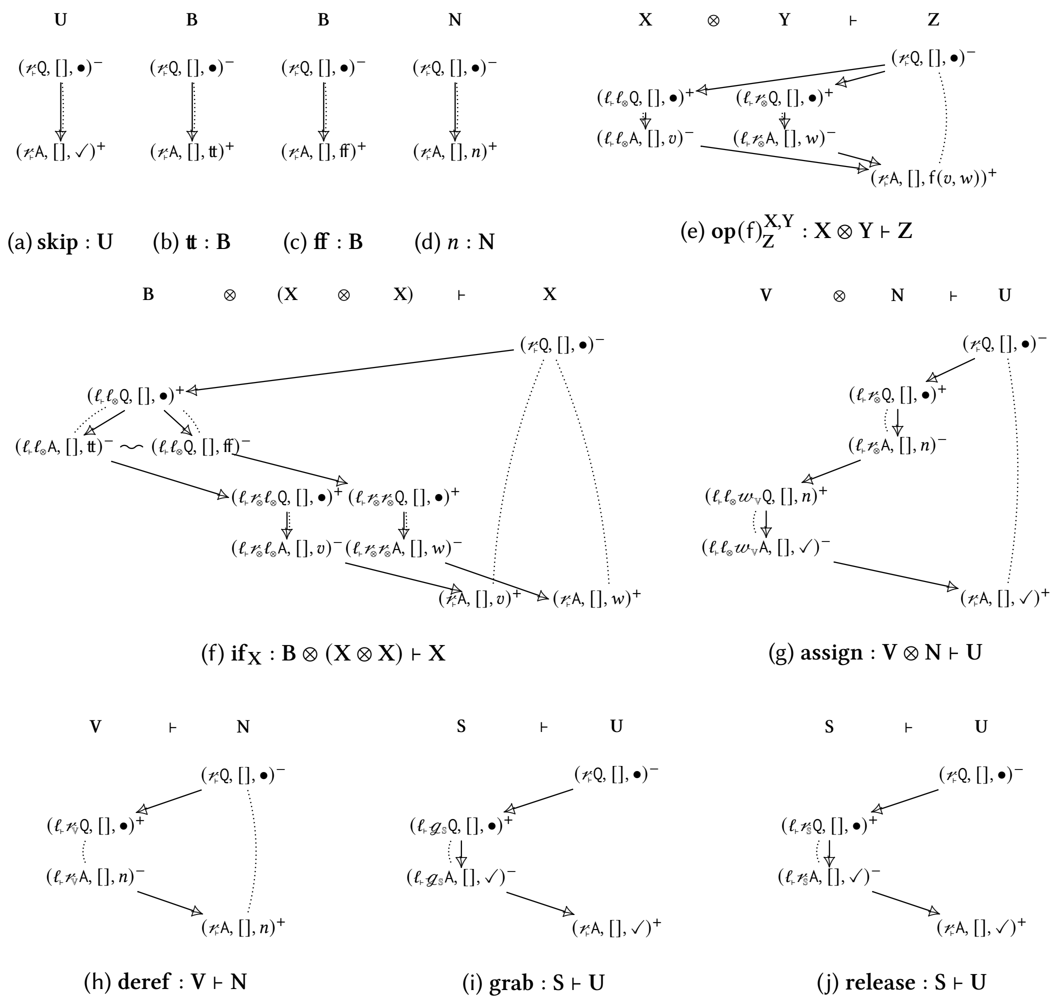


Fig. 25. Basic strategies for IPA primitives

B.3.5 New reference. Next, we introduce the strategy for reference initialization.

The intuition is that $\mathbf{newref}_{\mathbb{X}} : \mathbf{!V} \multimap \mathbb{X} \vdash \mathbb{X}$ applies its argument to $\mathsf{cell} : \mathbf{!V}$, the *memory cell*, which implements the stateful behaviour. Just like an actual memory, cell is inherently *sequential*: it treats read and write requests in some sequential order. In order to define it, we first define, for all $n \in \mathbb{N}$, cell_n as the language of non-empty prefixes of the infinite tree $\mathsf{cell}_n^{\emptyset}$, defined as:

$$\mathsf{cell}_n^E = (\not{r}_{\mathbb{V}}\mathsf{Q}, [\mathsf{e}], \bullet)^- \cdot (\not{r}_{\mathbb{V}}\mathsf{A}, [\mathsf{e}], n)^+ \cdot \mathsf{cell}_n^{E \uplus \{\mathsf{e}\}} \mid (w_{\mathbb{V}}\mathsf{Q}, [\mathsf{e}], k)^- \cdot (w_{\mathbb{V}}\mathsf{A}, [\mathsf{e}], \checkmark)^+ \cdot \mathsf{cell}_k^{E \uplus \{\mathsf{e}\}},$$

with $\mathsf{e} \notin E$. Intuitively, words in cell_n^E are alternating (*i.e.* Opponent and Player alternate) executions of read and write requests, for a memory cell initialized with value n: upon a read request, the memory cell returns n and carries on with cell_n. Upon a write request for k, the memory cell returns an acknowledgement and proceeds as cell_k. The set E propagates the set of exponential signatures already encountered: this lets us always pick fresh exponential signatures for new queries, ensuring words in cell_n^E are plays on $\mathbf{!V}$ in the sense of Definition 5.1. We may then define:

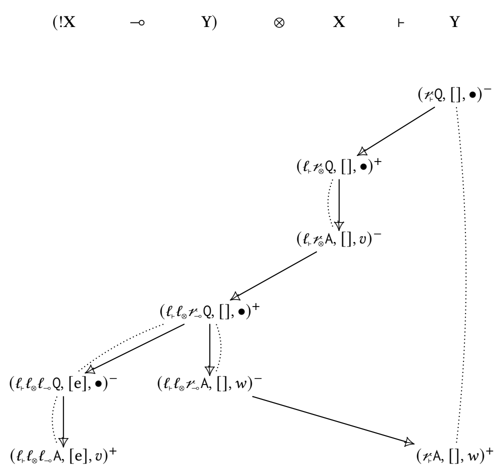


Fig. 26. $\mathbf{let}_{\mathbf{X},\mathbf{Y}} : (!\mathbf{X} \multimap \mathbf{Y}) \otimes \mathbf{X} \vdash \mathbf{Y}$

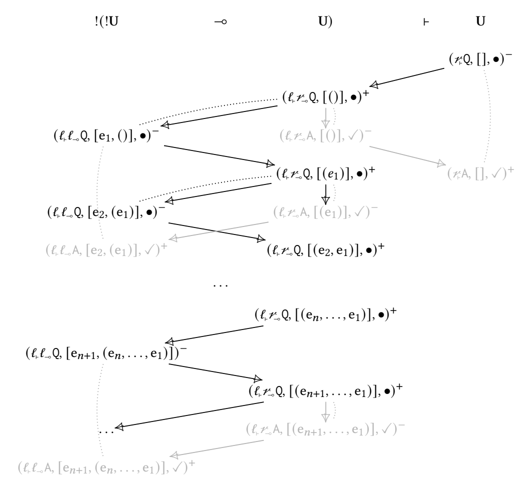


Fig. 27. Fixpoint combinator on $\mathbf{U}$

Definition B.16. We define a *prestrategy* precell : $!\mathbf{V}$ with components:

$$
\begin{array}{rcl}
|\mathsf{precell}| & = & \mathsf{cell}_0 \\
\leq_{\mathsf{precell}} & = & \sqsubseteq \\
s \;\#_{\mathsf{precell}}\; s' & \Leftrightarrow & \neg(s \sqsubseteq s' \vee s' \sqsubseteq s) \\
\partial_{\mathsf{precell}}(sa) & = & a\,,
\end{array}
$$

where $\sqsubseteq$ is the prefix ordering.

It is easy to see that this indeed defines a prestrategy. An illustration may be found in [Castellan and Clairambault 2020, Figure 39] (with a slightly different notation for moves). Next we define:

Definition B.17. We define a *strategy* cell : $!\mathbf{V}$ as: $\mathsf{cell} = \mathrm{cc}_{!\mathbf{V}} \odot \mathsf{precell} : !\mathbf{V}$.

Indeed, composition is well-defined on prestrategies, and the copycat envelope of a prestrategy is always a strategy [Castellan et al. 2017]. Intuitively, this wraps the sequential behaviour of cell by buffers, which exactly match the buffers of the Petri structure in Figure 15g.

To obtain the strategy for $\mathsf{newref}_{\mathbf{X}}$, we must add a copycat behaviour on $\mathbf{X}$:

Definition B.18. We define a strategy $\mathsf{newref}_{\mathbf{X}} : !\mathbf{V} \multimap \mathbf{X} \vdash \mathbf{X}$ with components:

$$
\begin{array}{rcl}
|\mathsf{newref}_{\mathbf{X}}| & = & |\mathsf{cell}| + |\mathrm{cc}_{\mathbf{X}}| \\
\leq_{\mathsf{newref}_{\mathbf{X}}} & = & (\leq_{\mathsf{cell}} + \leq_{\mathrm{cc}_{\mathbf{X}}}) \\
 & & \uplus\{(\mathscr{r}((\mathscr{r}_{\vdash}\mathsf{Q}, [], \bullet)), \ell(s)) \mid m \in |\mathsf{cell}|\} \\
 & & \uplus\{(\mathscr{r}((\ell_{\vdash}\mathsf{Q}, [], \bullet)), \ell(s)) \mid m \in |\mathsf{cell}|\} \\
\#_{\mathsf{newref}_{\mathbf{X}}} & = & \#_{\mathsf{cell}} + \#_{\mathrm{cc}_{\mathbf{X}}} \\
\partial_{\mathsf{newref}_{\mathbf{X}}}(\ell(m)) & = & \ell_{\vdash}\ell_{\multimap}\partial_{\mathsf{cell}}(m) \\
\partial_{\mathsf{newref}_{\mathbf{X}}}(\mathscr{r}(\ell_{\vdash}m)) & = & \ell_{\vdash}\mathscr{r}_{\multimap}m \\
\partial_{\mathsf{newref}_{\mathbf{X}}}(\mathscr{r}(\mathscr{r}_{\vdash}m)) & = & \mathscr{r}_{\vdash}m
\end{array}
$$

So cell is plugged after the first Player move of $\mathrm{cc}_{\mathbf{X}}$. In other words, $\mathsf{newref}_{\mathbf{X}}$ first plays as copycat on $\mathbf{X}$; and plays as cell on $!\mathbf{V}$ when it becomes available.

For the later unfolding, we shall use the following characterization of +-covered configurations:

Proposition B.19. *There is an order-isomorphism:*

$$\langle -, - \rangle : \mathscr{C}^{+}(\mathsf{precell}) \times \mathscr{C}^{\neq\emptyset}(\mathbf{X}) \cong \mathscr{C}^{+,\neq\emptyset}(\mathsf{newref}_{\mathbf{X}})$$

such that $\partial_{\mathsf{newref}_{\mathbf{X}}}(\langle s, x\rangle) = |s| \multimap x \vdash x \in \mathscr{C}(!\mathbf{V} \multimap \mathbf{X} \vdash \mathbf{X})$.

Proof. By Definition B.18, the characterization of confs. of composition, and Lemma B.2. □

Above, we implicitly use the one-to-one correspondence between $\mathscr{C}^{+}(\mathsf{precell})$ and even-length words in cell_0; and if s is an even length word in cell_0, then $|s| \in \mathscr{C}(!\mathbf{V})$ is its set of events.

In a +-covered configuration of newref, all read and write requests have been successfully handled. Proposition B.19 shows that besides the almost independent copycat behaviour on $\mathbf{X}$, +-covered configurations of newref exactly correspond to some sequential ordering of these requests.

B.3.6 New semaphore. The interpretation of semaphores work exactly as for references. We first define the alternating behaviour of a semaphore as the language of non-empty prefixes of:

$$\begin{array}{rcll}
\mathsf{lock}_0^E & = & (\mathscr{g}_{\mathbb{S}}\mathsf{Q}, [\mathsf{e}], \bullet)^- \cdot (\mathscr{g}_{\mathbb{S}}\mathsf{A}, [\mathsf{e}], \checkmark)^+ \cdot \mathsf{lock}_1^{E \uplus \{\mathsf{e}\}} & \mathsf{e} \notin E \\
\mathsf{lock}_n^E & = & (\mathscr{r}_{\mathbb{S}}\mathsf{Q}, [\mathsf{e}], \bullet)^- \cdot (\mathscr{r}_{\mathbb{S}}\mathsf{A}, [\mathsf{e}], \checkmark)^+ \cdot \mathsf{lock}_0^{E \uplus \{\mathsf{e}\}} & \mathsf{e} \notin E, n > 0
\end{array}$$

A semaphore with value 0 may be grabbed, carrying on with value 1. A semaphore with value $n > 0$ may be released, carrying on with value 0. As for references, the next step is to form:

Definition B.20. We define a *prestrategy* prelock : !S with components:

$$\begin{array}{rcl}
|\mathsf{prelock}| & = & \mathsf{lock}_0 \\
\leq_{\mathsf{prelock}} & = & \sqsubseteq \\
s \,\#_{\mathsf{prelock}}\, s' & \Leftrightarrow & \neg(s \sqsubseteq s' \vee s' \sqsubseteq s) \\
\partial_{\mathsf{prelock}}(sa) & = & a\,.
\end{array}$$

Definition B.21. We define a strategy lock : !S as $\mathsf{lock} = \mathrm{cc}_{!\mathbf{S}} \odot \mathsf{prelock} : !\mathbf{S}$.

Definition B.22. We define a strategy $\mathsf{newsem}_{\mathbf{X}} : !\mathbf{S} \multimap \mathbf{X} \vdash \mathbf{X}$ with components:

$$\begin{array}{rcl}
|\mathsf{newsem}_{\mathbf{X}}| & = & |\mathsf{lock}| + |\mathrm{cc}_{\mathbf{X}}| \\
\leq_{\mathsf{newref}_{\mathbf{X}}} & = & (\leq_{\mathsf{lock}} + \leq_{\mathrm{cc}_{\mathbf{X}}}) \\
& & \uplus\{(\mathscr{r}((\mathscr{r}_{\vdash}\mathsf{Q}, [\,], \bullet)), \ell(s)) \mid m \in |\mathsf{lock}|\} \\
& & \uplus\{(\mathscr{r}((\ell_{\vdash}\mathsf{Q}, [\,], \bullet)), \ell(s)) \mid m \in |\mathsf{lock}|\} \\
\#_{\mathsf{newsem}_{\mathbf{X}}} & = & \#_{\mathsf{lock}} + \#_{\mathrm{cc}_{\mathbf{X}}} \\
\partial_{\mathsf{newsem}_{\mathbf{X}}}(\ell(m)) & = & \ell_{\vdash}\ell_{\multimap}\partial_{\mathsf{lock}}(m) \\
\partial_{\mathsf{newsem}_{\mathbf{X}}}(\mathscr{r}(\ell_{\vdash}m)) & = & \ell_{\vdash}\mathscr{r}_{\multimap}m \\
\partial_{\mathsf{newsem}_{\mathbf{X}}}(\mathscr{r}(\mathscr{r}_{\vdash}m)) & = & \mathscr{r}_{\vdash}m
\end{array}$$

And, finally, we have a similar characterization of (non-empty) +-covered configurations:

Proposition B.23. *There is an order-isomorphism:*

$$\langle -, - \rangle : \mathscr{C}^{+}(\mathsf{prelock}) \times \mathscr{C}^{\neq\emptyset}(\mathbf{X}) \cong \mathscr{C}^{+,\neq\emptyset}(\mathsf{newsem}_{\mathbf{X}})$$

such that $\partial_{\mathsf{newsem}_{\mathbf{X}}}(\langle s, x\rangle) = |s| \multimap x \vdash x \in \mathscr{C}(!\mathbf{S} \multimap \mathbf{X} \vdash \mathbf{X})$.

This concludes the description of the IPA-structure of Strat.

B.4 Complement: on symmetry

B.4.1 Games and strategies with symmetry. *Thin concurrent games* [Castellan et al. 2019] share with AJM games the fact that strategies play explicit copy indices – in [Castellan et al. 2019] and AJM games those are natural numbers, whereas here they are exponential signatures. The consequence is that in order to satisfy required equational laws (typically, making ! a well-behaved exponential modality), one must be able to consider strategies up to their choice of copy indices.

In concurrent games, this *reindexing* is handled by *event structures with symmetry*:

Definition B.24. A **event structure with symmetry** is $E = (|E|, \leq_E, \#_E, \mathcal{S}(E))$ is an es $(|E|, \leq_E, \#_E)$ with $\mathcal{S}(E)$ a set of bijections between configurations:

$$\theta : x \cong_E y$$

comprising all identity bijections, closed under composition and inverse, and satisfying further bisimulation-like properties, omitted here [Castellan et al. 2019].

In [Castellan and Clairambault 2020; Castellan et al. 2019], both games and strategies are event structures with symmetry. Intuitively, in a game A, we have $\theta : x \cong_A y$ in $\mathcal{S}(A)$ when θ is an order-isomorphism only affecting copy indices – in the terminology of this paper, it changes the exponential signatures in the exponential stack, but leaves all other components unchanged.

The symmetry on the game yields a more permissive equivalence on strategies: namely, a *weak isomorphism* $\varphi : \sigma \simeq \tau$ is an invertible map of event structure such that the triangle

$$\begin{tikzcd} \sigma \arrow[rr, "\varphi"] \arrow[dr, "\partial_\sigma"'] & & \tau \arrow[dl, "\partial_\tau"] \\ & A & \end{tikzcd}$$

commutes *up to symmetry*, defined as $\{(\partial_\sigma s, \partial_\tau(\varphi(s))) \mid s \in x\} \in \mathcal{S}(A)$ for all $x \in \mathscr{C}(\sigma)$. Weak isomorphism makes the exponential satisfy all the required laws (typically, making each $!A$ a commutative comonoid), which were not satisfied up to plain isomorphism.

In turn, composition must preserve weak isomorphism. But that holds only for strategies that are *uniform*, *i.e.* invariant under the choice of copy indices. In [Castellan and Clairambault 2020; Castellan et al. 2019], uniformity of strategies is ensured by also adjoining them a symmetry. On a strategy $\sigma : A$, the bisimulation-like properties of $\mathcal{S}(\sigma)$ ensures that if Opponent changes their copy indices, σ may change its copy indices accordingly, but not more. This makes $\simeq$ a congruence, and strategies up to $\simeq$ satisfy all the required equational laws to model higher-order languages. This issue is discussed at length in [Castellan et al. 2019].

B.4.2 Removing symmetry. The model developed in [Castellan and Clairambault 2020] is a structure StratSym with two equivalences $\cong$ (standard isomorphism) and $\simeq$ (weak isomorphism[2]). Both are preserved by all constructions, but the laws of Seely categories are satisfied with respect to $\simeq$ only.

Now, the first observation is that though $\simeq$ is crucial in establishing adequacy for StratSym (for instance, the β-law in IPA is validated by the interpretation only up to $\simeq$), the statement itself (Theorem 4.40 in [Castellan and Clairambault 2020]) is independent of the equivalence relation. So once adequacy is established we can ignore $\simeq$, and from [Castellan and Clairambault 2020] we get an IPA-structure StratSym$/\cong$ with an adequate interpretation of IPA.

The next point is that symmetries do not carry operational behaviour, they are merely there to witness uniformity so that $\simeq$ is a congruence. As mere uniformity witnesses, they can be safely forgotten once $\simeq$ is out of the picture. Concretely, in all operations involved in the interpretation,

[2] In fact, [Castellan and Clairambault 2020] uses a variant called *positive iso*, but the difference is irrelevant for this discussion.

symmetries of the operand strategies are used in computing the symmetries of the resulting strategies only – the event structure itself never depends on the symmetries of operands. Consequently:

Proposition B.25. *There is a symmetry-forgetting* IPA*-functor* $\mathsf{StratSym}/\cong \;\to\; \mathsf{Strat}$.

From this, it follows by Lemma 2.3 that Strat is an adequate IPA-structure.

On a foundational level, it would be interesting to see how symmetries may be obtained by unfolding just as plain strategies. It would require setting up symmetries between histories of runs of Petri strategies as well. But this is not necessary for our purposes, so we leave that for later.

C THE IPA-STRUCTURE PStrat

C.1 The Precategory PStrat

The main step is to prove that Petri strategies are stable under composition.

C.1.1 Composition. Consider A, B, C arenas, $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ Petri strategies.

The idea is simple: as Petri strategies, both $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ abide by the rules of the game as long as the external Opponent does so. As no agent can be the first to break the rules, the whole interaction ends up correct. This kind of reasoning is very common in game semantics. To formalize it, the difficulty is not conceptual but purely notational: we need tools to project a run ρ on $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ to runs on $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$, and to an *interaction* in more familiar game-semantic terms:

Definition C.1. Consider the set $\mathsf{MInt} = \ell(\mathsf{Moves}) \uplus m(\mathsf{Moves}) \uplus r(\mathsf{Moves})$.
An **interaction** is a sequence $u \in \mathsf{MInt}^*$. We write Int for the set of all interactions.

As in traditional play-based game semantics, an interaction has three components: $\boldsymbol{\sigma}$ "plays" on ℓ, m, $\boldsymbol{\tau}$ "plays" on m, r, and the composite $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ "plays" on ℓ, r. Again as in game semantics, we shall *restrict* interactions to these various components. In more generality, we define:

Definition C.2. Consider $f : X \rightharpoonup Y$, and $s \in X^*$, we define $s \upharpoonright f \in Y^*$ the **restriction of** s **following** s as $\varepsilon \upharpoonright f = \varepsilon$, $sa \upharpoonright f = (s \upharpoonright f)f(a)$ if $f(a)$ is defined, and $sa \upharpoonright f = s \upharpoonright f$ otherwise.

A first direct application of this notion is to project $u \in \mathsf{Int}$ to its various components with

$$u_{\boldsymbol{\sigma}} = u \upharpoonright \ell_{\vdash}\ell^* \cup r_{\vdash}m^*\,, \qquad u_{\boldsymbol{\tau}} = u \upharpoonright \ell_{\vdash}m^* \cup r_{\vdash}r\,, \qquad u_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} = u \upharpoonright \ell_{\vdash}\ell^* \cup r_{\vdash}r^*$$

where $\ell^* : \ell(\mathsf{Moves}) \rightharpoonup \mathsf{Moves}$ sends $\ell(m)$ to m and is undefined otherwise, and likewise for m^* and r^*. Juxtaposition is function composition, and $\cup$ is the union of their graph.

We also use restriction to extract from a run ρ on $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ an *interaction*, and runs $\rho_{\boldsymbol{\sigma}}$ and $\rho_{\boldsymbol{\tau}}$ on $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ respectively. But the partial functions involved are more complex, and require us to understand better the shape of instantiated transitions of $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$:

Lemma C.3. *Consider $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ Petri structures.*
Then, instantiated transitions of $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ are exactly as in Figure 28 – in the sense that that there is a one-to-one correspondence between instantiated transitions in the premises and in the conclusion.

Using this description, we define in Figure 29 partial functions $\mathsf{lbl}_{\circledast} : \mathsf{IT}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \rightharpoonup \mathsf{MInt}$, $\pi_{\boldsymbol{\sigma}} : \mathsf{IT}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \rightharpoonup \mathsf{IT}_{\boldsymbol{\sigma}}$ and $\pi_{\boldsymbol{\tau}} : \mathsf{IT}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \rightharpoonup \mathsf{IT}_{\boldsymbol{\tau}}$ extracting various data from instantiated transitions, following the characterization of instantiated transitions of $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ given in Figure 28.

Finally, those projection functions are extended to instantiated transitions in context via:

$$\begin{array}{rcrcl} \mathsf{lbl}_{\circledast} & : & \mathsf{ITC}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} & \rightharpoonup & \mathsf{MInt} \\ & & \mathbf{t} \uplus \gamma & \mapsto & \mathsf{lbl}_{\circledast}(\mathbf{t}) \end{array} \qquad \begin{array}{rcrcl} \pi_{\boldsymbol{\sigma}} & : & \mathsf{ITC}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} & \rightharpoonup & \mathsf{ITC}_{\boldsymbol{\sigma}} \\ & & \mathbf{t} \uplus (\gamma +^{\odot} \gamma') & \mapsto & \pi_{\boldsymbol{\sigma}}(\mathbf{t}) \uplus \gamma \end{array}$$

and symmetrically for $\pi_{\boldsymbol{\tau}} : \mathsf{ITC}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \rightharpoonup \mathsf{ITC}_{\boldsymbol{\tau}}$. Using these, from a run $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ we extract:

$$\mathsf{Int}(\rho) = \rho \upharpoonright \mathsf{lbl}_{\circledast} \qquad \rho_{\boldsymbol{\sigma}} = \rho \upharpoonright \pi_{\boldsymbol{\sigma}}\,, \qquad \rho_{\boldsymbol{\tau}} = \rho \upharpoonright \pi_{\boldsymbol{\tau}}\,,$$

$$\frac{t^0(\!|\alpha|\!) : \alpha \longmapsto_\sigma \beta}{\ell^\odot(t^0)(\!|\ell^\odot(\alpha)|\!) : \ell^\odot(\alpha) \longmapsto_{\tau\odot\sigma} \ell^\odot(\beta)} \qquad \frac{t^0(\!|\alpha|\!) : \alpha \longmapsto_\tau \beta}{\mathscr{r}^\odot(t^0)(\!|\mathscr{r}^\odot(\alpha)|\!) : \mathscr{r}^\odot(\alpha) \longmapsto_{\tau\odot\sigma} \mathscr{r}^\odot(\beta)}$$

$$\frac{t^+(\!|\alpha|\!) : \alpha \xmapsto{\ell(m)}_\sigma \emptyset}{\ell^\odot(t^+)(\!|\ell^\odot(\alpha)|\!) : \ell^\odot(\alpha) \xmapsto{\ell(m)}_{\tau\odot\sigma} \emptyset} \qquad \frac{t^+(\!|\alpha|\!) : \alpha \xmapsto{\mathscr{r}(m)}_\tau \emptyset}{\mathscr{r}^\odot(t^+)(\!|\mathscr{r}^\odot(\alpha)|\!) : \mathscr{r}^\odot(\alpha) \xmapsto{\mathscr{r}(m)}_{\tau\odot\sigma} \emptyset}$$

$$\frac{t^-(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \xmapsto{\ell(m)}_\sigma \beta}{\ell^\odot(t^-)(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \xmapsto{\ell(m)}_{\tau\odot\sigma} \ell^\odot(\beta)} \qquad \frac{t^-(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \xmapsto{\mathscr{r}(m)}_\tau \beta}{\mathscr{r}^\odot(t^-)(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \xmapsto{\mathscr{r}(m)}_{\tau\odot\sigma} \mathscr{r}^\odot(\beta)}$$

$$\frac{t^+(\!|\alpha|\!) : \alpha \xmapsto{\mathscr{r}(m)} \emptyset \qquad t^-(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \xmapsto{\ell(m)} \beta}{(t^+ \circledast t^-)(\!|\ell^\odot(\alpha)|\!) : \ell^\odot(\alpha) \longmapsto \mathscr{r}^\odot(\beta)} \qquad \frac{t^-(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \xmapsto{\mathscr{r}(m)} \beta \qquad t^+(\!|\alpha|\!) : \alpha \xmapsto{\ell(m)} \emptyset}{(t^- \circledast t^+)(\!|\mathscr{r}^\odot(\alpha)|\!) : \mathscr{r}^\odot(\alpha) \longmapsto \ell^\odot(\beta)}$$

Fig. 28. Description of instantiated transitions of $\tau \odot \sigma$

		$\mathsf{lbl}_\circledast$	π_σ	π_τ
$\ell^\odot(t^0)(\!\|\ell^\odot(\alpha)\|\!)$	$\mapsto$		$t^0(\!\|\alpha\|\!)$	
$\mathscr{r}^\odot(t^0)(\!\|\mathscr{r}^\odot(\alpha)\|\!)$	$\mapsto$			$t^0(\!\|\alpha\|\!)$
$\ell^\odot(t^+)(\!\|\ell^\odot(\alpha)\|\!)$	$\mapsto$	$\ell(m)$	$t^+(\!\|\alpha\|\!)$	
$\mathscr{r}^\odot(t^+)(\!\|\mathscr{r}^\odot(\alpha)\|\!)$	$\mapsto$	$\mathscr{r}(m)$		$t^+(\!\|\alpha\|\!)$
$\ell^\odot(t^-)(\!\|(\mathrm{s},\mathrm{d})\|\!)$	$\mapsto$	$\ell(m)$	$t^-(\!\|(\mathrm{s},\mathrm{d})\|\!)$	
$\mathscr{r}^\odot(t^-)(\!\|(\mathrm{s},\mathrm{d})\|\!)$	$\mapsto$	$\mathscr{r}(m)$		$t^-(\!\|(\mathrm{s},\mathrm{d})\|\!)$
$(t^+ \circledast t^-)(\!\|\ell^\odot(\alpha)\|\!)$	$\mapsto$	$\mathscr{m}(m)$	$t^+(\!\|\alpha\|\!)$	$t^-(\!\|(\mathrm{s},\mathrm{d})\|\!)$
$(t^- \circledast t^+)(\!\|\mathscr{r}^\odot(\alpha)\|\!)$	$\mapsto$	$\mathscr{m}(m)$	$t^-(\!\|(\mathrm{s},\mathrm{d})\|\!)$	$t^+(\!\|\alpha\|\!)$

Fig. 29. Projections of instantiated transitions

which allow us to prove the following property:

Lemma C.4. *Consider $\rho : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \alpha$. Then, $\alpha = \alpha_\sigma +^\odot \alpha_\tau$ and*

$$\rho_\sigma : \emptyset \twoheadrightarrow \alpha_\sigma\,, \qquad\qquad \rho_\tau : \emptyset \twoheadrightarrow \alpha_\tau$$

where $\mathsf{play}(\rho) = \mathsf{Int}(\rho)_{\tau\odot\sigma}$, $\mathsf{play}(\rho_\sigma) = \mathsf{Int}(\rho)_\sigma$ *and* $\mathsf{play}(\rho_\tau) = \mathsf{Int}(\rho)_\tau$.
Moreover, $\mathsf{Coll}(\rho) = \mathsf{Coll}(\rho_\sigma) +^\odot \mathsf{Coll}(\rho_\tau)$.

Proof. A lengthy and grindy induction on ρ. For ρ empty this is clear, otherwise we reason by cases on the last instantiated transition in context of ρ, following the Figure 28.

Consider first that we have $\rho' = \rho(\ell^\odot(t^0)(\!|\ell^\odot(\mu)|\!) \uplus \gamma) : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \beta$ where $\rho : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \alpha$,

$$\ell^\odot(t^0)(\!|\ell^\odot(\alpha)|\!) : \ell^\odot(\mu) \longmapsto_{\tau\odot\sigma} \ell^\odot(\nu)\,, \qquad \ell^\odot(t^0)(\!|\ell^\odot(\alpha)|\!) \uplus \gamma : \alpha \longrightarrow_{\tau\odot\sigma} \beta$$

with necessarily $\alpha = \ell^\odot(\mu) \uplus \gamma$ and $\beta = \ell^\odot(\nu) \uplus \gamma$, and $t^0(\!|\mu|\!) : \mu \longmapsto_\sigma \nu$. By IH, $\alpha = \alpha_\sigma +^\odot \alpha_\tau$ with

$$\rho_\sigma : \emptyset \twoheadrightarrow_\sigma \alpha_\sigma\,, \qquad \rho_\tau : \emptyset \twoheadrightarrow_\tau \alpha_\tau\,,$$

so in particular that entails that $\gamma = \gamma_\sigma +^\odot \gamma_\tau$ with $\alpha_\sigma = \mu \uplus \gamma_\sigma$. Now, we have

$$t^0(\!|\mu|\!) \uplus \gamma_\sigma : \alpha_\sigma \longrightarrow_\sigma \beta_\sigma$$

writing $\beta_\sigma = \nu \uplus \gamma_\sigma$. Writing $\beta_\tau = \alpha_\tau = \gamma_\tau$, we have

$$(\rho(\ell^\odot(t^0)(\!|\ell^\odot(\mu)|\!) \uplus \gamma))_\sigma = \rho_\sigma(t^0(\!|\mu|\!) \uplus \gamma_\sigma) : \emptyset \twoheadrightarrow_\sigma \beta_\sigma$$

and $(\rho(\ell^{\odot}(t^{0})(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma))_{\tau} = \rho_{\tau} : \emptyset \twoheadrightarrow_{\tau} \beta_{\tau}$. Moreover $\mathsf{play}(\rho') = \mathsf{play}(\rho)$, $\mathsf{Int}(\rho') = \mathsf{Int}(\rho)$, $\mathsf{play}(\rho'_{\sigma}) = \mathsf{play}(\rho_{\sigma})$ and $\mathsf{play}(\rho'_{\tau}) = \mathsf{play}(\rho_{\tau})$, making required properties obvious. Finally, $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho) \cup \ell^{\odot}(\nu)$, $\mathsf{Coll}(\rho'_{\sigma}) = \mathsf{Coll}(\rho_{\sigma}) \cup \nu$ and $\mathsf{Coll}(\rho'_{\tau}) = \mathsf{Coll}(\rho_{\tau})$, so $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho'_{\sigma}) +^{\odot} \mathsf{Coll}(\rho'_{\tau})$ follows from IH. The case of a neutral transition from $\boldsymbol{\tau}$ is symmetric.

Next, consider that $\rho' = \rho(\ell^{\odot}(t^{+})(\!|\ell^{\odot}(\alpha)|\!) \uplus \gamma) : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \beta$, where $\rho : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \alpha$,

$$\ell^{\odot}(t^{+})(\!|\ell^{\odot}(\mu)|\!) : \ell^{\odot}(\mu) \overset{\ell_{\vdash}(m)}{\longmapsto}_{\tau\odot\sigma} \emptyset\,, \qquad \ell^{\odot}(t^{+})(\!|\ell^{\odot}(\alpha)|\!) \uplus \gamma : \alpha \longrightarrow_{\tau\odot\sigma} \beta$$

where necessarily $\alpha = \ell^{\odot}(\mu) \uplus \gamma$ and $\beta = \gamma$, and $t^{+}(\!|\mu|\!) : \mu \overset{\ell_{\vdash}(m)}{\longmapsto}_{\sigma} \emptyset$. Now, by IH, $\alpha = \alpha_{\sigma} +^{\odot} \alpha_{\tau}$ with

$$\rho_{\sigma} : \emptyset \twoheadrightarrow_{\sigma} \alpha_{\sigma}\,, \qquad \rho_{\tau} : \emptyset \twoheadrightarrow_{\tau} \alpha_{\tau}\,,$$

so in particular that entails $\gamma = \gamma_{\sigma} +^{\odot} \gamma_{\tau}$ with $\alpha_{\sigma} = \mu \uplus \gamma_{\sigma}$. Now, we have

$$t^{+}(\!|\mu|\!) \uplus \gamma_{\sigma} : \alpha_{\sigma} \longrightarrow_{\sigma} \beta_{\sigma}$$

writing $\beta_{\sigma} = \gamma_{\sigma}$. Writing $\beta_{\tau} = \alpha_{\tau} = \gamma_{\tau}$, we have

$$(\rho(\ell^{\odot}(t^{+})(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma))_{\sigma} = \rho_{\sigma}(t^{+}(\!|\mu|\!) \uplus \gamma_{\sigma}) : \emptyset \twoheadrightarrow_{\sigma} \beta_{\sigma}$$

and $(\rho(\ell^{\odot}(t^{+})(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma))_{\tau} = \rho_{\tau} : \emptyset \twoheadrightarrow_{\tau} \beta_{\tau}$. Moreover, $\mathsf{play}(\rho') = \mathsf{play}(\rho)\ell_{\vdash}(m)$, $\mathsf{Int}(\rho') = \mathsf{Int}(\rho)\ell(m)$, $\mathsf{play}(\rho'_{\sigma}) = \mathsf{play}(\rho_{\sigma})\ell_{\vdash}(m)$ and $\mathsf{play}(\rho'_{\tau}) = \mathsf{play}(\rho_{\tau})$, making the required properties clear. Finally, $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho), \mathsf{Coll}(\rho'_{\sigma}) = \mathsf{Coll}(\rho_{\sigma})$ and $\mathsf{Coll}(\rho'_{\tau}) = \mathsf{Coll}(\rho_{\tau})$, so $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho'_{\sigma}) +^{\odot} \mathsf{Coll}(\rho'_{\tau})$ follows from IH. The case of a positive transition from $\boldsymbol{\tau}$ is symmetric.

Next, consider that we have $\rho' = \rho(\mathscr{r}^{\odot}(t^{-})(\!|(\mathrm{s},\mathrm{d})|\!) \uplus \gamma) : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \beta$, where $\rho : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \alpha$,

$$\ell^{\odot}(t^{-})(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto} \ell^{\odot}(\nu)\,, \qquad \ell^{\odot}(t^{-})(\!|(\mathrm{s},\mathrm{d})|\!) \uplus \gamma : \alpha \longrightarrow_{\tau\odot\sigma} \beta$$

where necessarily, $\alpha = \gamma$ and $\beta = \ell^{\odot}(\nu) \uplus \gamma$, and $t^{-}(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto}_{\sigma} \nu$. By IH, $\alpha = \alpha_{\sigma} +^{\odot} \alpha_{\tau}$ with

$$\rho_{\sigma} : \emptyset \twoheadrightarrow_{\sigma} \alpha_{\sigma}\,, \qquad \rho_{\tau} : \emptyset \twoheadrightarrow_{\tau} \alpha_{\tau}\,,$$

so that $\gamma = \gamma_{\sigma} +^{\odot} \gamma_{\tau}$ with $\gamma_{\sigma} = \alpha_{\sigma}, \gamma_{\tau} = \alpha_{\tau}$. Writing $\beta_{\sigma} = \gamma_{\sigma} \uplus \nu$ and $\beta_{\tau} = \gamma_{\tau} = \alpha_{\tau}$, we have

$$(\rho(\mathscr{r}^{\odot}(t^{-})(\!|(\mathrm{s},\mathrm{d})|\!) \uplus \gamma))_{\sigma} = \rho_{\sigma}(t^{-}(\!|(\mathrm{s},\mathrm{d})|\!) \uplus \gamma_{\sigma}) : \emptyset \twoheadrightarrow_{\sigma} \beta_{\sigma}$$

and $(\rho(\mathscr{r}^{\odot}(t^{-})(\!|(\mathrm{s},\mathrm{d})|\!) \uplus \gamma))_{\tau} = \rho_{\tau} : \emptyset \twoheadrightarrow_{\tau} \beta_{\tau}$. Moreover, $\mathsf{play}(\rho') = \mathsf{play}(\rho)\ell_{\vdash}(m)$, $\mathsf{Int}(\rho') = \mathsf{Int}(\rho)\ell(m)$, $\mathsf{play}(\rho'_{\sigma}) = \mathsf{play}(\rho_{\sigma})\ell_{\vdash}(m)$ and $\mathsf{play}(\rho'_{\tau}) = \mathsf{play}(\rho_{\tau})$, making the required properties clear. Finally, $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho) \cup \ell^{\odot}(\nu)$, with $\mathsf{Coll}(\rho'_{\sigma}) = \mathsf{Coll}(\rho_{\sigma}) \cup \nu$ and $\mathsf{Coll}(\rho'_{\tau}) = \mathsf{Coll}(\rho_{\tau})$, so $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho'_{\sigma}) +^{\odot} \mathsf{Coll}(\rho'_{\tau})$ follows from IH. Negative transitions from $\boldsymbol{\tau}$ are symmetric.

Consider finally $\rho' = \rho((t^{+} \circledast t^{-})(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \beta$ where $\rho : \emptyset \twoheadrightarrow_{\tau\odot\sigma} \alpha$,

$$(t^{+} \circledast t^{-})(\!|\ell^{\odot}(\mu)|\!) : \ell^{\odot}(\mu) \longmapsto_{\tau\odot\sigma} \mathscr{r}^{\odot}(\nu)\,, \qquad (t^{+} \circledast t^{-})(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma : \alpha \longrightarrow_{\tau\odot\sigma} \beta$$

where necessarily, $\alpha = \ell^{\odot}(\mu) \uplus \gamma$, $\beta = \mathscr{r}^{\odot}(\nu) \uplus \gamma$, and where we necessarily have

$$t^{+}(\!|\mu|\!) : \mu \overset{\mathscr{r}_{\vdash}(m)}{\longmapsto}_{\sigma} \emptyset\,, \qquad t^{-}(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto}_{\tau} \nu\,,$$

with $\delta_{\sigma}\langle t^{+}\rangle(\mu) = (\mathrm{s},\mathrm{d})$, $\ell_{\vdash}(m) = (\ell_{\vdash}(\mathrm{m}), \mathrm{s}, \mathrm{d})$ where $\ell_{\vdash}(\mathrm{m}) = \partial_{\sigma}(t^{+})$; and $\delta_{\tau}\langle t^{-}\rangle(\mathrm{s},\mathrm{d}) = \nu$. Now,

$$\rho_{\sigma} : \emptyset \twoheadrightarrow_{\sigma} \alpha_{\sigma}\,, \qquad \rho_{\tau} : \emptyset \twoheadrightarrow_{\tau} \alpha_{\tau}$$

for $\alpha = \alpha_{\sigma} +^{\odot} \alpha_{\tau}$ by IH, and necessarily $\gamma = \gamma_{\sigma} +^{\odot} \gamma_{\tau}$ with $\alpha_{\sigma} = \mu \uplus \gamma_{\sigma}$ and $\alpha_{\tau} = \gamma_{\tau}$. Writing $\beta_{\sigma} = \gamma_{\sigma}$ and $\beta_{\tau} = \gamma_{\tau} \uplus \nu$, we have $\beta = \beta_{\sigma} \uplus \beta_{\tau}$. Hence, we can form the projected runs

$$\rho'_{\sigma} = \rho_{\sigma}(t^{+}(\!|\mu|\!) \uplus \gamma_{\sigma}) : \emptyset \twoheadrightarrow_{\sigma} \beta_{\sigma}\,, \qquad \rho'_{\tau} = \rho_{\tau}(t^{-}(\!|(\mathrm{s},\mathrm{d})|\!) \uplus \gamma_{\tau}) : \emptyset \twoheadrightarrow_{\tau} \beta_{\tau}\,,$$

satisfying $\mathsf{play}(\rho') = \mathsf{play}(\rho)$, $\mathsf{Int}(\rho') = \mathsf{Int}(\rho)\mathscr{m}(m)$, $\mathsf{play}(\rho'_{\sigma}) = \mathsf{play}(\rho_{\sigma})\mathscr{r}_{\vdash}(m)$ and $\mathsf{play}(\rho'_{\tau}) = \mathsf{play}(\rho_{\tau})\ell_{\vdash}(m)$ from which the required verifications are immediate. Finally, $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho) \cup \mathscr{r}^{\odot}(\nu)$ with $\mathsf{Coll}(\rho'_{\sigma}) = \mathsf{Coll}(\rho_{\sigma})$ and $\mathsf{Coll}(\rho'_{\tau}) = \mathsf{Coll}(\rho_{\tau}) \cup \nu$, so $\mathsf{Coll}(\rho') = \mathsf{Coll}(\rho'_{\sigma}) +^{\odot} \mathsf{Coll}(\rho'_{\tau})$ follows from IH. The case $t^{-} \circledast t^{+}$ is symmetric, concluding the proof. $\square$

Using this lemma, we shall now prove that a valid run of $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$ projects to valid runs on $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$. For this we will also exploit the following easy lemma.

Lemma C.5. *Consider A, B arenas, and $s \in |A \vdash B|^*$.*
Then, $s \in \mathsf{Plays}(A \vdash B)$ iff $s \upharpoonright \ell_{\vdash}^* \in \mathsf{Plays}(A)$ and $s \upharpoonright r_{\vdash}^* \in \mathsf{Plays}(B)$.

Proof. Straightforward. □

Lemma C.6. *Consider $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ such that $\mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash C)$.*
Then, $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$ and $\mathsf{play}(\rho_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C)$.

Proof. By induction on ρ. For ρ empty this is clear.
Consider first that we have $\rho' = \rho(\ell^{\odot}(t^0)(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma) : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$ where $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$,

$$\ell^{\odot}(t^0)(\!|\ell^{\odot}(\alpha)|\!) : \ell^{\odot}(\mu) \longmapsto_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \ell^{\odot}(\nu)\,, \qquad \ell^{\odot}(t^0)(\!|\ell^{\odot}(\alpha)|\!) \uplus \gamma : \alpha \longrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta\,,$$

in that case $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) = \mathsf{play}(\rho_{\boldsymbol{\sigma}})$ and $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) = \mathsf{play}(\rho_{\boldsymbol{\tau}})$, so the property follows from IH. The case of a neutral transition from $\boldsymbol{\tau}$ is symmetric.
Consider next that $\rho' = \rho(\ell^{\odot}(t^+)(\!|\ell^{\odot}(\alpha)|\!) \uplus \gamma) : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$, where $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$,

$$\ell^{\odot}(t^+)(\!|\ell^{\odot}(\mu)|\!) : \ell^{\odot}(\mu) \overset{\ell_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \emptyset\,, \qquad \ell^{\odot}(t^+)(\!|\ell^{\odot}(\alpha)|\!) \uplus \gamma : \alpha \longrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$$

where necessarily $\alpha = \ell^{\odot}(\mu) \uplus \gamma$ and $\beta = \gamma$, and $t^+(\!|\mu|\!) : \mu \overset{\ell_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\sigma}} \emptyset$. By IH, we have

$$\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)\,, \qquad \mathsf{play}(\rho_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C)$$

and as $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) = \mathsf{play}(\rho_{\boldsymbol{\tau}})$, we have $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C)$ as required. Now, we have

$$\rho_{\boldsymbol{\sigma}} : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}}\,, \qquad t^+(\!|\mu|\!) \uplus \gamma_{\boldsymbol{\sigma}} : \alpha_{\boldsymbol{\sigma}} \overset{\ell_{\vdash}(m)}{\longrightarrow}_{\boldsymbol{\sigma}} \beta_{\boldsymbol{\sigma}}$$

with components named as in the proof of Lemma C.4, and with $s = \mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$. Hence by condition *valid* of Petri strategies, $s\ell_{\vdash}(m) = \mathsf{play}(\rho'_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$, which concludes this case. The case of a positive transition from $\boldsymbol{\tau}$ is symmetric.
Consider next that $\rho' = \rho(r^{\odot}(t^-)(\!|(\mathrm{s}, \mathrm{d})|\!) \uplus \gamma) : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$, where $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$,

$$\ell^{\odot}(t^-)(\!|(\mathrm{s}, \mathrm{d})|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto} \ell^{\odot}(\nu)\,, \qquad \ell^{\odot}(t^-)(\!|(\mathrm{s}, \mathrm{d})|\!) \uplus \gamma : \alpha \longrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta\,,$$

in that case $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) = \mathsf{play}(\boldsymbol{\sigma})\ell_{\vdash}(m)^-$ and $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) = \mathsf{play}(\boldsymbol{\tau})$. By IH we have $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$ and $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) = \mathsf{play}(\rho_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C)$. But by hypothesis, we have $\mathsf{play}(\rho') = \mathsf{play}(\rho)\ell_{\vdash}(m) \in \mathsf{Plays}(A \vdash C)$. By Lemma C.5, $\mathsf{play}(\rho)\ell_{\vdash}(m) \upharpoonright \ell_{\vdash}^* = (\mathsf{play}(\rho) \upharpoonright \ell_{\vdash}^*)m \in \mathsf{Plays}(A)$. But $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) \upharpoonright \ell_{\vdash}^* = (\mathsf{play}(\rho) \upharpoonright \ell_{\vdash}^*)m \in \mathsf{Plays}(A)$, and $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) \upharpoonright r_{\vdash}^* = \mathsf{play}(\rho_{\boldsymbol{\sigma}}) \upharpoonright r_{\vdash}^*$, so $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$ by Lemma C.5. The case of a negative transition from $\boldsymbol{\tau}$ is symmetric.
Consider finally $\rho' = \rho((t^+ \circledast t^-)(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$ where $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$,

$$(t^+ \circledast t^-)(\!|\ell^{\odot}(\mu)|\!) : \ell^{\odot}(\mu) \longmapsto_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} r^{\odot}(\nu)\,, \qquad (t^+ \circledast t^-)(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma : \alpha \longrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$$

where necessarily, $\alpha = \ell^{\odot}(\mu) \uplus \gamma$, $\beta = r^{\odot}(\nu) \uplus \gamma$, and where we necessarily have

$$t^+(\!|\mu|\!) : \mu \overset{r_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\sigma}} \emptyset\,, \qquad t^-(\!|(\mathrm{s}, \mathrm{d})|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\tau}} \nu\,,$$

with $\delta_{\boldsymbol{\sigma}}\langle t^+\rangle(\mu) = (\mathrm{s}, \mathrm{d})$, $\ell_{\vdash}(m) = (\ell_{\vdash}(\mathrm{m}), \mathrm{s}, \mathrm{d})$ where $\ell_{\vdash}(\mathrm{m}) = \partial_{\boldsymbol{\sigma}}(t^+)$; and $\delta_{\boldsymbol{\tau}}\langle t^-\rangle(\mathrm{s}, \mathrm{d}) = \nu$. By IH,

$$\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)\,, \qquad \mathsf{play}(\rho_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C)\,.$$

Summing up the situation on the side of $\boldsymbol{\sigma}$, we have

$$\rho_{\boldsymbol{\sigma}} : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}}\,, \qquad t^+(\!|\mu|\!) \uplus \gamma_{\boldsymbol{\sigma}} : \alpha_{\boldsymbol{\sigma}} \overset{r_{\vdash}(m)}{\longrightarrow}_{\boldsymbol{\sigma}} \beta_{\boldsymbol{\sigma}}$$

with components as in the proof of Lemma C.4. By condition *valid* of Petri strategies, $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) = \mathsf{play}(\rho_{\boldsymbol{\sigma}})r_{\vdash}(m) \in \mathsf{Plays}(A \vdash B)$. By Lemma C.5, this entails $\mathsf{play}(\rho_{\boldsymbol{\sigma}})r_{\vdash}(m) \upharpoonright r_{\vdash}^* = (\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \upharpoonright r_{\vdash}^*)m \in \mathsf{Plays}(B)$. Now, Lemma C.4 also entails $\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) = \mathsf{Int}(\rho)_{\boldsymbol{\sigma}}$ and $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) = \mathsf{Int}(\rho)_{\boldsymbol{\tau}}$. So:

$$\mathsf{play}(\rho'_{\boldsymbol{\sigma}}) \upharpoonright r_{\vdash}^* = \mathsf{Int}(\rho) \upharpoonright \mathscr{m}^* = \mathsf{play}(\rho'_{\boldsymbol{\tau}}) \upharpoonright \ell_{\vdash}^*,$$

so that $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) \upharpoonright \ell_{\vdash}^* = (\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \upharpoonright \ell_{\vdash}^*)m \in \mathsf{Plays}(B)$. Now, we also have $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) \upharpoonright r_{\vdash}^* = \mathsf{play}(\rho_{\boldsymbol{\tau}}) \upharpoonright r_{\vdash}^* \in \mathsf{Plays}(C)$ by Lemma C.5; so by Lemma C.5 again we deduce $\mathsf{play}(\rho'_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C)$ as required. The case of a synchronized transition $t^- \circledast t^+$ is symmetric. □

We are finally in position to prove that Petri strategies are stable under composition.

Proposition C.7. *If $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ are Petri strategies, then so is $\boldsymbol{\tau} \odot \boldsymbol{\sigma} : A \vdash C$. Moreover, if $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are negative, so is $\boldsymbol{\tau} \odot \boldsymbol{\sigma}$.*

Proof. *Valid.* Consider $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ with $\mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash C)$, and $\mathsf{t}^+ : \alpha \xrightarrow{m'} \beta$. *W.l.o.g.*, assume $m' = \ell_{\vdash}(m)$. By Lemma C.4, α decomposes as $\alpha_{\boldsymbol{\sigma}} +^{\odot} \alpha_{\boldsymbol{\tau}}$, and we have:

$$\rho_{\boldsymbol{\sigma}} : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}}, \qquad \rho_{\boldsymbol{\tau}} : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}} \alpha_{\boldsymbol{\tau}}.$$

Now, t^+ must have the form $\mathsf{t}^+ = (\ell^{\odot}(t^+)(\!|\ell^{\odot}(\mu)|\!)) \uplus \gamma$ with $t^+(\!|\mu|\!) : \mu \overset{\ell_{\vdash}(m)}{\longmapsto} \emptyset$, and $\beta = \gamma$. Hence,

$$t^+(\!|\mu|\!) \uplus \gamma_{\boldsymbol{\sigma}} : \alpha_{\boldsymbol{\sigma}} \xrightarrow{\ell_{\vdash}(m)} \beta_{\boldsymbol{\sigma}},$$

with components named as in the proof of Lemma C.4. At this point we apply Lemma C.6, which ensures that $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$. Hence since $\boldsymbol{\sigma}$ is *valid*, $\mathsf{play}(\rho_{\boldsymbol{\sigma}})\ell_{\vdash}(m) \in \mathsf{Plays}(A \vdash B)$ as well. It follows by Lemma C.5 that $\mathsf{play}(\rho') = \mathsf{play}(\rho)\ell_{\vdash}(m) \in \mathsf{Plays}(A \vdash C)$ as well.

Receptive. Consider $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ with $s = \mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash C)$, and $sm' \in \mathsf{Plays}(A \vdash C)$ with m' negative – say *w.l.o.g.* that $m' = \ell_{\vdash}(m)^-$. By Lemma C.6, we have

$$\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B), \qquad \mathsf{play}(\rho_{\boldsymbol{\tau}}) \in \mathsf{Plays}(B \vdash C),$$

and by Lemma C.6 we may deduce that $\mathsf{play}(\rho_{\boldsymbol{\sigma}})\ell_{\vdash}(m) \in \mathsf{Plays}(A \vdash B)$. By *receptive*, there is

$$\mathsf{t}^- = t^-(\!|(\mathsf{s}, \mathsf{d})|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\sigma}} \beta$$

for $\beta \cap \alpha_{\boldsymbol{\sigma}} = \emptyset$. Hence $\ell^{\odot}(t^-)(\!|(\mathsf{s}, \mathsf{d})|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \ell^{\odot}(\beta)$ with $\ell^{\odot}(\beta) \cap \alpha = \emptyset$. For uniqueness, if

$$\mathsf{t}' : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta'$$

for some β'. Necessarily, $\mathsf{t}' = \ell^{\odot}(t')(\!|(\mathsf{s}', \mathsf{d}')|\!)$ for $t'(\!|(\mathsf{s}', \mathsf{d}')|\!) : \emptyset \overset{\ell_{\vdash}(m)}{\longmapsto}_{\boldsymbol{\sigma}} \beta''$ for $\beta' = \ell^{\odot}(\beta'')$. But by uniqueness of *receptivity* for $\boldsymbol{\sigma}$, we have $t'(\!|(\mathsf{s}', \mathsf{d}')|\!) = t(\!|(\mathsf{s}, \mathsf{d})|\!)$, so that $t = t'$, $(\mathsf{s}, \mathsf{d}) = (\mathsf{s}', \mathsf{d}')$.

Strongly safe. Consider $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ with $\mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash C)$, with a new instantiated transition in context t – we distinguish cases depending on its form.

Assume first $\mathsf{t} = \ell^{\odot}(t^0)(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma$ for $t^0(\!|\mu|\!) : \mu \longmapsto_{\boldsymbol{\sigma}} \nu$. By Lemma C.6, we have $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$, so that since $\boldsymbol{\sigma}$ is *strongly safe*, $\mathsf{new}(t^0(\!|\mu|\!))$ is fresh in $\rho_{\boldsymbol{\sigma}}$. But $\mathsf{new}(\ell^{\odot}(t^0)(\!|\ell^{\odot}(\mu)|\!)) = \ell^{\odot}(\mathsf{new}(t^0(\!|\mu|\!)))$. Moreover, by Lemma C.4, $\mathsf{Coll}(\rho) = \mathsf{Coll}(\rho_{\boldsymbol{\sigma}}) +^{\odot} \mathsf{Coll}(\rho_{\boldsymbol{\tau}})$. So it follows that $\mathsf{new}(\mathsf{t})$ is fresh in ρ as required. The case of a neutral transition from $\boldsymbol{\tau}$ is symmetric.

Next assume $\mathsf{t} = \ell^{\odot}(t^+)(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma$. Then $\mathsf{new}(\mathsf{t}) = \emptyset$. Idem for a positive transition from $\boldsymbol{\tau}$.

Next assume $\mathsf{t} = \ell^{\odot}(t^-)(\!|(\mathsf{s}, \mathsf{d})|\!) \uplus \gamma : \alpha \xrightarrow{\ell_{\vdash}(m)}_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$ with $\mathsf{play}(\rho)\ell_{\vdash}(m) \in \mathsf{Plays}(A \vdash C)$. We have

$$t^-(\!|(\mathsf{s}, \mathsf{d})|\!) : \emptyset \overset{(\ell_{\vdash}(m))}{\longmapsto} \nu,$$

and by Lemma C.6, we have $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$. From $\mathsf{play}(\rho)\ell_{\vdash}(m) \in \mathsf{Plays}(A \vdash C)$ and $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A \vdash B)$, it follows easily via Lemma C.5 that $\mathsf{play}(\rho_{\boldsymbol{\sigma}})\ell_{\vdash}(m) \in \mathsf{Plays}(A \vdash B)$. So since $\boldsymbol{\sigma}$ is *strongly safe*, $\mathsf{new}(t^-(\!|(\mathsf{s}, \mathsf{d})|\!))$ is fresh in $\rho_{\boldsymbol{\sigma}}$. We conclude as in the neutral case using Lemma C.4. The case of a negative transition from $\boldsymbol{\tau}$ is similar.

Finally, assume $\mathsf{t} = (t^+ \circledast t^-)(\!|\ell^{\odot}(\mu)|\!) \uplus \gamma$. Say that we have

$$(t^+ \circledast t^-)(\!|\ell^{\odot}(\mu)|\!) : \ell^{\odot}(\mu) \longmapsto_{\boldsymbol{\tau} \odot \boldsymbol{\sigma}} r^{\odot}(\nu)\,,$$

so that $\mathsf{new}(\mathsf{t}) = r^{\odot}(\nu)$ – assume $t^+(\!|\mu|\!) : \mu \xmapsto{r_{\vdash}(m)}_{\boldsymbol{\sigma}} \emptyset$ and $t^-(\!|(\mathsf{s}, \mathsf{d})|\!) : \emptyset \xmapsto{\ell_{\vdash}(m)}_{\boldsymbol{\tau}} \nu$. By Lemma C.5, $\mathsf{play}(\rho_{\boldsymbol{\tau}})\ell_{\vdash}(m) \in \mathsf{Plays}(B \vdash C)$. Therefore since $\boldsymbol{\tau}$ is *strongly safe*, $\mathsf{new}(t^-(\!|(\mathsf{s}, \mathsf{d})|\!)) = \nu$ is fresh in $\rho_{\boldsymbol{\tau}}$. But by Lemma C.4, $\mathsf{Coll}(\rho) = \mathsf{Coll}(\rho_{\boldsymbol{\sigma}}) +^{\odot} \mathsf{Coll}(\rho_{\boldsymbol{\tau}})$, so $\mathsf{new}(\mathsf{t}) = r^{\odot}(\nu)$ is fresh in ρ as required. The other synchronized case is symmetric.

Negative. Straightforward by inspection and negativity of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$. □

C.1.2 Copycat. First, we characterize the markings of copycat reachable through a play:

Recall that $\mathcal{L}_{\boldsymbol{\alpha}_A} = \mathsf{mult}(A)$, so that the set $\mathsf{TokIL}(\boldsymbol{\alpha}_A)$ of tokils of $\boldsymbol{\alpha}_A$ is in bijection with $|A|$. This lets us silently coerce a configuration $x \in \mathscr{C}(A)$ into a marking $x \in \mathscr{M}(\boldsymbol{\alpha}_A)$. In order to show that copycat is a Petri strategy, we first characterize the markings reachable by rule-abiding runs:

Lemma C.8. *Consider A an arena, and $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\alpha}_A} \alpha$ such that $\mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash A)$. Then, $|\mathsf{play}(\rho)| = x \vdash y$ with $x, y \in \mathscr{C}(A)$ and $y \supseteq^- x \cap y \subseteq^+ x$; and $\alpha = (y^- \setminus x) \uplus (x^+ \setminus y)$. Moreover, $\mathsf{Coll}(\rho) = y^- \uplus x^+$, where x^p is the subset of $x \in \mathscr{C}(A)$ whose moves have polarity p.*

Proof. By induction on ρ. For ρ empty, this is clear. Consider $\rho' = \rho\mathsf{t} : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\alpha}_A} \beta$ with $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\alpha}_A} \alpha$, and $\mathsf{play}(\rho') \in \mathsf{Plays}(A \vdash A)$. We also have $\mathsf{play}(\rho) \in \mathsf{Plays}(A \vdash A)$, so by IH,

$$|\mathsf{play}(\rho)| = x \vdash y\,, \qquad y \supseteq^- x \cap y \subseteq^+ x\,, \qquad \alpha = (y^- \setminus x) \uplus (x^+ \setminus y)\,,$$

we reason by cases depending on t. If $\mathsf{t} = (\mathsf{m}^+, r)(\!|\{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}|\!) \uplus \gamma$, then $\alpha = \{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\} \uplus \gamma$, and

$$\begin{array}{rcl} |\mathsf{play}(\rho')| & = & x \vdash y \uplus \{(\mathsf{m}, \mathsf{s}, \mathsf{d})\} \\ \beta & = & \alpha \setminus \{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}\,; \end{array}$$

since $\alpha = (y^- \setminus x) \uplus (x^+ \setminus y)$, $(\mathsf{s}, \mathsf{d})^{@\mathsf{m}} \in \alpha$ and $(\mathsf{m}, \mathsf{s}, \mathsf{d})$ is positive, $(\mathsf{m}, \mathsf{s}, \mathsf{d})$ is positive and must be in x. It follows that the invariant is preserved. The case $\mathsf{t} = (\mathsf{m}^-, \ell)(\!|(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}|\!) \uplus \gamma$ is symmetric.

If $\mathsf{t} = (\mathsf{m}^-, r)(\!|(\mathsf{s}, \mathsf{d})|\!) \uplus \gamma$, then $\alpha = \gamma$, and

$$\begin{array}{rcl} |\mathsf{play}(\rho')| & = & x \vdash (y \uplus \{(\mathsf{m}, \mathsf{s}, \mathsf{d})\}) \\ \beta & = & \alpha \uplus \{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}\,; \end{array}$$

and $(\mathsf{m}, \mathsf{s}, \mathsf{d})$ is negative. Since $|\mathsf{play}(\rho')| = x \vdash (y \uplus \{(\mathsf{m}, \mathsf{s}, \mathsf{d})\})$, $(\mathsf{m}, \mathsf{s}, \mathsf{d}) \notin y$. So, it cannot be in x. The invariant directly follows. The case $\mathsf{t} = (\mathsf{m}^+, \ell)(\!|(\mathsf{s}, \mathsf{d})@\mathsf{m}|\!) \uplus \gamma$ is symmetric. □

It is a direct application of this lemma to prove that copycat is a Petri strategy:

Proposition C.9. *For any arena A, $\boldsymbol{\alpha}_A : A \vdash A$ is a negative Petri strategy.*

Proof. *Valid.* Consider $s \in \mathsf{Plays}(A \vdash A)$ and

$$\rho : \emptyset \overset{s}{\longrightarrow\!\!\!\!\!\rightarrow} \alpha\,, \qquad\qquad \mathsf{t}^+ : \alpha \overset{m}{\longrightarrow} \beta\,,$$

and say *w.l.o.g.* that $m = r_{\vdash} a$ for $a \in |A|$. This means that $\mathsf{t}^+ = (\mathsf{m}^+, r)(\!|\{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}|\!) \uplus \gamma$, where $\alpha = \gamma \uplus \{(\mathsf{s}, \mathsf{d})^{@\mathsf{m}}\}$, and $m = (r_{\vdash}\mathsf{m}, \mathsf{s}, \mathsf{d})$. We must show that $s(r_{\vdash}\mathsf{m}, \mathsf{s}, \mathsf{d}) \in \mathsf{Plays}(A \vdash A)$.

By Lemma C.8, $\alpha = (y^- \setminus x) \uplus (x^+ \setminus y)$ where $|s| = x \vdash y$. As $\mathsf{m} \in \mathsf{mult}^+(A)$, $(\mathsf{m}, \mathsf{s}, \mathsf{d}) \in x^+ \setminus y$. As $(\mathsf{m}, \mathsf{s}, \mathsf{d}) \notin y$, we have condition *non-repetitive*. Next, we show $|sm| = x \vdash (y \uplus \{a\})$ is *down-closed*. Consider $a' \rightarrowtriangle_A a$. Since A is alternating, a' is negative. Since $x \in \mathscr{C}(A)$ and $a \in x$, we must have $a' \in x$ as well. But by Lemma C.8 we have $y \supseteq^- x \cap y \subseteq^+ x$, so $a' \in y$ as well. Finally, from conditions *locally conflicting*, *alternating* and *negative* of arenas, pairs of events in minimal conflict have the same polarity. Therefore, as $x \cap y \subseteq^- x$ and $x \cap y \subseteq^+ y$, we have $x \cup y$ consistent, so in particular $y \cup \{a\} \in \mathscr{C}(A)$. It follows that $sm \in \mathsf{Plays}(A \vdash A)$ as needed.

$$\frac{t^0 \llparenthesis \alpha \rrparenthesis : \alpha \longmapsto_{\boldsymbol{\sigma}} \beta}{\ell^{\otimes}(t^0)\llparenthesis \ell^{\otimes}(\alpha) \rrparenthesis : \ell^{\otimes}(\alpha) \longmapsto_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \ell^{\otimes}(\beta)} \qquad \frac{t^0 \llparenthesis \alpha \rrparenthesis : \alpha \longmapsto_{\boldsymbol{\tau}} \beta}{r^{\otimes}(t^0)\llparenthesis r^{\otimes}(\alpha) \rrparenthesis : r^{\otimes}(\alpha) \longmapsto_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} r^{\otimes}(\beta)}$$

$$\frac{t^+ \llparenthesis \alpha \rrparenthesis : \alpha \overset{\ell_\vdash m}{\longmapsto}_{\boldsymbol{\sigma}} \emptyset}{\ell^{\otimes}(t^+)\llparenthesis \ell^{\otimes}(\alpha) \rrparenthesis : \ell^{\otimes}(\alpha) \overset{\ell_\vdash \ell_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \emptyset} \qquad \frac{t^+ \llparenthesis \alpha \rrparenthesis : \alpha \overset{\ell_\vdash m}{\longmapsto}_{\boldsymbol{\tau}} \emptyset}{r^{\otimes}(t^+)\llparenthesis r^{\otimes}(\alpha) \rrparenthesis : r^{\otimes}(\alpha) \overset{\ell_\vdash r_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \emptyset}$$

$$\frac{t^+ \llparenthesis \alpha \rrparenthesis : \alpha \overset{r_\vdash m}{\longmapsto}_{\boldsymbol{\sigma}} \emptyset}{\ell^{\otimes}(t^+)\llparenthesis \ell^{\otimes}(\alpha) \rrparenthesis : \ell^{\otimes}(\alpha) \overset{r_\vdash \ell_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \emptyset} \qquad \frac{t^+ \llparenthesis \alpha \rrparenthesis : \alpha \overset{r_\vdash m}{\longmapsto}_{\boldsymbol{\tau}} \emptyset}{r^{\otimes}(t^+)\llparenthesis r^{\otimes}(\alpha) \rrparenthesis : r^{\otimes}(\alpha) \overset{r_\vdash r_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \emptyset}$$

$$\frac{t^- \llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{\ell_\vdash m}{\longmapsto}_{\boldsymbol{\sigma}} \beta}{\ell^{\otimes}(t^-)\llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{\ell_\vdash \ell_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \ell^{\otimes}(\beta)} \qquad \frac{t^- \llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{\ell_\vdash m}{\longmapsto}_{\boldsymbol{\tau}} \beta}{r^{\otimes}(t^-)\llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{\ell_\vdash r_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} r^{\otimes}(\beta)}$$

$$\frac{t^- \llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{r_\vdash m}{\longmapsto}_{\boldsymbol{\sigma}} \beta}{\ell^{\otimes}(t^-)\llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{r_\vdash \ell_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \ell^{\otimes}(\beta)} \qquad \frac{t^- \llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{r_\vdash m}{\longmapsto}_{\boldsymbol{\tau}} \beta}{r^{\otimes}(t^-)\llparenthesis (\mathrm{s},\mathrm{d}) \rrparenthesis : \emptyset \overset{r_\vdash r_\otimes m}{\longmapsto}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} r^{\otimes}(\beta)}$$

Fig. 30. Description of instantiated transitions of $\boldsymbol{\sigma} \otimes \boldsymbol{\tau}$

Receptive. Consider $\rho : \emptyset \longrightarrow\!\!\!\!\rightarrow_{\mathfrak{c}_A} \alpha$ with $\mathsf{play}(\rho) = s \in \mathsf{Plays}(A \vdash A)$ and $sm^- \in \mathsf{Plays}(A \vdash A)$. *W.l.o.g.* consider $m = r_\vdash(a)$. Decompose $a = (\mathrm{m}, \mathrm{s}, \mathrm{d})$ for $\mathrm{m}^- \in \mathsf{mult}(A)$ and token (s, d). By inspection there is a unique matching instantiated transition, namely $(\mathrm{m}^-, r)\llparenthesis (\mathrm{s}, \mathrm{d}) \rrparenthesis : \emptyset \overset{m}{\longmapsto} \{(\mathrm{s}, \mathrm{d})^{@\mathrm{m}}\}$. Moreover, by Lemma C.8 we have $\alpha = (y^- \setminus x) \uplus (x^+ \setminus y)$ where $|s| = x \vdash y$. But as $sm^- \in \mathsf{Plays}(A \vdash A)$, by *non-repetitive* we have $m^- \notin y$. It follows that $(\mathrm{s}, \mathrm{d})^{@\mathrm{m}} \notin \alpha$ as required.

Strongly safe. Consider $\rho : \emptyset \longrightarrow\!\!\!\!\rightarrow_{\mathfrak{c}_A} \alpha$ with $\mathsf{play}(\rho) = s \in \mathsf{Plays}(A \vdash A)$ extended with $\mathfrak{t}$. If $\mathfrak{t}$ is positive, then $\mathsf{new}(\mathfrak{t}) = \emptyset$ and there is nothing to prove. Hence, consider *w.l.o.g.* $\mathfrak{t} = (\mathrm{m}^-, r)\llparenthesis (\mathrm{s}, \mathrm{d}) \rrparenthesis \uplus \alpha$ with $s(r_\vdash(\mathrm{m}), \mathrm{s}, \mathrm{d})^- \in \mathsf{Plays}(A \vdash A)$. Then, $\mathsf{new}(\mathfrak{t}) = \{(\mathrm{s}, \mathrm{d})^{@\mathrm{m}}\}$. Write $|s| = x \vdash y \in \mathscr{C}(A \vdash A)$. By *non-repetitive*, $(\mathrm{m}, \mathrm{s}, \mathrm{d}) \notin y$. By Lemma C.8, $\mathsf{Coll}(\rho) = y^- \uplus x^+$; but as $(\mathrm{m}, \mathrm{s}, \mathrm{d}) \notin y$ and $(\mathrm{m}, \mathrm{s}, \mathrm{d}) \notin x^+$ (for polarity reasons), it follows that $\{(\mathrm{s}, \mathrm{d})^{@\mathrm{m}}\}$ is fresh in ρ.

Negative. Straightforward by inspection. □

We do not detail the clear fact that composition is stable under isomorphism of Petri strategies. Altogether, this concludes the proof of:

Corollary C.10. *There is* PStrat*, a precategory with objects arenas, and morphisms negative Petri strategies up to isomorphism.*

C.2 PStrat as an IPA-Structure: Operations

We examine the operations involved in the IPA-structure, and show preservation of Petri strategies.

C.2.1 *Tensor.*

The preservation of Petri strategies by the tensor operation is a simplification of composition, without the synchronized events.

Lemma C.11. *Consider $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ Petri structures.*

Then, instantiated transitions of $\boldsymbol{\sigma} \otimes \boldsymbol{\tau}$ are exactly as in Figure 30 – in the sense that there is a one-to-one correspondence between instantiated transitions in the premises and in the conclusion.

Using this description, we define in Figure 31 partial functions $\pi^{\otimes}_{\boldsymbol{\sigma}} : \mathsf{IT}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \rightharpoonup \mathsf{IT}_{\boldsymbol{\sigma}}$ and $\pi^{\otimes}_{\boldsymbol{\tau}} : \mathsf{IT}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \rightharpoonup \mathsf{IT}_{\boldsymbol{\tau}}$ extracting various data from instantiated transitions, following the characterization of instantiated transitions of $\boldsymbol{\sigma} \otimes \boldsymbol{\tau}$ given in Figure 30.

		$\pi_{\boldsymbol{\sigma}}$	$\pi_{\boldsymbol{\tau}}$
$\ell^{\otimes}(t^0)(\!\|\ell^{\otimes}(\alpha)\|\!)$	$\mapsto$	$t^0(\!\|\alpha\|\!)$	
$\ell^{\otimes}(t^+)(\!\|\ell^{\otimes}(\alpha)\|\!)$	$\mapsto$	$t^+(\!\|\alpha\|\!)$	
$\ell^{\otimes}(t^-)(\!\|(\mathrm{s},\mathrm{d})\|\!)$	$\mapsto$	$t^-(\!\|(\mathrm{s},\mathrm{d})\|\!)$	
$r^{\otimes}(t^0)(\!\|r^{\otimes}(\alpha)\|\!)$	$\mapsto$		$t^0(\!\|\alpha\|\!)$
$r^{\otimes}(t^+)(\!\|r^{\otimes}(\alpha)\|\!)$	$\mapsto$		$t^+(\!\|\alpha\|\!)$
$r^{\otimes}(t^-)(\!\|(\mathrm{s},\mathrm{d})\|\!)$	$\mapsto$		$t^-(\!\|(\mathrm{s},\mathrm{d})\|\!)$

Fig. 31. Projections of instantiated transitions

Finally, those projection functions are extended to instantiated transitions in context via:

$$\begin{array}{rcrcl} \pi^{\otimes}_{\boldsymbol{\sigma}} & : & \mathsf{ITC}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} & \rightharpoonup & \mathsf{ITC}_{\boldsymbol{\sigma}} \\ & & \mathbf{t} \uplus (\gamma +^{\otimes} \gamma') & \mapsto & \pi^{\otimes}_{\boldsymbol{\sigma}}(\mathbf{t}) \uplus \gamma \end{array} \qquad \begin{array}{rcrcl} \pi^{\otimes}_{\boldsymbol{\tau}} & : & \mathsf{ITC}_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} & \rightharpoonup & \mathsf{ITC}_{\boldsymbol{\tau}} \\ & & \mathbf{t} \uplus (\gamma +^{\otimes} \gamma') & \mapsto & \pi^{\otimes}_{\boldsymbol{\tau}}(\mathbf{t}) \uplus \gamma' \,. \end{array}$$

Using these, from a run $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \alpha$ we extract:

$$\rho_{\boldsymbol{\sigma}} = \rho \upharpoonright \pi^{\otimes}_{\boldsymbol{\sigma}}\,, \qquad \rho_{\boldsymbol{\tau}} = \rho \upharpoonright \pi^{\otimes}_{\boldsymbol{\tau}}\,,$$

we will also use for restriction the partial functions

$$\begin{array}{rcrcl} u & : & \mathsf{Moves} & \rightharpoonup & \mathsf{Moves} \\ & & \ell_{\vdash}\ell_{\otimes}m & \mapsto & \ell_{\vdash}m \\ & & r_{\vdash}\ell_{\otimes}m & \mapsto & r_{\vdash}m \end{array} \qquad \begin{array}{rcrcl} d & : & \mathsf{Moves} & \rightharpoonup & \mathsf{Moves} \\ & & \ell_{\vdash}r_{\otimes}m & \mapsto & \ell_{\vdash}m \\ & & r_{\vdash}r_{\otimes}m & \mapsto & r_{\vdash}m \end{array}$$

which allow us to prove the following property:

Lemma C.12. *Consider $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \alpha$. Then, $\alpha = \alpha_{\boldsymbol{\sigma}} +^{\otimes} \alpha_{\boldsymbol{\tau}}$ and*

$$\rho_{\boldsymbol{\sigma}} : \emptyset \twoheadrightarrow \alpha_{\boldsymbol{\sigma}}\,, \qquad \rho_{\boldsymbol{\tau}} : \emptyset \twoheadrightarrow \alpha_{\boldsymbol{\tau}}\,.$$

where $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) = \mathsf{play}(\rho) \upharpoonright u$ *and* $\mathsf{play}(\rho_{\boldsymbol{\tau}}) = \mathsf{play}(\rho) \upharpoonright d$.
Moreover, $\mathsf{Coll}(\rho) = \mathsf{Coll}(\rho_{\boldsymbol{\sigma}}) +^{\otimes} \mathsf{Coll}(\rho_{\boldsymbol{\tau}})$.

Proof. Exactly as for Lemma C.4, without synchronized transitions. □

Next, as for composition, we observe that these projections preserve valid plays. For that we shall first need the following easy lemma:

Lemma C.13. *Consider A, B arenas, and $s \in |A \otimes B|^*$.*
Then, $s \in \mathsf{Plays}(A \otimes B)$ iff $s \upharpoonright \ell_{\otimes}^* \in \mathsf{Plays}(A)$ and $s \upharpoonright r_{\otimes}^* \in \mathsf{Plays}(B)$.

Proof. Straightforward. □

Using this and Lemma C.12, we show that projections preserve valid runs. Consider $\boldsymbol{\sigma} : A_1 \vdash B_1$ and $\boldsymbol{\tau} : A_2 \vdash B_2$ Petri strategies.

Lemma C.14. *Consider $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \alpha$ such that* $\mathsf{play}(\rho) \in \mathsf{Plays}(A_1 \otimes A_2 \vdash B_1 \otimes B_2)$.
Then, $\mathsf{play}(\rho_{\boldsymbol{\sigma}}) \in \mathsf{Plays}(A_1 \vdash B_1)$ *and* $\mathsf{play}(\rho_{\boldsymbol{\tau}}) \in \mathsf{Plays}(A_2 \vdash B_2)$.

Proof. As for the proof of Lemma C.6 (without synchronization), using condition *valid* of Petri strategies along with Lemma C.13. □

Using Lemmas C.12, C.13, and C.14, we prove as for Proposition C.7:

Proposition C.15. *If $\boldsymbol{\sigma} : A_1 \vdash B_1$ and $\boldsymbol{\tau} : A_2 \vdash B_2$ are Petri strategies, so is $\boldsymbol{\sigma}\otimes\boldsymbol{\tau} : A_1 \otimes B_1 \vdash A_2 \otimes B_2$. Moreover, if $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are negative, so is $\boldsymbol{\sigma} \otimes \boldsymbol{\tau}$.*

C.2.2 Renamings. Before we go on to currying and promotion, we introduce a technical tool useful in ensuring that they preserve Petri strategies.

First, for any game A we write $\mathsf{Plays\_}(A)$ for the set of **negative plays** on A, *i.e.* those $s_1 \dots s_n \in \mathsf{Plays}(A)$ such that $\mathsf{pol}(s_1) = -$. If $f : \mathsf{Moves} \rightharpoonup \mathsf{Moves}$ and $s = s_1 \dots s_n \in \mathsf{Plays}(A)$ such that f is defined on $|A|$, then we write $f(s) = f(s_1) \dots f(s_n)$. In the sequel, we should be particularly interested in such functions on moves that can be decomposed in f and $(g_\mathsf{m})_{\mathsf{m} \in \mathsf{dom}(f)}$ where

$$f : \mathcal{M} \rightharpoonup \mathcal{M}, \qquad\qquad g_\mathsf{m} : \mathsf{Tok} \to \mathsf{Tok},$$

in which case we obtain a partial function between moves set as

$$\begin{array}{rcrcll} [f, (g_\mathsf{m})] & : & \mathsf{Moves} & \rightharpoonup & \mathsf{Moves} & \\ & & (\mathsf{m}, \mathsf{s}, \mathsf{d}) & \mapsto & (f(\mathsf{m}), \mathsf{s}', \mathsf{d}') & \qquad \text{where } (\mathsf{s}', \mathsf{d}') = g_\mathsf{m}(\mathsf{s}, \mathsf{d}). \end{array}$$

Definition C.16. Consider A, B games, $f, (g_\mathsf{m})$ s.t. $[f, (g_\mathsf{m})] : \mathsf{Moves} \rightharpoonup \mathsf{Moves}$ partial injection. We say $h = [f, (g_\mathsf{m})]$ is a **global renaming** from A to B, written $[f, (g_\mathsf{m})] : A \curvearrowright B$, if:

$$\begin{array}{rl} \textit{defined:} & \text{for all } a \in |A|,\ h(a) \text{ defined.} \\ \textit{polarity-preserving:} & \forall a \in |A|,\ \mathsf{pol}(ha) = \mathsf{pol}(a) \\ \textit{validity:} & \forall s \in \mathsf{Plays\_}(A),\ h(s) \in \mathsf{Plays\_}(B) \\ \textit{receptivity:} & \text{for all } s \in \mathsf{Plays\_}(A), \text{ for all } h(s)b^- \in \mathsf{Plays\_}(B), \\ & \text{there exists } sa^- \in \mathsf{Plays\_}(A) \text{ such that } h(a) = b. \\ \textit{courtesy:} & \text{for all } a \rightarrowtriangle_A b, \text{ either } h(a) \rightarrowtriangle_B h(b) \text{ or } (\mathsf{pol}(a), \mathsf{pol}(b)) = (-, +). \end{array}$$

Global renamings are used to transport Petri strategies across games. The following definition, first applied simply on Petri structures, extends Definition 3.14 in that it also renames tokens rather than merely rerouting visible transitions.

Definition C.17. Consider A, B games, $\boldsymbol{\sigma}$ a Petri structure on $\mathsf{mult}(A)$, and $h = [f, (g_\mathsf{m})] : A \curvearrowright B$. We define the **renaming** $\boldsymbol{\sigma}[h]$ on $\mathsf{mult}(B)$, with the same components as $\boldsymbol{\sigma}$, except:

$$\begin{array}{rcll} \partial_{\boldsymbol{\sigma}[h]}(t) & = & f(\partial_{\boldsymbol{\sigma}}(t)) & \\ \delta_{\boldsymbol{\sigma}[h]}\langle t^+ \rangle(\alpha) & = & g_\mathsf{m}(\delta_{\boldsymbol{\sigma}}\langle t^+ \rangle(\alpha)) & \qquad \text{for } \mathsf{m} = \partial_{\boldsymbol{\sigma}}(t^+) \\ \delta_{\boldsymbol{\sigma}[h]}\langle t^- \rangle(g_\mathsf{m}(\mathsf{s}, \mathsf{d})) & = & \delta_{\boldsymbol{\sigma}}\langle t^- \rangle(\mathsf{s}, \mathsf{d}) & \qquad \text{for } \mathsf{m} = \partial_{\boldsymbol{\sigma}}(t^-). \end{array}$$

observing that by hypothesis, g_m is injective for all $\mathsf{m} \in \mathsf{dom}(f)$.

In order to use global renaming to transport Petri strategies, we must transport valid runs. Consider A, B games, $h = [f, (g_\mathsf{m})] : A \curvearrowright B$, and $\boldsymbol{\sigma} : A$ a Petri strategy. Then we set

$$\begin{array}{rcrcll} -[h] & : & \mathsf{IT}_{\boldsymbol{\sigma}} & \to & \mathsf{IT}_{\boldsymbol{\sigma}[h]} & \\ & & t^0\langle \alpha \rangle & \mapsto & t^0\langle \alpha \rangle & \\ & & t^-\langle (\mathsf{s}, \mathsf{d}) \rangle & \mapsto & t^-\langle g_\mathsf{m}(\mathsf{s}, \mathsf{d}) \rangle & \qquad \text{where } \mathsf{m} = \partial_{\boldsymbol{\sigma}}(t^-) \\ & & t^+\langle \alpha \rangle & \mapsto & t^+\langle \alpha \rangle & \end{array}$$

extended to instantiated transitions in context with $(\mathbf{t} \uplus \gamma)[h] = \mathbf{t}[h] \uplus \gamma$. It is immediate from the definition that this substitution leaves pre- and post-conditions of instantiated transitions unchanged, so that it lifts to *runs*: for any $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha$, $\rho[h] : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}[h]} \alpha$ is defined pointwise.

We shall now prove that this preserves valid runs. First, an easy observation:

Lemma C.18. *Consider A a game, and $\boldsymbol{\sigma} : A$ a negative Petri strategy. Then, for all $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha$ s.t. $\mathsf{play}(\rho) \in \mathsf{Plays}(A)$, we have $\mathsf{play}(\rho) \in \mathsf{Plays\_}(A)$.*

Proof. By *negative*, the first transition of ρ cannot be positive or neutral (as those require at least one tokil). Thus, it is negative. □

Lemma C.19. *Consider A a game, $\boldsymbol{\sigma} : A$ negative, $h = [f, (g_{\mathrm{m}})] : A \curvearrowright B$, and $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha$. If* $\mathsf{play}(\rho) \in \mathsf{Plays}\_(A)$*, then* $\mathsf{play}(\rho[h]) = h(\mathsf{play}(\rho)) \in \mathsf{Plays}\_(B)$.

Proof. Straightforward by induction on ρ. □

We shall also use a sort of reciprocal statement:

Lemma C.20. *Consider A a game, $\boldsymbol{\sigma} : A$ negative, $h = [f, (g_{\mathrm{m}})] : A \curvearrowright B$, and $\rho' : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}[h]} \alpha$. If* $\mathsf{play}(\rho') \in \mathsf{Plays}(B)$*, there is a unique $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha$ s.t.* $\mathsf{play}(\rho) \in \mathsf{Plays}(A)$ *and* $\rho' = \rho[h]$.

Proof. Straightforward by induction on ρ'. □

Proposition C.21. *Consider A a game, $\boldsymbol{\sigma} : A$ negative, $h = [f, (g_{\mathrm{m}})] : A \curvearrowright B$. Then, $\boldsymbol{\sigma}[h] : B$ is a negative Petri strategy.*

Proof. *Valid.* Consider $\rho' : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}[h]} \alpha$ such that $\mathsf{play}(\rho') \in \mathsf{Plays}(B)$. Consider $\mathsf{t}^+ : \alpha \xrightarrow{b}_{\boldsymbol{\sigma}[h]} \beta$, write $\mathsf{t} = t^+(\!|\mu|\!) \uplus \gamma$. By Lemma C.20, there is a unique $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha$ such that $\mathsf{play}(\rho) \in \mathsf{Plays}(A)$ and $\rho' = \rho[h]$. By definition of transitions of $\boldsymbol{\sigma}[h]$, we have $b = h(a)$ with

$$t^+(\!|\mu|\!) \uplus \gamma : \alpha \xrightarrow{a} \beta$$

and $\mathsf{play}(\rho)a \in \mathsf{Plays}(A)$ as $\boldsymbol{\sigma}$ is *valid*. Note actually $\mathsf{play}(\rho)a \in \mathsf{Plays}\_(A)$ by Lemma C.18. Hence, $h(\mathsf{play}(\rho)a) = \mathsf{play}(\rho[h])b \in \mathsf{Plays}\_(B)$ by condition *validity* of global renamings, as required.

Receptive. Consider $\rho' : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}[h]} \alpha$ such that $s' = \mathsf{play}(\rho') \in \mathsf{Plays}(B)$. Consider $sb^- \in \mathsf{Plays}(B)$. By Lemma C.20, there is a unique $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha$ such that $s = \mathsf{play}(\rho) \in \mathsf{Plays}(A)$ and $h(s) = s'$. By condition *receptivity* of global renamings, there is $sa^- \in \mathsf{Plays}(A)$ such that $h(a) = b$. As $\boldsymbol{\sigma}$ is *receptive*, there is a unique $\mathsf{t}^- \in \mathsf{IT}_{\boldsymbol{\sigma}}$ such that $\mathsf{t}^- : \emptyset \xmapsto{a^-}_{\boldsymbol{\sigma}} \beta$ for some β. By definition of $\boldsymbol{\sigma}[h]$, $\mathsf{t}^- : \emptyset \xmapsto{b^-}_{\boldsymbol{\sigma}[h]} \beta$ as required. Uniqueness follows immediately from uniqueness for $\boldsymbol{\sigma}$.

Strongly safe. Consider $\rho' : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}[h]} \alpha$ such that $s' = \mathsf{play}(\rho') \in \mathsf{Plays}(B)$. By Lemma C.20, there is a unique $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha$ such that $s = \mathsf{play}(\rho) \in \mathsf{Plays}(A)$ and $h(s) = s'$. If $\mathsf{t}' : \alpha \longrightarrow_{\boldsymbol{\sigma}[h]} \beta$ then also $\mathsf{t}' : \alpha \longrightarrow_{\boldsymbol{\sigma}} \beta$, and $\mathsf{new}(\mathsf{t}')$ is fresh in ρ, so fresh in ρ'. If $\mathsf{t}' : \alpha \xrightarrow{b}_{\boldsymbol{\sigma}[h]} \beta$ with $s'b \in \mathsf{Plays}(B)$, then again by Lemma C.20, $\mathsf{t}' = \mathsf{t}[h]$ and $b = h(a)$ for $\mathsf{t} : \alpha \xrightarrow{a}_{\boldsymbol{\sigma}} \beta$ with $sa \in \mathsf{Plays}(A)$. As $\boldsymbol{\sigma}$ is *strongly safe*, it follows that $\mathsf{new}(\mathsf{t})$ is fresh in ρ, but $\mathsf{new}(\mathsf{t}) = \mathsf{new}(\mathsf{t}')$ so $\mathsf{new}(\mathsf{t}')$ is fresh in ρ' as required.

Negative. Straightforward from the fact that $\boldsymbol{\sigma}$ is *negative*. □

C.2.3 Currying. This is a simple application of global renaming.

Lemma C.22. *Consider $\Gamma, x : A, \Delta$ a list of variable/arena declarations, and O well-opened. Then,*

$$(\Lambda_x, (\mathrm{id})) : (!(\&[\Gamma, x : A, \Delta]) \vdash O) \curvearrowright (!(\&[\Gamma, \Delta]) \vdash\ !A \multimap O)$$

where Λ_x is defined in Definition 3.15.

Proof. Immediate verification. □

Corollary C.23. *Consider $\boldsymbol{\sigma} : !(\&[\Gamma, x : A, \Delta]) \vdash O$ a negative Petri strategy. Then, $\Lambda^{\Gamma,\Delta}_{x:A,O}(\boldsymbol{\sigma}) : !(\&[\Gamma, \Delta]) \vdash\ !A \multimap O$ is a negative Petri strategy.*

C.2.4 Functorial promotion. Rather than directly dealing with Definition 3.16, we decompose it: first, a functorial promotion, and secondly, a renaming corresponding to *digging*.

We first define functorial promotion on Petri structures:

$$\frac{t^0(\!|\alpha|\!) : \alpha \longmapsto_{\sigma} \beta}{t^0(\!|\mathrm{e} :: \alpha|\!) : \mathrm{e} :: \alpha \longmapsto_{!\sigma} \mathrm{e} :: \beta} \qquad \frac{t^-(\!|(\mathrm{s},\mathrm{d})|\!) : \emptyset \overset{m}{\longmapsto}_{\sigma} \beta}{t^-(\!|(\mathrm{e} :: \mathrm{s},\mathrm{d})|\!) : \emptyset \overset{\mathrm{e}::m}{\longmapsto}_{!\sigma} \mathrm{e} :: \beta} \qquad \frac{t^+(\!|\alpha|\!) : \alpha \overset{m}{\longmapsto}_{\sigma} \emptyset}{t^+(\!|\mathrm{e} :: \alpha|\!) : \mathrm{e} :: \alpha \overset{\mathrm{e}::m}{\longmapsto}_{\sigma} \emptyset}$$

Fig. 32. Description of instantiated transitions of $!\sigma$

Definition C.24. Consider $\sigma \in \mathsf{PStruct}(M, N)$. We set $\mathcal{L}_{!\sigma} = \mathcal{L}_{\sigma}$, $\mathcal{T}_{!\sigma} = \mathcal{T}_{\sigma}$ with the same polarities, $\partial_{!\sigma} = \partial_{\sigma}$, and pre- and post-conditions are also unchanged. Finally, the *transition table* is:

$$\begin{array}{rcll} \delta_{!\sigma}\langle t^0\rangle(\mathrm{e} :: \alpha) & = & \mathrm{e} :: \beta & \text{if } \delta_{\sigma}\langle t\rangle(\alpha) = \beta \\ \delta_{!\sigma}\langle t^+\rangle(\mathrm{e} :: \alpha) & = & (\mathrm{e} :: \mathrm{s}, \mathrm{d}) & \text{if } \delta_{\sigma}\langle t\rangle(\alpha) = (\mathrm{s}, \mathrm{d}) \\ \delta_{!\sigma}\langle t^-\rangle(\mathrm{e} :: \mathrm{s}, \mathrm{d}) & = & \mathrm{e} :: \alpha & \text{if } \delta_{\sigma}\langle t\rangle(\mathrm{s}, \mathrm{d}) = \alpha \end{array}$$

where $\mathrm{e} :: \alpha$ is $\{(\mathrm{e} :: \mathrm{s}_i, \mathrm{d}_i)^{@l_i} \mid (\mathrm{s}_i, \mathrm{d}_i)^{@l_i} \in \alpha\}$.
With this definition, we obtain $!\sigma \in \mathsf{PStruct}(!M, !N)$.

We prove that this operation preserves Petri strategies – the proof follows closely that of tensor, of which the ! can be regarded as an infinitary version.

Lemma C.25. *Consider σ a Petri structure.*
Then, instantiated transitions of $!\sigma$ are exactly as in Figure 32 – in the sense that there is a one-to-one correspondence between instantiated transitions in the premises and in the conclusion.

Using this description, we define for each $\mathrm{e} \in \mathcal{E}$ a partial function

$$\begin{array}{rcrcl} \pi^!_{\mathrm{e}} & : & \mathsf{IT}_{!\sigma} & \rightharpoonup & \mathsf{IT}_{\sigma} \\ & & t^0(\!|\mathrm{e} :: \alpha|\!) & \mapsto & t^0(\!|\alpha|\!) \\ & & t^-(\!|(\mathrm{e} :: \mathrm{s}, \mathrm{d})|\!) & \mapsto & t^-(\!|(\mathrm{s}, \mathrm{d})|\!) \\ & & t^+(\!|\mathrm{e} :: \alpha|\!) & \mapsto & t^+(\!|\alpha|\!) \end{array}$$

and undefined otherwise. In order to extend those to instantiated transitions in context, first define

$$\left|\left|\left|_{e \in \mathcal{E}} \alpha_e \right.\right.\right. = \biguplus_{e \in \mathcal{E}} e :: \alpha_e$$

for $(\alpha_{\mathrm{e}})_{\mathrm{e} \in \mathcal{E}}$ a family of conditions empty almost everywhere. We may then set:

$$\begin{array}{rcrcl} \pi^!_{\mathrm{e}} & : & \mathsf{ITC}_{!\sigma} & \rightharpoonup & \mathsf{ITC}_{\sigma} \\ & & \mathbf{t} \uplus (|||_{\mathrm{e} \in \mathcal{E}}\, \gamma_{\mathrm{e}}) & \mapsto & \pi^!_{\mathrm{e}}(\mathbf{t}) \uplus \gamma_{\mathrm{e}}\,. \end{array}$$

Using these, from a run $\rho : \emptyset \longrightarrow\!\!\!\!\rightarrow_{!\sigma} \alpha$ we extract, for all $\mathrm{e} \in \mathcal{E}$:

$$\rho_{\mathrm{e}} = \rho \upharpoonright \pi^!_{\mathrm{e}}\,,$$

we will also use for restrictions the partial functions

$$\begin{array}{rcrcl} \mathrm{e} & : & \mathsf{Moves} & \rightharpoonup & \mathsf{Moves} \\ & & (\mathrm{m}, \mathrm{e} :: \mathrm{s}, \mathrm{d}) & \mapsto & (\mathrm{m}, \mathrm{s}, \mathrm{d}) \end{array}$$

and undefined otherwise – the abuse of notations should not create confusion.
Now, as for the tensor we can prove:

Lemma C.26. *Consider $\rho : \emptyset \longrightarrow\!\!\!\!\rightarrow_{!\sigma} \alpha$. Then, $\alpha = |||_{\mathrm{e} \in \mathcal{E}}\, \alpha_{\mathrm{e}}$ and*

$$\rho_{\mathrm{e}} : \emptyset \longrightarrow\!\!\!\!\rightarrow_{\sigma} \alpha_{\mathrm{e}}$$

for all $\mathrm{e} \in \mathcal{E}$, where $\mathsf{play}(\rho_{\mathrm{e}}) = \mathsf{play}(\rho) \upharpoonright \mathrm{e}$.
Moreover, $\mathsf{Coll}(\rho) = |||_{\mathrm{e} \in \mathcal{E}}\, \mathsf{Coll}(\rho_{\mathrm{e}})$.

Proof. The proof is the same as for Lemma C.4, without synchronized transitions. ◻

The construction goes on as for the tensor, with preservation of plays via projections:

Lemma C.27. *Consider A an arena and $s \in |!A|^*$.*
Then, $s \in \mathsf{Plays}(!A)$ iff $s \upharpoonright \mathrm{e} \in \mathsf{Plays}(A)$ for all $\mathrm{e} \in \mathcal{E}$.

Proof. Straightforward. ◻

Consider now $\boldsymbol{\sigma} : A \vdash B$ a Petri strategy.
Using Lemmas C.27 and C.26, we show that projections preserve valid runs.

Lemma C.28. *Consider $\rho : \emptyset \longrightarrow\!\!\!\!\rightarrow_{!\sigma} \alpha$ such that $\mathsf{play}(\rho) \in \mathsf{Plays}(!A \vdash !B)$.*
Then, for all $\mathrm{e} \in \mathcal{E}$, $\mathsf{play}(\rho_{\mathrm{e}}) \in \mathsf{Plays}(A \vdash B)$.

Proof. As for the proof of Lemma C.6 (without synchronization), using condition *valid* of Petri strategies along with Lemma C.27. ◻

Using Lemmas C.26, C.27 and C.28, we prove as for Proposition C.7:

Proposition C.29. *If $\boldsymbol{\sigma} : A \vdash B$ is a Petri strategy, then so is $!\boldsymbol{\sigma} : !A \vdash !B$.*
Moreover, if $\boldsymbol{\sigma}$ is negative then so is $!\boldsymbol{\sigma}$.

C.2.5 Local renamings. To match Definition 3.16, we must also rename following *digging*.
Recall that digging is the following map:

$$\begin{array}{rrcl} \mathsf{dig} & : \qquad \mathsf{Moves} & \rightharpoonup & \mathsf{Moves} \\ & (\mathsf{m}, \mathsf{e} :: \mathsf{e}' :: \mathsf{s}, \mathsf{d}) & \mapsto & (\mathsf{m}, \langle \mathsf{e}, \mathsf{e}' \rangle :: l, \mathsf{d}) \end{array}$$

and undefined otherwise. To rename a strategy following this, it is convenient to introduce:

Definition C.30. Consider A, B arenas.
A **(local) renaming** from A to B is a partial injection $f : \mathsf{Moves} \rightharpoonup \mathsf{Moves}$ defined on $|A|$, s.t.:

$$\begin{array}{rl} \textit{validity:} & \text{for all } x \in \mathscr{C}(A),\ fx \in \mathscr{C}(B), \\ \textit{polarity-preserving:} & \text{for all } a \in |A|,\ \mathsf{pol}(f(a)) = \mathsf{pol}(a), \\ \textit{receptivity:} & \text{for all } x \in \mathscr{C}(A), \text{ if } f(x) \vdash_B b^-, \\ & \text{then there is } x \vdash_A a \text{ such that } f(a) = b, \\ \textit{courtesy:} & \text{for all } a \rightarrowtriangle_A a', \text{ either } f(a) \rightarrowtriangle_B f(a') \text{ or } (\mathsf{pol}(a), \mathsf{pol}(a')) = (-, +). \end{array}$$

We write $f : A \curvearrowright B$ to mean that f is a renaming from A to B.

It is clear in particular that $\mathsf{dig} : !!A^\perp \to !A$ is a renaming, for any arena A. This is a variant of Definition C.16, closer to the usual lifting operation used for this purpose in concurrent games.
Clearly, a local renaming is a global renaming. But local renamings are sometimes more convenient, because if $f : A^\perp \curvearrowright B^\perp$ and $g : A' \curvearrowright B'$ are local renaming, then it is obvious that $f \vdash g : A \vdash B \curvearrowright A' \vdash B'$ (defined in the obvious way) is still a local renaming – this is not always the case for global renamings for non-negative games.

Definition C.31. Consider $\boldsymbol{\sigma} : A \vdash B$ a negative Petri strategy and $f : A^\perp \curvearrowright A'^\perp, g : B \curvearrowright B'$.
Then, we define $g \cdot \boldsymbol{\sigma} \cdot f = \boldsymbol{\sigma}[f \vdash g] : A' \vdash B'$.

This yields a negative Petri strategy by Proposition C.21.

C.2.6 Digging. We may finally perform digging and deduce the correctness of promotion:

Proposition C.32. *Consider $\boldsymbol{\sigma} : !A \vdash B$ a negative Petri strategy. Then, $\boldsymbol{\sigma}^\dagger : !A \vdash !B$ is a negative Petri strategy.*

Proof. It is a direct verification that $\boldsymbol{\sigma}^\dagger = (!\boldsymbol{\sigma})[\mathsf{dig} \vdash \mathsf{id}]$, which is a negative Petri strategy by Propositions C.29 and Definition C.31. ◻

C.3 PStrat as an IPA-Structure: Primitives

We now show that the Petri structures representing the primitives of IPA are indeed Petri strategies.

Given a Petri structure $\boldsymbol{\sigma}$ and a play s of a game A, we say that s is reachable by $\boldsymbol{\sigma}$ when there exists a run ρ of $\boldsymbol{\sigma}$ with $\mathsf{play}(\rho) = s$. Given a game A, we define the Scott order on $\mathscr{C}(A)$ as follows: $x \sqsubseteq_A y := (x \supseteq^- \subseteq^+ y)$ which was already encountered in Lemma C.8 for copycat.

C.3.1 Variable and Evaluation. For variable, we notice that $\mathbf{var}_{x:M} = \text{æ}_M[\iota^x_\& \vdash \mathsf{id}]$ and we conclude easily by C.21 since $\iota^x_\& : M^\perp \curvearrowright [\Gamma, x : M, \Delta]^\perp$ is a local renaming. For the evaluation, the map Ω defined in Section 3.4 is a global renaming $((M \multimap N) \vdash (M \multimap N)) \curvearrowright ((M \multimap N) \otimes M \vdash N)$.

C.3.2 Contraction. We now show that the Petri structure $\mathbf{c}_A$ is a Petri strategy on $!A \vdash !A \otimes !A$.

Given a move a, we write $\ell_!(a)$ for $(\mathsf{m}, \ell_!(\mathsf{e}) :: \mathsf{s}, \mathsf{d})$ for $a = (\mathsf{m}, \mathsf{e} :: \mathsf{s}, \mathsf{d})$, and similarly for $r_!(a)$. It is not defined on moves with an empty stack.

Lemma C.33. *Consider $s \in \mathsf{play}(!A \vdash !A \otimes !A)$ reachable by $\mathbf{c}_A$. Then, $|s| = (\ell_!(x_1) \uplus r_!(x_2) \vdash y_1 \otimes y_2)$ and $y_i \sqsubseteq x_i$.*

Proof. We prove the implication by induction on the length of s. It holds for all plays of length zero. We assume the implication holds for all plays of length n.

Consider $s' = s \cdot a$ reachable by $\boldsymbol{\sigma}$ and s has length n. We apply the induction hypothesis to s (which is reachable by $\boldsymbol{\sigma}$) and obtain that, writing $|s| = (\ell_!(x_1) \uplus \ell_!(x_2) \vdash y_1 \otimes y_2)$, we have $y_i \sqsubseteq x_i$.

- If a is negative, and on $\ell_\vdash$, then the inequality for s' holds by definition of $\sqsubseteq$.
- If a is negative, and on $r_\vdash$, then because s' is a play, the parent of a must exist and belong to s. By induction, that parent must have an exponential stack of the form $\ell_!\mathsf{e} :: \mathsf{s}$ or $r_!\mathsf{e} :: \mathsf{s}$ and we can conclude.
- If a is positive, and on the left, ie. $m = \ell_\vdash m_0$: in the run producing a, there must be a token in m_0^- that triggered t. That token must be (s, d) since t has a trivial transition function. That token can only be produced by one of the negative transitions $r_\vdash \ell_\otimes m_0$ or $r_\vdash r_\otimes m_0$ – assume the former. This directly shows that $\mathsf{s} = \ell_!\mathsf{e} :: \mathsf{s}'$ for some s', and that $(r_\vdash \ell_\otimes m_0, \mathsf{e} :: \mathsf{s}, \mathsf{d}) \in |s|$, which implies that $(m_0, \mathsf{e} :: \mathsf{s}, \mathsf{d}) \in y$. As a result $|s| = (x \cup \{(m_0, \ell_!\mathsf{e} :: \mathsf{s}', \mathsf{d})\}) \vdash y \otimes z$ satisfies the desired property.
- If a is positive and on the right for instance $m = r_\vdash \ell_\otimes m_0$. Then a similar line of reasoning shows that $\mathsf{s} = \mathsf{e} :: \mathsf{s}'$ and we must have $(\ell_\vdash m_0, \ell_!\mathsf{e} :: \mathsf{s}', \mathsf{d}) \in |s|$ which entails the desired property. ◻

Lemma C.34. *$\mathbf{c}_A$ is a negative Petri strategy on $!A \vdash !A \otimes !A$.*

Proof. *Negative.* Simple inspection of the net.

Strong safety. Consider $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$ with s a play, and $\mathsf{t} : \alpha \xrightarrow{a} \beta$ with sa also a play (note that there is no neutral transition). Note that positive transitions do not create tokens, so there is nothing to check. For negative transitions, it follows from the injectivity of transition functions and the fact that plays are non-repetitive.

Validity. Consider $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$ a run of $\mathbf{c}_A$ and s a play of the game. Assume that ρ can extend by $\mathsf{t}^+ : \alpha \xrightarrow{a} \beta$. By Lemma C.33, we know that $|s| = (\ell_!(x_1) \uplus r_!(x_2) \vdash y_1 \otimes y_2)$ with $y_i \sqsubseteq x_i$. There are three transitions, hence three cases. We detail the case for $a = \ell_{\vdash} a_0$: then by inspecting the net we have that a_0 must be of the form $\ell_!(a_1)$ with $a_1 \in y$ or $r_!(a_1)$ with $a_1 \in z$ – assume the former. From $a_1 \in z$, we deduce that the justifier of a is already present in s; and moreover a cannot conflict with anything in s. That sa is non-repetitive follows from strong safety and that transition functions are injective.

Receptivity. Consider sa^- a play of $!A \vdash !A \otimes !A$ and $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$. If a is on the right of $\vdash$, then we can use the corresponding transition whose function domain is total on stacks of $!A$. For a on the left, a cannot be minimal so its justifier a_0 must occur in s. By Lemma C.33, its exponential stack must start with $\ell_!(\mathrm{e})$ or $r_!(\mathrm{e})$, and thus so must that of a. As a result, a will be accepted by the transition corresponding to its address. ◻

C.3.3 Fixpoint.

We start by characterising the plays of Y_O, where O is a well-opened arena. We reuse the same encodings as in Section B.3.4.

Lemma C.35. *Let $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$ be a run of Y_O such that s is a play.*
Then there exists suffix-closed $J \subseteq \mathcal{E}^+$, configurations $z, y_\epsilon \in \mathscr{C}(O)$, $(y_{\mathrm{s}})_{\mathrm{s}\in J}$ and $(z_{\mathrm{s}} \in \mathscr{C}^{\neq\emptyset}(O))_{\mathrm{s}\in J}$ with J empty if y_ϵ is, $z \sqsubseteq y_\epsilon$ and $z_{\mathrm{s}} \sqsubseteq y_{\mathrm{s}}$ for all s and

$$|s| = (\emptyset \multimap () :: y_\epsilon) \uplus (\mathrm{e} :: (\mathrm{s}) :: z_{e\cdot\mathrm{s}} \multimap (\mathrm{e}\cdot\mathrm{s}) :: y_{\mathrm{s}}) \vdash z,$$

plus if $y_\epsilon = \emptyset$ then $J = \emptyset$, and if $\mathrm{e}\cdot\mathrm{s} \in J$, then $y_{\mathrm{s}} \neq \emptyset$.

Proof. A direct induction over the run, using the transition table. ◻

Lemma C.36. *Y_O is a negative Petri strategy $!(!O \multimap O) \vdash O$.*

Proof. *Negativity* and *receptivity* are easily verified.

Validity. Consider $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$ be a run of Y_O such that s is a play of the game. Consider now an extension of ρ by the positive transition $\mathrm{t} : \alpha \xrightarrow{a} \beta$. We show that sa is a valid play. First, if a or a conflicting move occurs already in s, given the shape of the net, this means that Opponent played twice the same move or two conflicting moves earlier in s which is absurd. It remains to show that the predecessor of a occurs in s, which is a consequence of Lemma C.35

Strong-safety. All negative transitions have injective transition functions, and the two negative transitions $r_{\vdash}\mathrm{m}^-$ and $\ell_{\vdash}\ell_{\multimap}\mathrm{m}^-$ which have a common postcondition (m^-), have disjoint codomains, hence Y_O is strongly safe. ◻

C.3.4 Queries, Conditional, Constants.

For these IPA structures defined on linear games, a simple inspection shows that they define they are IPA strategies.

C.3.5 Let bindings.

We now move on to showing that **let** is a Petri strategy on $(!\mathbf{X} \multimap \mathbf{Y}) \otimes \mathbf{X} \vdash \mathbf{Y}$.

Lemma C.37. *Let $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$ be a valid run for* **let**. *Then $|s| = ((\uplus_{\mathrm{e}\in I}\mathrm{e} :: x_{\mathrm{e}}) \multimap y) \otimes z \vdash w$ with: (1) if $y \neq \emptyset$ then z is maximal in $\mathscr{C}(\mathbf{X})$; (2) $w \sqsubseteq y$ and $x_{\mathrm{e}} \subseteq z$ for all $\mathrm{e} \in I$.*

Proof. By induction on ρ. ◻

Lemma C.38. **let** *is a Petri strategy on* $(!\mathbf{X} \multimap \mathbf{Y}) \otimes \mathbf{X} \vdash \mathbf{Y}$.

Proof. As usual, receptivity and negativity are clear. Strong safety is clear on the forwarding transitions. For the transition s, we note that the token in location 3 is never in $\mathrm{eat}(\mathrm{s})$, and the other token at location 5 has a stack given by Opponent, so there cannot be any risk of confusion.

Validity follows from Lemma C.37. ◻

C.3.6 Newref and newsem. We now show that **newref** and **newsem** are valid Petri strategies. We focus our attention on **newref**, the proof for **newsem** being similar.

We start by recovering, out of a run of **newref**, a memory trace. A memory trace is a word on the alphabet $\Sigma := \mathcal{E} \times \{r, w\} \times \mathcal{D}$. There is a a partial function $\pi : \mathsf{IT}_{\mathbf{newref}} \rightharpoonup \Sigma$ as follows:

$$\pi(\mathsf{w}(\!|\{([\mathrm{e}], \mathrm{d})^{@3}, \_\}|\!)) = (\mathrm{e}, w, \mathrm{d}) \qquad \pi(\mathsf{r}(\!|\{([\mathrm{e}], \_)^{@5}, (\_, \mathrm{d})^{@2}\}|\!)) = (\mathrm{e}, r, \mathrm{d}),$$

and undefined everywhere else. We write $\mathsf{Tr}(\rho) = \rho \upharpoonright \pi$.

A memory trace is **consistent** when (1) exponential signatures occurring in it are all distinct, and (2) each read reads the last value written before, or zero if there are no writes.

Lemma C.39. *Consider a run $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$ of* **newref** *such that s is a play. Then:*

- $\mathsf{Tr}(\rho)$ *is a consistent memory trace.*
- *If ρ is not empty, then there is a unique tokil $(\mathrm{s}, \mathrm{d})^{@2}$ in α such that: if $\mathsf{Tr}(\rho)$ is empty then $\mathrm{s} = []$ and $\mathrm{d} = 0$, otherwise $\mathrm{s} = [\mathrm{e}]$ with e and d the components of the last operation in $\mathsf{Tr}(\rho)$.*
- *$|s|$ has the shape $((\uplus_{\mathrm{e} \in I} \mathrm{e} :: x_{\mathrm{e}}) \multimap y) \vdash z$ with $z \sqsubseteq y$ and $x_{\mathrm{e}} \in \mathscr{C}(\mathbf{V})$ such that:*
  - *if x_{e} is non-empty then e occurs in $\mathsf{Tr}(\rho)$ and the value coincide in the case of a read.*
  - *For every signature e occurring in $\mathsf{Tr}(\rho)$, x_{e} is non-empty.*

Proof. We proceed by induction on ρ, the base case being trivial. We assume $\rho = \rho' \cdot \mathsf{t}$ with $\mathsf{t} : \alpha \to \beta$. For the visible transitions $r_{\vdash}(-)$ and $\ell_{\vdash} r_{\multimap}(-)$, this is a proof similar to copycat.

- If t is on $\ell_{\vdash}\ell_{\multimap}w_{\mathbb{V}}\mathsf{Q}^-$ or $\ell_{\vdash}\ell_{\multimap}r_{\mathbb{V}}\mathsf{Q}^-$, there is nothing to add to the induction hypothesis.
- If $\mathsf{t} = \gamma \uplus \mathsf{w}\{([\mathrm{e}], \mathrm{d})^{@2}, ([\mathrm{e}'], \mathrm{d}')^{@3}\}$: then $\mathsf{Tr}(\rho) = \mathsf{Tr}(\rho')([\mathrm{e}'], w, \mathrm{d}')$ is still a consistent trace. Moreover, from the tokil $([\mathrm{e}'], \mathrm{d}')^{@3}$, we deduce that in ρ' there must be a visible transition with move $(\ell_{\vdash}\ell_{\multimap}w_{\mathbb{V}}\mathsf{Q}^-, [\mathrm{e}], \mathrm{d}')$.
- If $\mathsf{t} = \gamma \uplus \mathsf{w}\{([\mathrm{e}], \mathrm{d})^{@2}, ([\mathrm{e}'], \bullet)^{@5}\}$: the same line of reasoning works, except that $\mathsf{Tr}(\rho) = \mathsf{Tr}(\rho')([\mathrm{e}', r, \mathrm{d}])$ is no longer automatically consistent. However, by induction we know that in α there is a unique token at location 2, and that its value is the last value written or zero if there is not any – which shows that $\mathsf{Tr}(\rho)$ is indeed consistent.
- If $\mathsf{t} = \gamma \uplus \ell_{\vdash}\ell_{\multimap}w_{\mathbb{V}}\mathsf{A}([\mathrm{e}, \mathrm{d}]^{@4})$: the first two conditions are trivially true. Moreover, since $([\mathrm{e}], \mathrm{d})^{@4}$ belongs to α, there must have been a transition w in α before that put it there. That shows that there must be an element in $\mathsf{Tr}(\rho)$ with exponential token e as desired. ◻

Lemma C.40. **newref** *is a Petri strategy on* $!\mathbf{V} \multimap \mathbf{X} \vdash \mathbf{X}$.

Proof. Negativity and receptivity follow by inspection of the net and transition tables.

Strong-safety. Consider a run $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$ such that s is a play, that can be extended by a transition $\mathsf{t} : \alpha \longrightarrow \beta$ that is negative or neutral, with $\mathsf{play}(\rho\mathsf{t})$ being a play.

Initial question. If t is a negative transition $\alpha \xrightarrow{a} \beta$ on the address $r_{\vdash}\mathsf{Q}^-$. Then necessarily $\rho = \epsilon$ and so $\mathsf{Coll}(\rho) = \emptyset$.

Final return. If t is a negative transition $\alpha \xrightarrow{a} \beta$ on the address $\ell_{\vdash}r_{\multimap}\mathsf{A}^-$: trivial since the function of this transition is simply the identity, it follows from sa being a play hence non-repetitive.

Request. If t is a negative transition $\alpha \xrightarrow{a} \beta$ on the address $\ell_{\vdash}\ell_{\multimap}w_{\mathbb{V}}\mathsf{Q}^-$ or $\ell_{\vdash}\ell_{\multimap}r_{\mathbb{V}}\mathsf{Q}^-$. In both cases, the transition function is again the identity, so we can conclude by the same argument.

Atomic operation. If t arises from w or r. The two cases being symmetric, we only show for w. From the Petri structure, we get that $\alpha = \gamma \uplus \{([\mathrm{e}], d)^{@3}, (\mathrm{s}, \mathrm{d}')^{@2}\}$ and $\beta = \gamma \uplus \{([\mathrm{e}], \mathrm{d})^{@2}, ([\mathrm{e}], \checkmark)^{@6}\}$.

We show that $\mathsf{new}(\mathsf{t})$ is fresh in ρ. For the token in location 6, which is always in $\mathsf{new}(\mathsf{t})$, only the transition w writes to 6, so if the tokil $([\mathrm{e}], \checkmark)^{@6}$ appeared before in ρ, it means that

there would be already a tokil $([\mathrm{e}], \mathrm{d}'')^{@3}$ in $\mathsf{Coll}(\rho)$. This is not possible because 3 is only fed via the negative transition on $\ell_\vdash \ell_\multimap w_\forall \mathsf{Q}^-$. This means that Opponent would have played $(\ell_\vdash \ell_\multimap w_\forall \mathsf{Q}^-, [\mathrm{e}], \mathrm{d}'')$ which violates the fact that s is non-repetitive (if $\mathrm{d} = \mathrm{d}''$) or that s is a play (if $\mathrm{d} \neq \mathrm{d}''$ as those moves are in conflict).
Finally, if $([\mathrm{e}], \mathrm{d})^{@2}$ is in $\mathsf{Coll}(\rho)$, then it means that a previous instance of $\mathtt{w}$ or $\mathtt{r}$ produced it, which means that there must have been a tokil $([\mathrm{e}], \mathrm{d})^{@3}$ (for $\mathtt{w}$) or $([\mathrm{e}], \bullet)^{@5}$ (for $\mathtt{r}$). That implies there has been two Opponent moves on addresses of the form $\ell_\vdash \ell_\multimap(-)$ with the same exponential address, which is not allowed by the game as they are all in conflict.

Valid. Consider a run $\rho : \emptyset \xrightarrow{s}\!\!\!\!\!\rightarrow \alpha$ with s a play, and a positive extension $\mathsf{t} : \alpha \xrightarrow{a} \beta$. There are several cases depending on the address of a:

- If a is on $r_\vdash \mathsf{A}$: easy since ρ is non empty it must contain its justifying move. Moreover a or a conflicting move with a cannot occur in s, since we simply forward moves received from address $\ell_\vdash r_\multimap \mathsf{A}$.
- If a is on $\ell_\vdash r_\multimap \mathsf{Q}$: same reasoning.
- If a is for instance on $\ell_\vdash \ell_\multimap w_\forall \mathsf{A}$ (the case for $r_\forall$ is similar). This means that in location 4, there must be a tokil $([\mathrm{e}], \checkmark)^{@4}$. That tokil proves that, there must be an entry (e, w, d) (for some d) in $\mathsf{Tr}(\rho)$. By Lemma C.39, we know that in s there must be justifying move $(\ell_\vdash \ell_\multimap w_\forall \mathsf{Q}, [\mathrm{e}], d)$. Moreover, if a or a conflicting move would be already present in a, then we could apply the same reasoning and find a contradiction with the fact that $\mathsf{Tr}(\rho)$ cannot repeat twice the same exponential token. □

D THE UNFOLDING

We provide some detailed proofs of the unfolding to strategies.

D.1 Construction of the Unfolding

Fix a game A, and a Petri strategy $\boldsymbol{\sigma} : A$. First, for a valid run $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha$, we write $\mathrm{post}(\rho) = \alpha$. If $\mathrm{x} \in \mathsf{Hist}(\boldsymbol{\sigma})$, we write $\mathrm{post}(\mathrm{x}) = \mathrm{post}(\rho)$ for any ρ such that $\mathrm{x} = \mathsf{IT}_\rho$. This is justified by:

Lemma D.1. *Consider $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha$ and $\rho' : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha'$ valid runs such that $\mathsf{IT}_\rho = \mathsf{IT}_{\rho'}$. Then, $\alpha = \alpha'$.*

Proof. Exploiting *strong safety*, it is immediate by induction on ρ that:

$$\alpha = \left(\uplus\{\mathrm{post}(\mathbf{t}) \mid \mathbf{t} \in \mathsf{IT}_\rho\}\right) \setminus \left(\uplus\{\mathrm{pre}(\mathbf{t}) \mid \mathbf{t} \in \mathsf{IT}_\rho\}\right)$$

from which the result immediately follows. □

We aim to prove that valid runs exactly correspond to linearizations of histories. The first step is:

Lemma D.2. *Consider ρ a valid run of $\boldsymbol{\sigma}$ of the form $\rho = \rho_0 \cdot (\mathbf{t} \uplus \alpha) \cdot (\mathbf{t}' \uplus \alpha')$. If $\mathbf{t}$ is maximal in IT_ρ, then $\rho_0 \cdot (\mathbf{t}' \uplus \beta') \cdot (\mathbf{t} \uplus \beta)$ is a valid run for some β, β'.*

Proof. First, we show that for $\beta' = \mathrm{post}(\rho_0) \setminus \mathrm{pre}(\mathbf{t}')$, ρ_0 extends by $\mathbf{t}' \uplus \beta'$ which means showing:

$$\begin{array}{lrcl} (1) & \beta' \cap \mathrm{pre}(\mathbf{t}') & = & \emptyset \\ (2) & \mathrm{pre}(\mathbf{t}') & \subseteq & \mathrm{post}(\rho_0) \\ (3) & \beta' \cap \mathrm{post}(\mathbf{t}') & = & \emptyset \end{array}$$

First, (1) is by construction of β'. For (2), consider $e \in \mathrm{pre}(\mathbf{t}')$. Then e must either be in $\alpha \subseteq \mathrm{post}(\rho_0)$, or in $\mathrm{post}(\mathbf{t})$. If it is in $\mathrm{post}(\mathbf{t})$, then it cannot be a token *produced* by $\mathbf{t}$ (i.e. in $\mathrm{new}(\mathbf{t})$) as $\mathbf{t}$ and $\mathbf{t}'$ are incomparable. So it must be in $\mathrm{pre}(\mathbf{t}) \subseteq \mathrm{post}(\rho_0)$ as desired. For (3), consider $e \in \beta' \cap \mathrm{post}(\mathbf{t}')$. The tokil e must be in $\mathrm{new}(\mathbf{t}')$, which implies since $\mathbf{t}$ and $\mathbf{t}'$ are incomparable that e does not appear in $\mathrm{pre}(\mathbf{t})$. Since $e \in \mathrm{post}(\rho_0)$, it must be that $e \in \alpha$. Since α' must be disjoint from $\mathrm{post}(\mathbf{t}')$, we have $e \in \mathrm{pre}(\mathbf{t}')$ which is absurd. Hence, $\rho_1 = \rho_0 \cdot (\mathbf{t}' \uplus \beta')$ is indeed a valid run.

We now let $\beta = \mathrm{post}(\rho_1) \setminus \mathrm{pre}(\mathbf{t})$ and must prove (1) $\mathrm{pre}(\mathbf{t}) \subseteq \mathrm{post}(\rho_1)$; and (2) $\beta \cap \mathrm{post}(\mathbf{t}) = \emptyset$. For (1), if $e \in \mathrm{pre}(\mathbf{t})$, then e is in $\mathrm{post}(\rho_0)$. As a result, either e is not in $\mathrm{pre}(\mathbf{t}')$, which implies that $e \in \beta'$ hence $e \in \mathrm{post}(\rho_1)$ (as desired), or $e \in \mathrm{pre}(\mathbf{t}')$ as well. In the second case, we have then $e \in \mathrm{pre}(\mathbf{t}) \cap \mathrm{pre}(\mathbf{t}')$. Because, in ρ, $\mathbf{t}$ comes before $\mathbf{t}'$, this implies that e cannot be eaten by $\mathbf{t}$, in other words $e \in \mathrm{post}(\mathbf{t})$. This implies $e \notin \mathrm{eat}(\mathbf{t}')$ as the two transitions are incomparable, i.e. $e \in \mathrm{post}(\mathbf{t}') \subseteq \mathrm{post}(\rho_1)$. For (2), consider $e \in \beta \cap \mathrm{post}(\mathbf{t})$, ie. in particular $e \in \mathrm{new}(\mathbf{t})$. As $e \in \mathrm{post}(\rho_1)$, e either belongs to β' or $\mathrm{post}(\mathbf{t}')$. In the first case, this means that $e \in \mathrm{post}(\rho_0) \cap \mathrm{post}(\mathbf{t})$, which can only be if $e \in \mathrm{pre}(\mathbf{t})$ which is absurd. In the second case, it means that $e \in \mathrm{post}(\mathbf{t}')$, which in turn means that $e \in \mathrm{pre}(\mathbf{t}')$ as e is produced by $\mathbf{t}$ so it cannot be produced by $\mathbf{t}'$ as well by strong safety. But that is not possible either as it would imply a dependency from $\mathbf{t}$ to $\mathbf{t}'$. □

Lemma D.3. *Consider* $\mathsf{x} \in \mathsf{Hist}(\boldsymbol{\sigma})$ *and* $\mathbf{t}$ *a maximal element of* x.
Then, there exists a valid run ρ *ending in* $\mathbf{t} \uplus \alpha$ *(for some context* α*) such that* $\mathsf{IT}_\rho = \mathsf{x}$.

Proof. Consider a run ρ_0 spanning x, which must have the shape:

$$\rho_0 = \rho_1 \cdot (\mathbf{t} \uplus \alpha) \cdot (\mathbf{t}_1 \uplus \alpha_1) \cdot \ldots \cdot (\mathbf{t}_n \uplus \alpha_n)\,.$$

We proceed by induction on n. If $n = 0$, then $\mathbf{t}$ already occurs at the end of ρ. For $n+1$, we consider ρ' the prefix of ρ where the last transition has been removed. By IH, we get a run χ with $\mathsf{IT}_\chi = \mathsf{x} \setminus \{\mathbf{t}_{n+1}\}$ and χ ends with $\mathbf{t}$. By Lemma D.1, we have $\mathrm{post}(\chi) = \mathrm{post}(\rho')$; from that it is immediate that $\chi \cdot (\mathbf{t}_{n+1} \uplus \alpha_{n+1})$ is a valid run, and we conclude by Lemma D.2. □

Lemma D.4. *Consider* $\mathsf{x} \in \mathsf{Hist}(\boldsymbol{\sigma})$.
Then, valid runs ρ *such that* $\mathsf{x} = \mathsf{IT}_\rho$ *exactly correspond to linearizations of* x.

Proof. Clearly, all runs preserve $\leq_{\mathsf{x}}$. For the converse, for any transition $\mathbf{t}$ maximal in $\mathscr{T}(\mathsf{x})$, we obtain a run where it is played last by taking any valid run ρ such that $\mathsf{x} = \mathsf{IT}_\rho$, and pushing $\mathbf{t}$ to the end via local permutations – maximality of $\mathbf{t}$ ensures that there is no obstruction – see Lemma D.3. Iterating this process, we can indeed obtain any linearization. □

Proposition 5.10. *The set comprising all* $\mathscr{T}(\mathsf{x})$ *for* $\mathsf{x} \in \mathsf{Hist}(\boldsymbol{\sigma})$*, is a rigid family written* $\mathscr{T}(\boldsymbol{\sigma})$*. Moreover,* $\mathscr{T}(\boldsymbol{\sigma})$ *(ordered by rigid inclusion) is order-isomorphic to* $\mathsf{Hist}(\boldsymbol{\sigma})$ *(ordered by inclusion).*

Proof. First, the claimed order-isomorphism is clear by construction.

Rigid-closed. Now if $\mathscr{p} \in \mathscr{T}(\boldsymbol{\sigma})$ and $\mathscr{q} \hookrightarrow \mathscr{p}$, by Lemma D.4 there is a valid run ρ playing $\mathscr{q}$ first. Truncating ρ after $\mathscr{q}$, we get ρ' such that $\mathscr{T}(\rho') = \mathscr{q}$ by construction.

Binary-compatible. Take $X \subseteq_f \mathsf{Hist}(\boldsymbol{\sigma})$. Clearly, if *(1)* there are $\mathsf{x}, \mathsf{y} \in X$, visible instantiated transitions $\mathbf{t}$ in x and $\mathbf{t}'$ in y labelled by conflicting events of $A \vdash B$; or *(2)* there are $\mathsf{x}, \mathsf{y} \in X$, $\mathbf{t} : \alpha \longrightarrow_\sigma \beta$ in x and $\mathbf{t}' : \alpha' \longrightarrow_\sigma \beta'$ in y such that $\alpha \cap \alpha' \neq \emptyset$; then there cannot be a valid run witnessing $\cup X$: *(1)* would contradict validity of the run, while *(2)* would contradict *strong safety* as the same tokil would have to be consumed twice. Reciprocally, if we have neither *(1)* nor *(2)*, then any valid runs $(\rho_{\mathsf{x}})_{\mathsf{x} \in X}$ may be directly "zipped" into a valid run witnessing $\cup X \in \mathsf{Hist}(\boldsymbol{\sigma})$.

This concludes the proof, as it brings compatibility of $X \subseteq_f \mathscr{T}(\boldsymbol{\sigma})$ to pairwise compatibility. □

Proposition 5.12. *The event structure* $\mathscr{U}(\boldsymbol{\sigma}) = \mathrm{Pr}(\mathscr{T}(\boldsymbol{\sigma})) \downarrow \mathscr{V}_{\boldsymbol{\sigma}}$*, equipped with the display map*

$$\begin{array}{rcccl} \partial_{\mathscr{U}(\boldsymbol{\sigma})} & : & |\mathscr{U}(\boldsymbol{\sigma})| & \to & |A \vdash B| \\ & & \mathscr{q} & \mapsto & \partial_{\boldsymbol{\sigma}}(\mathsf{top}(\mathscr{q})) \end{array}$$

is a strategy in the sense of Definition 4.6. Moreover, $\mathscr{U}(\boldsymbol{\sigma})$ *is negative if* $\boldsymbol{\sigma}$ *is.*

Proof. It remains to prove *courteous*, *receptive* and *negative*. Recall the order-isomorphism

$$K_{\boldsymbol{\sigma}} : \mathscr{C}(\mathscr{U}(\boldsymbol{\sigma})) \cong \mathscr{T}^V(\boldsymbol{\sigma})$$

obtained in (4). We now prove the remaining conditions.

Courteous. Consider $\mathsf{x}_1 \rightarrowtriangle_{\mathscr{U}(\boldsymbol{\sigma})} \mathsf{x}_2$ such that $\mathrm{pol}(\mathsf{x}_1) = +$ or $\mathrm{pol}(\mathsf{x}_2) = -$. So $\mathsf{x}_1, \mathsf{x}_2 \in \mathscr{T}(\boldsymbol{\sigma})$ with respective top elements $\mathrm{top}(\mathsf{x}_1) = \mathbf{t}_1$ and $\mathrm{top}(\mathsf{x}_2) = \mathbf{t}_2$, such that $\mathrm{pol}(\mathbf{t}_1) = +$ or $\mathrm{pol}(\mathbf{t}_2) = -$. By definition, $\mathsf{x}_1 \hookrightarrow \mathsf{x}_2$, so that $\mathbf{t}_1, \mathbf{t}_2 \in \mathsf{x}_2$ with $\mathbf{t}_1 <_{\mathsf{x}_2} \mathbf{t}_2$. By definition, this means there is a sequence

$$\mathbf{t}_1 \lhd_{\mathsf{x}_2} \mathbf{t}'_1 \lhd_{\mathsf{x}_2} \dots \mathbf{t}'_n \lhd_{\mathsf{x}_2} \mathbf{t}_2$$

where, following Definition 5.8, each $\lhd_{\mathsf{x}_2}$ is either $\lhd_{\boldsymbol{\sigma}}$ or $\lhd_A$. Now, seeking a contradiction, assume $n \geq 1$. Assume first that $\mathbf{t}_1$ is positive. Then, we cannot have $\mathbf{t}_1 \lhd_{\mathsf{x}_2} \mathbf{t}'_1$ as $\mathrm{post}(\mathbf{t}_1) = \mathrm{new}(\mathbf{t}_1) = \emptyset$. But we also cannot have $\mathbf{t}_1 \lhd_A \mathbf{t}'_1$, as $\mathbf{t}_1$ is visible but not $\mathbf{t}'_1$. Assuming that $\mathbf{t}_2$ is negative is symmetric: we cannot have $\mathbf{t}'_n \lhd_{\mathsf{x}_2} \mathbf{t}_2$ as $\mathrm{pre}(\mathbf{t}_2) = \mathrm{eat}(\mathbf{t}_2) = \emptyset$, and we cannot have $\mathbf{t}'_n \lhd_A \mathbf{t}_2$ because $\mathbf{t}_2$ is visible but not $\mathbf{t}'_n$. So, $n = 0$ and we have $\mathbf{t}_1 \lhd_{\mathsf{x}_2} \mathbf{t}_2$. But again, for the same reason this cannot be because $\mathbf{t}_1 \lhd_{\boldsymbol{\sigma}} \mathbf{t}_2$, so $\mathbf{t}_1 \lhd_A \mathbf{t}_2$, which means $\partial_{\boldsymbol{\sigma}}(\mathbf{t}_1) \rightarrowtriangle_A \partial_{\boldsymbol{\sigma}}(\mathbf{t}_2)$. Hence, $\partial_{\mathscr{U}(\boldsymbol{\sigma})} \rightarrowtriangle_A \partial_{\mathscr{U}(\boldsymbol{\sigma})}$.

Receptive. Consider $x \in \mathscr{C}(\mathscr{U}(\boldsymbol{\sigma}))$ and $\partial_{\mathscr{U}(\boldsymbol{\sigma})}(x) \vdash_A a^-$. So we have $K_{\boldsymbol{\sigma}}(x) \in \mathscr{T}^V(\boldsymbol{\sigma})$ with $\partial_{\boldsymbol{\sigma}}(K_{\boldsymbol{\sigma}}(x)) \vdash_A a^-$. We show that there is a unique matching extension $\mathbf{t}^-$ of $K_{\boldsymbol{\sigma}}(x)$, and conclude by the fact that $K_{\boldsymbol{\sigma}}$ is an order-isomorphism. For *existence*, consider $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha$ a valid run such that $K_{\boldsymbol{\sigma}}(x) = \mathscr{T}(\rho)$. Consider $s \in \mathrm{play}(\rho)$, so in particular $s \in \mathsf{Plays}(A)$. By hypothesis, $sa \in \mathsf{Plays}(A)$ as well. So by condition *receptive* of Petri strategies, there is a unique $\mathbf{t}^- : \emptyset \overset{a^-}{\longmapsto}_{\boldsymbol{\sigma}} \beta$ for some $\beta \cap \alpha = \emptyset$, so that $\rho' = \rho(\mathbf{t}^- \uplus \gamma)$ is a valid run for some γ; providing the expected extension of $K_{\boldsymbol{\sigma}}(x)$. *Uniqueness* follows immediately from uniqueness of $\mathbf{t}^-$.

Negative. Consider $\mathsf{x} \in |\mathscr{U}(\boldsymbol{\sigma})|$ minimal. This means that x is a prime history with exactly one visible itransition $\mathbf{t}$, with $\mathbf{t} = \mathrm{top}(\mathsf{x})$. So there is a run

$$\emptyset \mathbf{t}_1 \dots \mathbf{t}_n (\mathbf{t} \uplus \gamma) : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha\,,$$

but by condition *negative* of Petri strategies, $\mathbf{t}_1$ cannot be neutral and have no preconditions. So $n = 0$, and we have a one-itransition run $\mathbf{t} : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha$. But likewise, by condition *negative* of Petri strategies this entails that $\mathbf{t}$ is negative, so x is negative as required. □

D.2 The Unfolding as a Functor

Proposition 5.15. *Consider $\boldsymbol{\sigma} : A \vdash B$ and $\boldsymbol{\tau} : B \vdash C$ Petri strategies. Then, there is an order-iso:*

$$(- \odot -) : \{(\mathsf{x}^{\boldsymbol{\tau}}, \mathsf{x}^{\boldsymbol{\sigma}}) \in \mathscr{T}^+(\boldsymbol{\tau}) \times \mathscr{T}^+(\boldsymbol{\sigma}) \mid \textit{causally compatible}\} \cong \mathscr{T}^+(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$$

such that for $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}^+(\boldsymbol{\sigma}), \mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}^+(\boldsymbol{\tau})$ causally compatible, $\partial_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}}(\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}) = \mathsf{x}^{\boldsymbol{\sigma}}_A \vdash \mathsf{x}^{\boldsymbol{\tau}}_C$.

Proof. For $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\sigma})$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}(\boldsymbol{\tau})$, we set $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}$ as the set of instantiated transitions obtained from $\mathsf{x}^{\boldsymbol{\sigma}}$ and $\mathsf{x}^{\boldsymbol{\tau}}$ by the rules of Figure 28 (following Lemma C.3). We prove by induction that for all $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\sigma})$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}(\boldsymbol{\tau})$ causally compatible, then $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$, and

$$\mathrm{post}(\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}) = \mathrm{post}(\mathsf{x}^{\boldsymbol{\sigma}}) +^{\odot} \mathrm{post}(\mathsf{x}^{\boldsymbol{\tau}})\,.$$

If $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}$ is empty, there is nothing to prove. If $\mathsf{x}^{\boldsymbol{\sigma}}$ or $\mathsf{x}^{\boldsymbol{\tau}}$ have a maximal neutral instantiated transition, say *w.l.o.g.* that it is $\mathsf{x}^{\boldsymbol{\sigma}}$ with maximal $\mathbf{t} = t^0 (\!| \mu |\!) \in \mathsf{x}^{\boldsymbol{\sigma}}$. Then, setting $\mathsf{y}^{\boldsymbol{\sigma}} = \mathsf{x}^{\boldsymbol{\sigma}} \setminus \{\mathbf{t}\}$ yields $\mathsf{y}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\sigma})$ by Proposition 5.10; and with also $\mathsf{y}^{\boldsymbol{\tau}} = \mathsf{x}^{\boldsymbol{\tau}}$, it is direct that $\mathsf{y}^{\boldsymbol{\sigma}}$ and $\mathsf{y}^{\boldsymbol{\tau}}$ are still causally compatible. By IH, we have $\mathsf{y}^{\boldsymbol{\tau}} \odot \mathsf{y}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$ and $\mathrm{post}(\mathsf{y}^{\boldsymbol{\tau}} \odot \mathsf{y}^{\boldsymbol{\sigma}}) = \mathrm{post}(\mathsf{y}^{\boldsymbol{\sigma}}) +^{\odot} \mathrm{post}(\mathsf{y}^{\boldsymbol{\tau}})$. This means that there is a run $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ projecting to $\rho_{\boldsymbol{\sigma}} : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}}$ with $\alpha_{\boldsymbol{\sigma}} = \mathrm{post}(\mathsf{y}^{\boldsymbol{\sigma}})$. Since $\mathbf{t}$ is enabled in $\mathrm{post}(\mathsf{y}^{\boldsymbol{\sigma}})$ it follows that $\ell^{\odot}(t^0)(\!| \ell^{\odot}(\mu) |\!)$ is enabled in $\mathsf{y}^{\boldsymbol{\tau}} \odot \mathsf{y}^{\boldsymbol{\sigma}}$, and

$$\rho(\ell^{\odot}(t^0)(\!| \ell^{\odot}(\mu) |\!) : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \beta$$

where by construction $\beta = \mathrm{post}(\mathsf{x}^{\boldsymbol{\sigma}}) +^{\odot} \mathrm{post}(\mathsf{x}^{\boldsymbol{\tau}})$ as needed. The symmetric reasoning applies if $\mathsf{x}^{\boldsymbol{\tau}}$ has a maximal neutral instantiated transition – so assume all maximal transitions in $\mathsf{x}^{\boldsymbol{\sigma}}, x^{\boldsymbol{\tau}}$ visible.

Now, by causal compatibility of $\mathsf{x}^{\boldsymbol{\sigma}}$ and $\mathsf{x}^{\boldsymbol{\tau}}$, there is an element of $\mathsf{x}_A \parallel \mathsf{x}_B \parallel \mathsf{x}_C$ (following the notations of Section 5.3.1) which is maximal for $\lhd$. If it is in x_A, it has the form $\partial^{\ell}_{\boldsymbol{\sigma}}(\mathbf{t})$ for $\mathbf{t} \in \mathsf{x}^{\boldsymbol{\sigma}}$ positive or negative. In both cases, the same argument as in the neutral case applies (with the additional observation that the obtained run yields a valid play from the hypothesis). The reasoning is the same if it is in x_C. The last (key) case is if it is in x_B. Then there are instantiated transitions

$$t^{+}\llparenthesis\mu\rrparenthesis : \mu \xmapsto{r_{\vdash} m}_{\boldsymbol{\sigma}} \emptyset\,, \qquad\qquad t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis : \emptyset \xmapsto{\ell_{\vdash} m}_{\boldsymbol{\tau}} \nu\,,$$

or the dual – symmetric – situation, respectively maximal in $\mathsf{x}^{\boldsymbol{\sigma}}$ and $\mathsf{x}^{\boldsymbol{\tau}}$; and by necessity $m = (\mathrm{m}, \mathrm{s}, \mathrm{d})$ where $\partial_{\boldsymbol{\sigma}}(t^{+}) = r_{\vdash}\mathrm{m}$, $\partial_{\boldsymbol{\tau}}(t^{-}) = \ell_{\vdash}\mathrm{m}$, $\delta\langle t^{+}\rangle(\mu) = (\mathrm{s},\mathrm{d})$ and $\delta\langle t^{-}\rangle(\mathrm{s},\mathrm{d}) = \nu$. Setting $\mathsf{y}^{\boldsymbol{\sigma}} \setminus \{t^{+}\llparenthesis\mu\rrparenthesis\}$ and $\mathsf{y}^{\boldsymbol{\tau}} \setminus \{t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis\}$, it is straightforward that they are still causally compatible histories. By IH, $\mathsf{y}^{\boldsymbol{\tau}} \odot \mathsf{y}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$ with $\mathrm{post}(\mathsf{y}^{\boldsymbol{\tau}} \odot \mathsf{y}^{\boldsymbol{\sigma}}) = \mathrm{post}(\mathsf{y}^{\boldsymbol{\sigma}}) +^{\odot} \mathrm{post}(\mathsf{y}^{\boldsymbol{\tau}})$. It follows that there is a run

$$\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \mathrm{post}(\mathsf{y}^{\boldsymbol{\tau}} \odot \mathsf{y}^{\boldsymbol{\sigma}})\,.$$

Since $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\sigma})$ with $t^{+}\llparenthesis\mu\rrparenthesis$ maximal and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}(\boldsymbol{\sigma})$ with $t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis$ maximal, there are

$$\xi_{\boldsymbol{\sigma}}(t^{+}\llparenthesis\mu\rrparenthesis \uplus \gamma_{\boldsymbol{\sigma}}) : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \mathrm{post}(\mathsf{x}^{\boldsymbol{\sigma}})\,, \qquad \xi_{\boldsymbol{\tau}}(t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis \uplus \gamma_{\boldsymbol{\tau}}) : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}} \mathrm{post}(\mathsf{x}^{\boldsymbol{\tau}})\,,$$

valid runs by Lemma D.3, with $\mathsf{x}^{\boldsymbol{\sigma}} = \mathsf{IT}_{\xi_{\boldsymbol{\sigma}}}$ and $\mathsf{x}^{\boldsymbol{\tau}} = \mathsf{IT}_{\xi_{\boldsymbol{\tau}}}$. By Lemma D.1, $\mathrm{post}(\xi_{\boldsymbol{\sigma}}) = \mathrm{post}(\rho_{\boldsymbol{\sigma}}) = \mathrm{post}(\mathsf{y}^{\boldsymbol{\sigma}})$ and $\mathrm{post}(\xi_{\boldsymbol{\tau}}) = \mathrm{post}(\rho_{\boldsymbol{\tau}}) = \mathrm{post}(\mathsf{y}^{\boldsymbol{\tau}})$. It follows that the transition

$$(t^{+} \circledast t^{-})\llparenthesis\ell^{\odot}(\mu)\rrparenthesis : \ell^{\odot}(\mu) \longmapsto_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} r^{\odot}(\nu)$$

is enabled in $\mathrm{post}(\mathsf{y}^{\boldsymbol{\tau}} \odot \mathsf{y}^{\boldsymbol{\sigma}})$, hence it can be appended to ρ, witnessing $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$.

In the other direction, given $\mathsf{y} \in \mathscr{T}(\boldsymbol{\tau} \odot \boldsymbol{\sigma})$, consider a valid run $\rho : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}\odot\boldsymbol{\sigma}} \alpha$ such that $\mathsf{y} = \mathsf{IT}_{\rho}$. By Lemmas C.4 and C.6, we then have $\alpha = \alpha_{\boldsymbol{\sigma}} +^{\odot} \alpha_{\boldsymbol{\tau}}$ with

$$\rho_{\boldsymbol{\sigma}} : \emptyset \twoheadrightarrow_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}}\,, \qquad \rho_{\boldsymbol{\tau}} : \emptyset \twoheadrightarrow_{\boldsymbol{\tau}} \alpha_{\boldsymbol{\tau}}$$

valid runs. Recall that $\rho_{\boldsymbol{\sigma}} = \rho \upharpoonright \pi_{\boldsymbol{\sigma}}$ and $\rho_{\boldsymbol{\tau}} = \rho \upharpoonright \pi_{\boldsymbol{\tau}}$ – hence, setting $\mathsf{x}^{\boldsymbol{\sigma}} = \pi_{\boldsymbol{\sigma}}(\mathsf{y})$ and $\mathsf{x}^{\boldsymbol{\tau}} = \pi_{\boldsymbol{\tau}}(\mathsf{y})$, we have $\mathsf{x}^{\boldsymbol{\sigma}} = \mathsf{IT}_{\rho_{\boldsymbol{\sigma}}}$ and $\mathsf{x}^{\boldsymbol{\tau}} = \mathsf{IT}_{\rho_{\boldsymbol{\tau}}}$ so that $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}(\boldsymbol{\sigma})$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}(\boldsymbol{\tau})$. Causal compatibility is direct as ρ provides a linearization of $\lhd$.

It is direct that these constructions are inverse; it remains to show that they preserve +-covered histories. If $\mathsf{x}^{\boldsymbol{\sigma}}$ and $\mathsf{x}^{\boldsymbol{\tau}}$ causally compatible are +-covered, then consider $\mathbf{t}$ maximal in $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}$. If $\mathbf{t} = \ell^{\odot}(t^{0})\llparenthesis\ell^{\odot}(\mu)\rrparenthesis$, this directly contradicts +-coveredness of $\mathsf{x}^{\boldsymbol{\sigma}}$, and likewise for $r^{\odot}(t^{0})\llparenthesis r^{\odot}(\mu)\rrparenthesis$. If $\mathbf{t} = \ell^{\odot}(t^{-})\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis$, then $t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis$ is maximal in $\mathsf{x}^{\boldsymbol{\sigma}}$, contradiction – likewise for $r^{\odot}(t^{-})\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis$. If $\mathbf{t} = (t^{+} \circledast t^{-})\llparenthesis\ell^{\odot}(\mu)\rrparenthesis$ with $t^{+}\llparenthesis\mu\rrparenthesis \in \mathsf{x}^{\boldsymbol{\sigma}}$ and $t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis \in \mathsf{x}^{\boldsymbol{\tau}}$, then $t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis$ is maximal in $\mathsf{x}^{\boldsymbol{\tau}}$, contradiction – likewise, $\mathbf{t} = (t^{-} \circledast t^{+})\llparenthesis r^{\odot}(\mu)\rrparenthesis$ leads to a contradiction. So, $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}$ is +-covered.

Reciprocally, assume $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}$ +-covered. Consider $\mathbf{t} \in \mathsf{x}^{\boldsymbol{\sigma}}$ maximal. If $\mathbf{t} = t^{0}\llparenthesis\mu\rrparenthesis$, $\ell^{\odot}(t^{0})\llparenthesis\ell^{\odot}(\mu)\rrparenthesis$ is maximal in $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}$, contradiction. If $\mathbf{t} = t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis$, then since $\mathsf{x}^{\boldsymbol{\tau}} \odot \mathsf{x}^{\boldsymbol{\sigma}}$ is +-covered, there is

$$\ell^{\odot}(t^{-})\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis \rightarrowtriangle_{\mathsf{x}^{\boldsymbol{\tau}}\odot\mathsf{x}^{\boldsymbol{\sigma}}} \mathbf{t}'$$

and a direct case analysis shows that $\pi_{\boldsymbol{\sigma}}\mathbf{t}'$ is defined with $t^{-}\llparenthesis(\mathrm{s},\mathrm{d})\rrparenthesis \rightarrowtriangle_{\mathsf{x}^{\boldsymbol{\sigma}}} \pi_{\boldsymbol{\sigma}}\mathbf{t}'$, contradicting the maximality of $\mathbf{t}$. The last case has $\mathbf{t}$ positive; and symmetrically, $\mathsf{x}^{\boldsymbol{\tau}}$ is +-covered. □

As detailed in Proposition 5.16, it follows that unfolding preserves composition up to iso.

Next, we show the same for copycat. If A is an arena, then we have obvious bijections

$$\mathsf{IT}^{+}_{\mathbf{æ}_A} \cong |A \vdash A|^{+} \qquad\qquad \mathsf{IT}^{-}_{\mathbf{æ}_A} \cong |A \vdash A|^{-}\,,$$

and coercing silently through these, we have:

Lemma D.5. *Consider A an arena, and $\rho : \emptyset \twoheadrightarrow_{\mathbf{æ}_A} \alpha$ a valid run.*
Then, $\mathsf{IT}_{\rho} = \mathsf{play}(\rho)$, with negative maximal transitions in bijection with α.

Proof. Straightforward by induction on ρ. □

Lemma D.6. *Consider A an arena. Then, we have the order-isomorphism*

$$\mathscr{T}^+(\mathbf{c\!c}_A) \cong \{x \vdash x \mid x \in \mathscr{C}(A)\}$$

with $\partial_{\mathbf{c\!c}_A}(x \vdash x) = x \vdash x$.

Proof. The isomorphism simply applies the bijection $\mathsf{IT}_{\mathbf{c\!c}_A} \simeq |A|$. From left to right, recall first that by Lemma C.8, for $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\mathbf{c\!c}_A} \alpha$ a valid run, we have

$$|\mathsf{play}(\rho)| = x \vdash y\,, \qquad y \supseteq^- x \cap y \subseteq^+ x\,, \qquad \alpha = (y^- \setminus x) \uplus (x^+ \setminus y)\,.$$

By Lemma D.5, the history IT_ρ is $+$-covered iff $\alpha = \emptyset$, *i.e.* $y^- \subseteq x$ and $x^+ \subseteq y$. But as $y \supseteq^- x \cap y \subseteq^+ x$ this entails $x = y$. In that case, $\partial_{\mathbf{c\!c}_A}(\mathsf{IT}_\rho) = |\mathsf{play}(\rho)| = x \vdash x$ as needed. Reciprocally, for any $x \in \mathscr{C}(A)$, it is straightforward to build a valid run $\rho : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\mathbf{c\!c}_A} \emptyset$ s.t. $\mathsf{IT}_\rho = x \vdash x$ as required. □

From that, preservation of copycat follows:

Proposition D.7. *Consider A an arena. Then,* $\mathscr{U}(\mathbf{c\!c}_A) \cong \mathrm{c\!c}_A$.

Proof. We compose label-preserving order-isomorphisms:

$$\begin{array}{rcl}
\mathscr{C}^+(\mathscr{U}(\mathbf{c\!c}_A)) & \cong & \mathscr{T}^+(\mathbf{c\!c}_A) \\
& \cong & \{x \vdash x \mid x \in \mathscr{C}(A)\} \\
& \cong & \mathscr{C}^+(\mathrm{c\!c}_A)
\end{array}$$

by Lemmas 5.13 and D.6. From this it follows that $\mathscr{U}(\mathbf{c\!c}_A) \cong \mathrm{c\!c}_A$ by Lemma 4.9. □

Corollary D.8. *We have a functor of precategories* $\mathscr{U} : \mathsf{PStrat} \to \mathsf{Strat}$.

D.3 The Unfolding Preserves Operations

Next, we prove that the unfolding preserves all operations of the IPA-structure.

D.3.1 Tensor. Preservation of the tensor operation is easy via the following observation:

Lemma D.9. *Consider $\boldsymbol{\sigma} : A_1 \vdash B_1, \boldsymbol{\tau} : A_2 \vdash B_2$ Petri strategies. Then, we have an order-isomorphism*

$$(- \otimes -) : \mathscr{T}^+(\boldsymbol{\sigma}) \times \mathscr{T}^+(\boldsymbol{\tau}) \cong \mathscr{T}^+(\boldsymbol{\sigma} \otimes \boldsymbol{\tau})$$

s.t. $\partial_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}}(\mathsf{x}^{\boldsymbol{\sigma}} \otimes \mathsf{x}^{\boldsymbol{\tau}}) = (x_{A_1} \otimes x_{A_2}) \vdash (x_{B_1} \otimes x_{B_2})$ *where* $\partial_{\boldsymbol{\sigma}}(\mathsf{x}^{\boldsymbol{\sigma}}) = x_{A_1} \vdash x_{B_1}$ *and* $\partial_{\boldsymbol{\tau}}(\mathsf{x}^{\boldsymbol{\tau}}) = x_{A_2} \vdash x_{B_2}$.

Proof. Consider $\mathsf{x}^{\boldsymbol{\sigma}} \in \mathscr{T}^+(\boldsymbol{\sigma})$ and $\mathsf{x}^{\boldsymbol{\tau}} \in \mathscr{T}^+(\boldsymbol{\tau})$. By definition, there are valid runs

$$\rho^{\boldsymbol{\sigma}} : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha_{\boldsymbol{\sigma}}\,, \qquad\qquad \rho^{\boldsymbol{\tau}} : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\tau}} \alpha_{\boldsymbol{\tau}}$$

such that $\mathsf{x}^{\boldsymbol{\sigma}} = \mathsf{IT}_{\rho^{\boldsymbol{\sigma}}}$ and $\mathsf{x}^{\boldsymbol{\tau}} = \mathsf{IT}_{\rho^{\boldsymbol{\tau}}}$. We define the history $\mathsf{x}^{\boldsymbol{\sigma}} \otimes \mathsf{x}^{\boldsymbol{\tau}}$ as

$$\begin{array}{rcl}
\mathsf{x}^{\boldsymbol{\sigma}} \otimes \mathsf{x}^{\boldsymbol{\tau}} & = & \{\ell^\otimes(t^{0,+})(\!|\ell^\otimes(\mu)|\!) \mid t^{0,+}(\!|\mu|\!) \in \mathsf{x}^{\boldsymbol{\sigma}}\,\} \\
& \uplus & \{\ell^\otimes(t^-)(\!|(\mathsf{s},\mathsf{d})|\!) \mid t^-(\!|(\mathsf{s},\mathsf{d})|\!) \in \mathsf{x}^{\boldsymbol{\sigma}}\,\} \\
& \uplus & \{r^\otimes(t^{0,+})(\!|r^\otimes(\mu)|\!) \mid t^{0,+}(\!|\mu|\!) \in \mathsf{x}^{\boldsymbol{\tau}}\} \\
& \uplus & \{r^\otimes(t^-)(\!|(\mathsf{s},\mathsf{d})|\!) \mid t^-(\!|(\mathsf{s},\mathsf{d})|\!) \in \mathsf{x}^{\boldsymbol{\tau}}\}\,.
\end{array}$$

This must be the history of a valid run – to show that, we build

$$\ell^\otimes(\rho^{\boldsymbol{\sigma}}) : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \ell^\otimes(\alpha_{\boldsymbol{\sigma}})\,, \qquad r^\otimes(\rho^{\boldsymbol{\tau}}) \uplus \ell^\otimes(\alpha_{\boldsymbol{\sigma}}) : \ell^\otimes(\alpha_{\boldsymbol{\sigma}}) \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \ell^\otimes(\alpha_{\boldsymbol{\sigma}}) \uplus r^\otimes(\alpha_{\boldsymbol{\tau}})$$

which by concatenation (and Lemma C.13) yields a valid run $\rho^{\boldsymbol{\sigma}} \otimes \rho^{\boldsymbol{\tau}} : \emptyset \longrightarrow\!\!\!\!\!\rightarrow_{\boldsymbol{\sigma}\otimes\boldsymbol{\tau}} \alpha_{\boldsymbol{\sigma}} +^\otimes \alpha_{\boldsymbol{\tau}}$; and it is immediate that $\mathsf{x}^{\boldsymbol{\tau}} \otimes \mathsf{x}^{\boldsymbol{\sigma}} = \mathsf{IT}_{\rho^{\boldsymbol{\tau}}\otimes\rho^{\boldsymbol{\sigma}}}$. By definition of the causal ordering of instantiated transitions, it is also immediate that $\mathsf{x}^{\boldsymbol{\tau}} \otimes \mathsf{x}^{\boldsymbol{\sigma}}$ is $+$-covered; and that this preserves the labelling.

Reciprocally, for any $\mathrm{x} \in \mathscr{T}^+(\boldsymbol{\sigma} \otimes \boldsymbol{\tau})$ we consider the projections

$$\mathrm{x}^{\sigma} = \pi_{\sigma}(\mathrm{x})\,, \qquad \mathrm{x}^{\tau} = \pi_{\tau}(\mathrm{x})\,,$$

and it follows from Lemma C.14 that $\mathrm{x}^{\sigma} \in \mathscr{T}(\boldsymbol{\sigma})$ and $\mathrm{x}^{\tau} \in \mathscr{T}(\boldsymbol{\tau})$. From the definition of the causal ordering of instantiated transitions, x^{σ} and x^{τ} are still +-covered.

Finally, these two transformations are inverses as required. □

Again, from this we can conclude that the unfolding preserves the tensor.

Corollary D.10. *Consider $\boldsymbol{\sigma} : A_1 \vdash B_1$, $\boldsymbol{\tau} : A_2 \vdash B_2$ Petri strategies. Then, we have $\mathscr{U}(\boldsymbol{\sigma} \otimes \boldsymbol{\tau}) \cong \mathscr{U}(\boldsymbol{\sigma}) \otimes \mathscr{U}(\boldsymbol{\tau})$.*

Proof. We compose label-preserving isomorphisms:

$$\begin{array}{rcl}
\mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma} \otimes \boldsymbol{\tau})) & \cong & \mathscr{T}^+(\boldsymbol{\sigma} \otimes \boldsymbol{\tau}) \\
& \cong & \mathscr{T}^+(\boldsymbol{\sigma}) \times \mathscr{T}^+(\boldsymbol{\tau}) \\
& \cong & \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma})) \times \mathscr{C}^+(\mathscr{U}(\boldsymbol{\tau})) \\
& \cong & \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma}) \otimes \mathscr{U}(\boldsymbol{\tau}))
\end{array}$$

by Lemmas 5.13, D.9, Lemma 5.13 again, and Proposition B.5. □

D.3.2 Renaming. Before detailing the unfolding of currying and promotion, we show that it preserves renaming. We have already established in Lemma C.19 that (global) renamings preserve valid runs. In order for renamings to preserve the unfolding, we must ensure that the dependency between instantiated transitions is preserved as well.

Lemma D.11. *Consider A, B games, $h = [f, (g_{\mathrm{m}})] : A \curvearrowright B$, $\boldsymbol{\sigma} : A$, and $\rho : \emptyset \twoheadrightarrow_{\sigma} \alpha$ valid. For all $\mathbf{t}, \mathbf{t}' \in \mathsf{IT}_{\rho}$, we have $\mathbf{t} \leq_{\rho} \mathbf{t}'$ iff $\mathbf{t}[h] \leq_{\rho[h]} \mathbf{t}'[h]$.*

Proof. Consider $\mathbf{t} \rightarrowtriangle_{\rho} \mathbf{t}'$. *W.l.o.g.* we assume that this dependency cannot be deduced otherwise by transitivity. By Lemma D.4, we can assume that $\mathbf{t}$ and $\mathbf{t}'$ appear subsequently in ρ.

Assume first $\mathbf{t} \rightarrowtriangle_A \mathbf{t}'$. By definition,

$$\mathbf{t} : \alpha \xmapsto{a}_{\sigma} \beta\,, \qquad \mathbf{t}' : \alpha' \xmapsto{a'}_{\sigma} \beta'$$

with $a \rightarrowtriangle_A a'$, while by construction, $\mathbf{t}[h] : \alpha \xmapsto{ha}_{\sigma[h]} \beta$ and $\mathbf{t}'[h] : \alpha' \xmapsto{ha'}_{\sigma[h]} \beta'$. Now, we distinguish cases depending on the polarity of a, a'. If $\mathrm{pol}_A(a) = +$ or $\mathrm{pol}_A(a') = -$, then by *courtesy* we have $ha \rightarrowtriangle_B ha'$, so that $\mathbf{t}[h] \rightarrowtriangle_B \mathbf{t}'[h]$ by Definition 5.8. If $\mathrm{pol}_A(a) = -$ and $\mathrm{pol}_A(a') = +$, then

$$\mathbf{t} = t^-(\!|(\mathrm{s}, \mathrm{d})|\!) : \emptyset \xmapsto{a}_{\sigma} \beta\,, \qquad \mathbf{t}' = t^+(\!|\alpha'|\!) : \alpha' \xmapsto{a'}_{\sigma} \emptyset\,.$$

Assume, seeking a contradiction, that $\beta \cap \alpha' = \emptyset$, and consider the prefix of ρ:

$$\rho'(t^-(\!|(\mathrm{s}, \mathrm{d})|\!) \uplus \gamma)(t^+(\!|\alpha'|\!) \uplus \gamma') : \emptyset \twoheadrightarrow_{\sigma} \nu\,,$$

but if indeed $\beta \cap \alpha' = \emptyset$, then $\mathbf{t}$ and $\mathbf{t}'$ permute as in

$$\rho'(t^+(\!|\alpha'|\!) \uplus \mu)(t^-(\!|(\mathrm{s}, \mathrm{d})|\!) \uplus \mu') : \emptyset \twoheadrightarrow_{\sigma} \nu$$

and by *valid*, this entails $\mathsf{play}(\rho')a' \in \mathsf{Plays}(A)$, contradicting $a \rightarrowtriangle_A a'$. Assume now $\mathbf{t} \rightarrowtriangle_{\sigma} \mathbf{t}'$. This comes either from $\mathsf{new}(\mathbf{t}) \cap \mathsf{pre}(\mathbf{t}') \neq \emptyset$, or $\mathsf{post}(\mathbf{t}) \cap \mathsf{eat}(\mathbf{t}') \neq \emptyset$. But the renaming of instantiated transtions does not change pre- and post-conditions, so $\mathbf{t}[h] \rightarrowtriangle_{\rho[h]} \mathbf{t}'[h]$ still.

Reciprocally, assume $\mathbf{t}[h] \leq_{\rho[h]} \mathbf{t}'[h]$. Seeking a contradiction, assume $\neg(\mathbf{t} \leq_{\rho} \mathbf{t}')$. By Lemma D.4, we can assume that $\mathbf{t}'$ appears before $\mathbf{t}$ in ρ. Hence, $\mathbf{t}'[h]$ appears before $\mathbf{t}[h]$ in $\rho[h]$. But by Lemma C.19 $\rho[h]$ is valid, so by Lemma D.4 $\mathbf{t}[h]$ must appear before $\mathbf{t}'[h]$, contradiction. □

Next, we need to show that valid runs are also *reflected* by renamings:

Lemma D.12. *Consider A, B games, $\boldsymbol{\sigma} : A$, $h = [f, (g_{\mathsf{m}})] : A \curvearrowright B$, and $\rho' : \emptyset \twoheadrightarrow_{\sigma[h]} \alpha$ valid. Then, there is a unique $\rho : \emptyset \twoheadrightarrow_{\sigma} \alpha$ valid such that $\rho' = \rho[h]$.*

Proof. By induction on ρ'. If it is empty, this is clear. Consider $\rho'\mathsf{t}^0$ with $\rho' : \emptyset \twoheadrightarrow_{\sigma[h]} \alpha$ and $\mathsf{t}^0 = t^0(\!|\mu|\!) \uplus \gamma : \alpha \longrightarrow_{\sigma[h]} \beta$, with $t^0(\!|\mu|\!) : \mu \longmapsto_{\sigma[h]} \nu$. By IH, there is $\rho : \emptyset \twoheadrightarrow_{\sigma} \alpha$. By definition, we still have $t^0(\!|\mu|\!) \uplus \gamma : \alpha \longrightarrow_{\sigma} \beta$, so $\rho\mathsf{t}^0 : \emptyset \twoheadrightarrow_{\sigma} \beta$ and as required, $(\rho\mathsf{t}^0)[h] = \rho'\mathsf{t}^0$.

Next, consider $\rho'\mathsf{t}^+$ with $\rho' : \emptyset \twoheadrightarrow_{\sigma[h]} \alpha$ and $\mathsf{t}^+ = t^+(\!|\mu|\!) \uplus \gamma : \alpha \longrightarrow_{\sigma[h]} \beta$, with $t^+(\!|\mu|\!) : \mu \overset{b'}{\longmapsto}_{\sigma[h]} \emptyset$. Necessarily, $b' = (f(\mathsf{m}), \mathsf{d}', \mathsf{s}')$ for $\mathsf{m} = \partial_{\sigma}(t^+)$, and $(\mathsf{d}', \mathsf{s}') = g_{\mathsf{m}}(\mathsf{d}, \mathsf{s})$ for $(\mathsf{d}, \mathsf{s}) = \delta\langle t^+\rangle(\mu)$ – so $b = h(b)$ for $b = (\mathsf{m}, \mathsf{d}, \mathsf{s})$. But then, by definition, $t^+(\!|\mu|\!) : \mu \overset{b}{\longmapsto}_{\sigma} \emptyset$ as well, so $\mathsf{t}^+ : \alpha \longrightarrow_{\sigma} \beta$. By IH, there is $\rho : \emptyset \twoheadrightarrow_{\sigma} \alpha$ such that $\rho[h] = \rho'$. By *valid*, $\rho\mathsf{t}^+$ is still valid, and $(\rho\mathsf{t}^+)[h] = \rho'\mathsf{t}^+$.

Finally, consider $\rho'\mathsf{t}^-$ with $\rho' : \emptyset \twoheadrightarrow_{\sigma[h]} \alpha$ and $\mathsf{t}^- = t^-(\!|(\mathsf{s}', \mathsf{d}')|\!) \uplus \gamma : \alpha \longrightarrow_{\sigma[h]} \beta$, with $t^-(\!|(\mathsf{s}', \mathsf{d}')|\!) : \emptyset \overset{b'}{\longmapsto}_{\sigma[h]} \nu$. Here, necessarily, $b' = (\mathsf{m}', \mathsf{s}', \mathsf{d}')$ where $\mathsf{m}' = f(\mathsf{m})$, $\mathsf{m} = \partial_{\sigma}(t^-)$, $(\mathsf{s}', \mathsf{d}') = g_{\mathsf{m}}(\mathsf{s}, \mathsf{d})$. In other words, $b' = h(b)$ with $b = (\mathsf{m}, \mathsf{s}, \mathsf{d})$. Now, by IH we have $\rho : \emptyset \twoheadrightarrow_{\sigma} \alpha$ with $\rho[h] = \rho'$. Besides, we have the transition $t^-(\!|(\mathsf{s}, \mathsf{d})|\!) : \emptyset \overset{b}{\longmapsto}_{\sigma} \nu$, so that $(t^-(\!|(\mathsf{s}, \mathsf{d})|\!) \uplus \gamma) : \alpha \longrightarrow_{\sigma} \beta$. Its validity follows immediately from *receptivity* of global renamings (and the fact that they are injective). It is clear that $(\rho(t^-(\!|(\mathsf{s}, \mathsf{d})|\!) \uplus \gamma))[h] = \rho'\mathsf{t}^-$ as required.

Finally, uniqueness is immediate by induction and injectivity of h. □

Lemma D.13. *Consider A, B games, $\boldsymbol{\sigma} : A$ a Petri strategy, and $h = [f, (g_{\mathsf{m}})] : A \curvearrowright B$. Then, $\mathscr{U}(\boldsymbol{\sigma}[h]) \cong \mathscr{U}(\boldsymbol{\sigma})[h]$.*

Proof. We construct an order-isomorphism

$$-[h] : \mathscr{T}^+(\boldsymbol{\sigma}) \cong \mathscr{T}^+(\boldsymbol{\sigma}[h]) \tag{5}$$

such that $\partial_{\sigma[h]}(\mathsf{x}) = h(\partial_{\sigma}(\mathsf{x}))$ for all $\mathsf{x} \in \mathscr{T}^+(\boldsymbol{\sigma})$. Given $\mathsf{x} \in \mathscr{T}^+(\boldsymbol{\sigma})$, there is some ρ a valid run in $\boldsymbol{\sigma}$ such that $\mathsf{x} = \mathscr{T}(\rho)$. By Lemma C.19, $\rho[h]$ is a valid run of $\boldsymbol{\sigma}[h]$, so we may consider $\mathsf{x}[h] = \mathscr{T}(\rho[h])$. By Lemma D.11, $-[h]$ on instantiated transitions is an order-isomorphism $-[h] : \mathsf{x} \cong \mathsf{x}[h]$, so $\mathsf{x}[h] \in \mathscr{T}^+(\boldsymbol{\sigma}[h])$. Together with Lemma D.12, this easily entails that we get $-[h] : \mathscr{T}(\boldsymbol{\sigma}) \cong \mathscr{T}(\boldsymbol{\sigma}[h])$ an order-isomorphism. By definition, for each $\mathsf{x} \in \mathscr{T}(\boldsymbol{\sigma})$ we have an order-iso $\mathsf{x} \cong \mathsf{x}[h]$ defined by applying $-[h]$ on each transition – thus, $-[h]$ preserves and reflects +-covered histories. The requirement *w.r.t.* labels is obvious by construction.

By Lemma 5.13, we also obtain an order-isomorphism

$$-[h] : \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma})) \cong \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma}[h]))$$

such that $\partial_{\mathscr{U}(\sigma[h])}(x[h]) = h(\partial_{\mathscr{U}(\sigma)}(x))$ for any $x \in \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma}))$, so, from Definition 3.14, an order-isomorphism $-[h] : \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma})[h]) \cong \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma}[h]))$ such that $\partial_{\mathscr{U}(\sigma[h])}(x[h]) = \partial_{\mathscr{U}(\sigma)[h]}(x)$ for all $x \in \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma})[h])$. By Lemma 5.13, it follows that $\mathscr{U}(\boldsymbol{\sigma})[h]$ and $\mathscr{U}(\boldsymbol{\sigma}[h])$ are isomorphic. □

D.3.3 Currying. Follows from Lemma D.13, as currying is obtained with the same global renaming both in PStrat and in Strat.

D.3.4 Promotion. Let us start with characterizing +-covered traces of the functorial promotion.

Lemma D.14. *Consider $\boldsymbol{\sigma} : A \vdash B$ a Petri strategy. Then, we have an order-iso*

$$[-] : \mathsf{Fam}(\mathscr{T}^{+,\neq\emptyset}(\boldsymbol{\sigma})) \cong \mathscr{T}^+(!\boldsymbol{\sigma})$$

satisfying that for all $(\mathsf{x}^{\mathsf{e}})_{\mathsf{e}\in E} \in \mathsf{Fam}(\mathscr{T}^{+,\neq\emptyset}(\boldsymbol{\sigma}))$, we have

$$\partial_{!\sigma}([(\mathsf{x}^{\mathsf{e}})_{\mathsf{e}\in E}]) = \left(\biguplus_{\mathsf{e}\in E} \mathsf{e} :: x_A^e\right) \vdash \left(\biguplus_{\mathsf{e}\in E} \mathsf{e} :: x_B^e\right)$$

writing $\partial_{\sigma}(\mathsf{x}^e) = x_A^{\mathsf{e}} \vdash x_B^{\mathsf{e}}$ for all $\mathsf{e} \in E$.

Proof. This is a n-ary adaptation of the proof of Lemma D.9. Consider $(\mathsf{x}^{\mathsf{e}})_{\mathsf{e}\in E} \in \mathsf{Fam}(\mathscr{T}^{+,\neq\emptyset}(\boldsymbol{\sigma}))$. By definition, there is a valid run

$$\rho^{\mathsf{e}} : \emptyset \longrightarrow\!\!\!\!\rightarrow_{\boldsymbol{\sigma}} \alpha^{\mathsf{e}}$$

for all $\mathsf{e} \in E$ such that $\mathsf{x}^{\mathsf{e}} = \mathsf{IT}_{\rho^{\mathsf{e}}}$. We define the history $[(\mathsf{x}^{\mathsf{e}})_{\mathsf{e}\in E}]$ as

$$\begin{array}{rcl} [\mathsf{x}^{\mathsf{e}} \mid \mathsf{e} \in E] & = & \{t^0(\!|\mathsf{e} :: \alpha|\!) \mid \mathsf{e} \in E, t^0(\!|\alpha|\!) \in \mathsf{x}^{\mathsf{e}}\} \\ & \uplus & \{t^+(\!|\mathsf{e} :: \alpha|\!) \mid \mathsf{e} \in E, t^+(\!|\alpha|\!) \in \mathsf{x}^{\mathsf{e}}\} \\ & \uplus & \{t^-(\!|(\mathsf{e} :: \mathsf{s}, \mathsf{d})|\!) \mid \mathsf{e} \in E, t^-(\!|(\mathsf{s}, \mathsf{d})|\!) \in \mathsf{x}^{\mathsf{e}}\}\,. \end{array}$$

We construct a run ρ obtained by concatenating all ρ^{e}s in the obvious way as in Lemma D.9. Exploiting Lemma C.27, it is a valid run and $[\mathsf{x}^{\mathsf{e}} \mid \mathsf{e} \in E] = \mathsf{IT}_\rho$ by construction. By definition of the causal ordering of instantiated transitions, it is also immediate that $[\mathsf{x}^{\mathsf{e}} \mid \mathsf{e} \in E]$ is +-covered; and that this preserves the labelling. Reciprocally, for any $\mathsf{x} \in \mathscr{T}(!\boldsymbol{\sigma})$, we consider the projections

$$\mathsf{x}^{\mathsf{e}} = \pi^!_{\mathsf{e}}(\mathsf{x})$$

and it follows from Lemma C.28 that $\mathsf{x}^{\mathsf{e}} \in \mathscr{T}(\boldsymbol{\sigma})$ for all $\mathsf{e} \in E$. From the definition of the causal ordering of instantiated transitions, each x^{e} is still +-covered.

Finally, these two transformations are inverses as required. □

Corollary D.15. *Consider $\boldsymbol{\sigma} : !A \vdash B$ a Petri strategy. Then, we have $\mathscr{U}(\boldsymbol{\sigma}^\dagger) \cong \mathscr{U}(\boldsymbol{\sigma})^\dagger$.*

Proof. We exploit the following sequence of label-preserving order-isomorphisms:

$$\begin{array}{rcl} \mathscr{C}^+(\mathscr{U}(\boldsymbol{\sigma}^\dagger)) & \cong & \mathscr{T}^+(\boldsymbol{\sigma}^\dagger) \\ & = & \mathscr{T}^+((!\boldsymbol{\sigma})[\mathsf{dig} \vdash \mathsf{id}]) \\ & \cong & \mathscr{T}^+(!\boldsymbol{\sigma}) \\ & \cong & \mathsf{Fam}(\mathscr{T}^{+,\neq\emptyset}(\boldsymbol{\sigma})) \\ & \cong & \mathsf{Fam}(\mathscr{C}^{+,\neq\emptyset}(\mathscr{U}(\boldsymbol{\sigma}))) \end{array}$$

using first Lemma 5.13, then via a direct verification as in Proposition C.32, then applying (5), followed by Lemma D.14, and then Lemma 5.13 – with the obvious verification that it specializes to an iso between non-empty configurations and histories. It is a direct verification that this sequence of isomorphisms preserves display maps.

Consequently, it follows that $\mathscr{U}(\boldsymbol{\sigma}^\dagger) \cong \mathscr{U}(\boldsymbol{\sigma})^\dagger$ from Proposition B.9. □

D.4 The Unfolding Preserves Primitives

D.4.1 Variable and Evaluation. Follows from Lemma D.13.

D.4.2 Queries, Conditional, Constants. It is a simple calculation to compute the unfolding for these linear Petri strategies and check we obtain the desired finite strategy.

D.4.3 Fixpoint. We prove the following proposition:

Proposition D.16. *For any well-opened arena O, $\mathscr{U}(\mathrm{Y}_O) \cong \mathrm{Y}_O$.*

Proof. Using Lemmas 4.9 and 5.13, the required isomorphism boils down to an order-iso

$$\mathscr{T}^+(\mathrm{Y}_O) \cong \mathscr{C}^+(\mathrm{Y}_O)$$

preserving display maps. We build it using Lemmas C.35 and B.15. It is clearly injective so we simply have to show it is surjective. Consider a +-covered configuration x of Y_O represented as a tuple $\langle J, z, (y_{\mathsf{s}})_{\mathsf{s}\in J}\rangle$. We can construct a set of itransitions realising x as follows:

- For each $(\mathsf{m}, \mathsf{s}, d)^- \in z$ we include the itransition $\not{r}_{\vdash}\mathsf{m}(\!|(\mathsf{s}, \mathsf{d})|\!)$, and $\ell_{\vdash}\not{r}_{\multimap}(\!|\{(\ell_!\blacklozenge :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^-}\}|\!)$

- For each $(\mathsf{m}, \mathsf{s}, \mathsf{d})^+ \in z$, we include the itransitions $\ell_{\vdash} r_{\multimap} (\!|(\ell_! \blacklozenge :: \mathsf{s}, \mathsf{d})|\!)$ and $r_{\vdash}\mathsf{m}(\!|\{(\ell_! \blacklozenge :: \mathsf{s}, \mathsf{d})^{@\mathsf{m}^+}\}|\!)$
- For each $(\mathsf{m}, \mathsf{s}_0, \mathsf{d})^- \in y_{\mathsf{e}\cdot\mathsf{s}}$, we include the itransitions $r_{\vdash} r_{\multimap}\mathsf{m}(\!|(\mathsf{e} :: (\mathsf{s}) :: \mathsf{s}_0, \mathsf{d})|\!)$ and $r_{\vdash} r_{\multimap}\mathsf{m}(\!|\{((\mathsf{e}\cdot \mathsf{s}) :: \mathsf{s}_0, \mathsf{d})^{@\mathsf{m}^-}\}|\!)$.
- For each $(\mathsf{m}, \mathsf{s}_0, \mathsf{d})^+ \in y_{\mathsf{e}\cdot\mathsf{s}}$, we include the itransitions $r_{\vdash} \ell_{\multimap}\mathsf{m}(\!|((\mathsf{e}\cdot \mathsf{s}) :: \mathsf{s}_0, \mathsf{d})|\!)$ and $r_{\vdash} \ell_{\multimap}\mathsf{m}(\!|\{((\mathsf{e}\cdot \mathsf{s}) :: \mathsf{s}_0, \mathsf{d})^{@\mathsf{m}^+}\}|\!)$,

and it is easy to see that this set of transitions is reachable by a valid run of Y_O. □

D.4.4 Contraction. The reasoning follows a similar and simpler route as for the fixpoint operator.

D.4.5 Let bindings. We first characterise the configurations of the strategy interpreting lets.

Lemma D.17. *The +-covered configurations of* **let** *are order-isomorphic to tuples* $\langle I \subseteq \mathcal{E}, x \in \mathscr{C}(\mathbf{X}), y, y' \in \mathscr{C}(\mathbf{Y})\rangle$ *such that:*

1. $y \neq \emptyset$ *iff* $x \neq \emptyset$
2. x *is maximal iff* $y' \neq \emptyset$
3. *if* $y' \neq \emptyset$*, then* $y = y'$.
4. *if* $y' = \emptyset$*, then* $I = \emptyset$.

The isomorphism sends such tuples to $((\uplus_{\mathsf{e}\in I}(\mathsf{e} :: x)) \multimap y') \otimes x \vdash y$.

Lemma D.18. *We have* $\mathscr{U}(\mathbf{let}) \cong \mathbf{let}$.

Proof. As before we rely on Lemmas 4.9 and 5.13 to build the isomorphism. Lemma C.37 together with Lemma D.17 induce an injective map from $\mathscr{T}^+(\mathbf{let})$ into $\mathscr{C}^+(\mathbf{let})$. We show it is surjective by constructing a set of itransitions of **let** from $x \in \mathscr{C}^+(\mathbf{let})$ corresponding to a $\langle I, x, y, y'\rangle$.

- If $y \neq \emptyset$, then we have itransitions $r_{\vdash}\mathsf{Q}^-(\!|([], \bullet)|\!)$ and $\ell_{\vdash} r_{\otimes}\mathsf{Q}(\!|\{([], \bullet)^{@1}|\!)$
- If x has an event $([], \mathsf{d})$ maximal in $\mathbf{X}$, then we have the two itransitions $\ell_{\vdash} r_{\otimes}\mathsf{A}^-(\!|([], \mathsf{d})|\!)$, and $\ell_{\vdash}\ell_{\otimes} r_{\multimap}\mathsf{Q}^+(\!|\{([], \bullet)^{@2}\}|\!)$
- For every $\mathsf{e} \in I$, we have three itransitions $\ell_{\vdash}\ell_{\otimes}\ell_{\multimap}\mathsf{Q}^-(\!|([\mathsf{e}], \bullet)|\!)$, $\mathsf{s}(\!|\{([], \bullet)^{@4}, ([], \mathsf{d})^{@3}\}|\!)$, and $\ell_{\vdash}\ell_{\otimes}\ell_{\multimap}\mathsf{A}^+(\!|\{([\mathsf{e}], \mathsf{d})^{@5}\}|\!)$ where d is the value of the maximal event in x.
- If y' has a maximal $([], \mathsf{d})$, we have itransitions $\ell_{\vdash}\ell_{\otimes} r_{\multimap}\mathsf{A}^-(\!|([], \mathsf{d})|\!)$ and $r_{\vdash}\mathsf{A}^+(\!|\{([], \mathsf{d}]^{@6}\}|\!)$. □

D.4.6 Newref and newsem. We now show that the unfolding of the net for newref is indeed the strategy newref. Our first step is to show that the consistent memory traces described in Section C.3.6 correspond to +-covered configurations of **newref**:

Lemma D.19. *There is an order-isomorphism between* $\mathscr{C}^+(\mathsf{precell})$ *and the set of consistent memory traces ordered by prefix.*

Proof. Direct from the definition of precell. □

We now show the main result:

Proposition D.20. $\mathscr{U}(\mathbf{newref}) \cong \mathsf{newref}$.

Proof. By Lemmas 4.9 and 5.13, this amounts to building an order-iso:

$$\mathscr{T}^+(\mathbf{newref}) \cong \mathscr{C}^+(\mathsf{newref}).$$

From left-to-right. We focus on non-empty histories and configurations and use characterisation of $\mathscr{C}^+(\mathsf{newref})$ from Proposition B.19.

Consider a non-empty history $\mathrm{y} \in \mathscr{T}^+(\mathbf{newref})$. It is reached by a run $\rho : \emptyset \xrightarrow{s}\!\!\!\!\rightarrow \alpha$. By Lemma C.39, we know that there $\mathsf{Tr}(\rho)$ is a consistent memory trace, and that $|s|$ must have the shape $((\uplus_{\mathsf{e}\in I}(\mathsf{e} :: x_{\mathsf{e}})) \multimap w) \vdash z$. Since y is +-covered, we observe that $w = z$. We can thus map y to $\langle \mathsf{Tr}(\rho), w\rangle \in \mathscr{C}^+(\mathsf{newref})$ using the isomorphism of Proposition B.19. Note that by the side conditions of Lemma C.39, the family $(x_{\mathsf{e}})_{\mathsf{e} \in I}$ is entirely determined by $\mathsf{Tr}(\rho)$: I matches the length of $\mathsf{Tr}(\rho)$ and each x_{e} is a two-event configuration corresponding to the memory operation $\mathsf{Tr}(\rho)_{\mathsf{e}}$.

The last check it to show that this does not depend on the particular run ρ chosen. Clearly x only depends on $|s|$. For $\mathsf{Tr}(\rho)$, we observe that it is actually directly recoverable from the set of transitions y. First, we define the set of memory operations O_{y} to contain $(w, \mathsf{e}, \mathsf{d})$ if $\mathsf{w}\{([\mathsf{e}], \mathsf{d})^{@3}, \_\} \in \mathsf{y}$ and $(r, \mathsf{e}, \mathsf{d})$ if $\mathsf{r}\{([\mathsf{e}], \_)^{@5}, (\_, \mathsf{d})^{@2}\} \in \mathsf{y}$. Then, the causal order on y induces a linear order on O_{y} due to the threading of exponential signatures. The resulting trace is exactly $\mathsf{Tr}(\rho)$.

From right-to-left. Consider now a $\langle \rho, x \rangle \in \mathscr{C}^{+,\neq\emptyset}(\mathsf{newref})$ where ρ is a consistent memory trace. We can build a history y containing the following instantiated transitions:

- The initial negative itransition on $\mathscr{r}_{\vdash}\mathsf{Q}(\!|([], \bullet)|\!)$.
- The positive itransition $\ell_{\vdash}\ell_{\multimap}\mathsf{Q}(\!|\{([], \bullet)^{@1}\}|\!)$.
- If x contains a move $(\mathsf{A}, [], d)$, then the itransitions $\ell_{\vdash}\mathscr{r}_{\multimap}\mathsf{A}^{-}(\!|([], \mathsf{d})|\!)$ and $\mathscr{r}_{\vdash}\mathsf{A}^{+}(\!|\{([], \mathsf{d})^{@7}\}|\!)$.
- If $\rho_i = (w, \mathsf{e}, \mathsf{d})$ then the itransitions $\ell_{\vdash}\ell_{\multimap}\mathscr{w}_{\mathbb{V}}\mathsf{Q}^{-}(\!|([\mathsf{e}], \mathsf{d})|\!)$, $\mathsf{w}(\!|\{([\mathsf{e}], \mathsf{d})^{@3}, \{[\mathsf{e}'], \mathsf{d}'\}^{@2}\}|\!)$ where: e' is the exponential token of ρ_{i-1} (or $[]$ if $i = 0$), and d' the value observed by ρ_{i-1} (or zero if $i = 0$); and $\ell_{\vdash}\ell_{\multimap}\mathscr{w}_{\mathbb{V}}\mathsf{A}^{+}(\!|\{([\mathsf{e}], \checkmark)^{@4}\}|\!)$.
- And similarly if $\rho_i = (r, \mathsf{e}, \mathsf{d})$.

From this description, it is easy to build a valid run reaching y establishing that $\mathsf{y} \in \mathscr{T}(\mathbf{newref})$. An easy verification shows that all maximal itransitions in y are positive. $\square$