amU
Aix Marseille Université

NNT : 2025AIXM0435

THÈSE DE DOCTORAT

Soutenue à AMU – Aix-Marseille Université
le 4 décembre 2025 par

Lison BLONDEAU-PATISSIER

Jeux concurrents à pointeurs et calcul à ressource
Pointer Concurrent Games and the Resource Calculus

Discipline
Mathematics

École doctorale
Mathématiques et informatique (ED 184)

Laboratoire
Institut de mathématiques de Marseille
(I2M, UMR 7373)
Laboratoire d'Informatique et des Systèmes
(LIS, UMR 7020)

Composition du jury

Giulio MANZONETTO Professeur des universités, Paris Cité	Rapporteur
Guy McCUSKER Professor of Computing, University of Bath	Rapporteur
Claudia FAGGIAN Chargée de recherche, Paris Cité	Examinatrice
Marie KERJEAN Chargée de recherche, Sorbonne Paris Nord	Examinatrice
Stefano GUERRINI Professeur des universités, Sorbonne Paris Nord	Président du jury
Lionel VAUX AUCLAIR Maître de conférences, Aix-Marseille Université	Directeur de thèse
Pierre CLAIRAMBAULT Directeur de recherche, Aix-Marseille Université	Co-directeur de thèse

INSTITUT
DE MATHÉMATIQUES
DE MARSEILLE

LIS
LABORATOIRE D'INFORMATIQUE & SYSTÈMES

Pointer Concurrent Games and the Resource Calculus

Disclaimer
This page is based on the corresponding page of Ken Arroyo Ohori's thesis, with minimal changes.

No copyright
🄍🄌 This book is released into the public domain using the CC0 code.
To the extent possible under law, I waive all copyright and related or neighbouring rights to this work.

To view a copy of the CC0 code, visit `http://creativecommons.org/publicdomain/zero/1.0/`.

Colophon
This document was typeset with the help of KOMA-Script and LaTeX using the kaobook class.

Affidavit

I, undersigned, Lison Blondeau-Patissier, hereby declare that the work presented in this manuscript is my own work, carried out under the scientific supervision of Pierre Clairambault and Lionel Vaux Auclair, in accordance with the principles of honesty, integrity and responsibility inherent to the research mission. The research work and the writing of this manuscript have been carried out in compliance with both the french national charter for Research Integrity and the Aix-Marseille University charter on the fight against plagiarism.

This work has not been submitted previously either in this country or in another country in the same or in a similar version to any other examination body.

Lyon, 09/09/25

Lison Blondeau-Patissier

Liste de publications et participation aux conférences

Liste des publications réalisées dans le cadre du projet de thèse :

Revues

1. *Strategies as Resources Terms, and Their Categorical Semantics*, Lison Blondeau-Patissier, Pierre Clairambault and Lionel Vaux Auclair. (Version longue de 3.)
   Logical Methods in Computer Science (LMCS), 2025.
   `https://doi.org/10.46298/lmcs-21(4:9)2025`.

Actes de conférences

2. *Positional Injectivity for Innocent Strategies*, Lison Blondeau-Patissier and Pierre Clairambault.
   Formal Structures for Computation and Deduction (FSCD), 2021.
   `https://doi.org/10.4230/LIPIcs.FSCD.2021.17`.
3. *Strategies as Resource Terms, and Their Categorical Semantics*, Lison Blondeau-Patissier, Pierre Clairambault and Lionel Vaux Auclair.
   Formal Structures for Computation and Deduction (FSCD), 2023.
   `https://doi.org/10.4230/LIPIcs.FSCD.2023.13`.
4. *Resource Categories from Differential Categories*, Lison Blondeau-Patissier.
   Journées Francophones des Langages Applicatifs (JFLA), 2024.
   `https://inria.hal.science/hal-04406440`.

Pré-publications

5. *Extensional Taylor Expansion*, Lison Blondeau-Patissier, Pierre Clairambault and Lionel Vaux Auclair.
   2025. `https://doi.org/10.48550/arXiv.2305.08489`.

Participation aux conférences et écoles d'été pendant la période de thèse :

Présentations lors de conférences

1. Formal Structures for Computation and Deduction (FSCD).
   Juillet 2021, en ligne. `https://fscd2021.dc.uba.ar`.
   Exposé : *Positional Injectivity for Innocent Strategies*.
2. Journées du GT Scalp.
   Novembre 2021, Fontainebleau. `https://www.irif.fr/gt-scalp/journees-2021`.
   Exposé : *Positional Injectivity for Innocent Strategies*.
3. Formal Structures for Computation and Deduction (FSCD).
   Juillet 2023, Roma (Italie). `https://easyconferences.eu/fscd2023`.
   Exposé : *Strategies as Resource Terms, and Their Categorical Semantics*.
4. Séminaire Chocola.
   Septembre 2023, Lyon. `https://chocola.ens-lyon.fr/events/meeting-2023-09-28`.
   Exposé : *Strategies as Resource Terms, and Their Categorical Semantics*,

5. Workshop GALOP.
   Janvier 2024, London (Royaume-Uni). https://popl24.sigplan.org/home/galop-2024.
   Exposé : *Taylor Expansion is Game Semantics*.
6. Journées Francophones des Langages Applicatifs (JFLA).
   Janvier 2024, Saint-Jacut-de-la-Mer. https://jfla.inria.fr/jfla2024.html.
   Exposé : *Resource Categories from Differential Categories*.

Poster

1. Journées Nationales du GDR IM.
   Mars-Avril 2022, Lille. https://jnim2022.sciencesconf.org.
   Poster : *An Extensional Resource lambda-calculus and its Categorical Semantics*.

Autres – Écoles et conférences suivies

1. Rencontres mensuelles *Chocola*,
   2021–2025, Lyon. https://chocola.ens-lyon.fr.
2. Mois thématique 2022 *Logique et Interactions*, Marseille :
   - *École d'hiver de logique linéaire*. https://conferences.cirm-math.fr/2685.html.
   - *Logique de la programmation probabiliste*. https://conferences.cirm-math.fr/2686.html.
   - *Logique et structures supérieures*. https://conferences.cirm-math.fr/2689.html.
3. *Journées du GT Scalp*,
   Février 2023, Marseille. https://conferences.cirm-math.fr/2992.html.
   Novembre 2023, Orléans. https://www.irif.fr/gt-scalp/journees-2023.
4. *λ-calcul différentiel et logique linéaire différentielle, 20 ans après*,
   Mai 2024, Marseille. https://conferences.cirm-math.fr/2980.html.
5. *Avancées en Sémantiques Interactives et Quantitatives*,
   Mai 2025, Marseille. https://conferences.cirm-math.fr/3518.html.

Résumé et mots clés

Cette thèse présente les jeux concurrents à pointeurs, et étudie les liens entre la sémantique des jeux d'une part et le λ-calcul à ressources d'autre part.

On s'intéresse tout d'abord aux liens entre sémantique des jeux et modèle relationnel. On commence par introduire un nouveau modèle de jeux, les *jeux concurrents à pointeurs* (PCG). Ce modèle s'inspire à la fois des jeux HO traditionnels et des jeux concurrents. On établit une bijection entre les *augmentations* (quotientées par isomorphisme) dans PCG et les *parties* (quotientées par homotopie) des *stratégies innocentes* dans HO. Ce modèle nous permet d'obtenir un premier résultat d'injectivité positionnelle dans PCG, qui se traduit en un résultat d'injectivité positionnelle pour les stratégies innocentes, *finies* et *totales* dans HO. On montre également que les stratégies innocentes *partielles infinies* ne sont *pas* positionnellement injectives.

On introduit ensuite le calcul à ressources *extensionnel*, c'est-à-dire typé de façon à ce que les termes en forme normale soient également en forme η-longue. Ces termes sont en bijection avec les classes d'isomorphisme d'augmentations dans PCG.

On peut maintenant s'intéresser à l'aspect *dynamique* de la sémantique. On définit une opération de *composition* dans PCG, et on montre que PCG est une catégorie symétrique monoïdale fermée. La correspondance entre PCG et HO s'étend en un foncteur cartésien fermé strict.

Pour étudier l'interprétation du calcul à ressources dans PCG, on cherche à exprimer plus précisément sa structure catégorique. Pour cela, on introduit les *catégories à ressources*, inspirées des catégories différentielles. On définit l'interprétation du calcul à ressources dans une catégorie à ressources, et on montre qu'elle est compatible avec la β-réduction. PCG forme une catégorie à ressources, dans laquelle l'interprétation du calcul à ressources coïncide avec la bijection établie précédemment pour les termes en forme normale.

Mots clés : Sémantique dénotationnelle.
Sémantique des jeux → innocence, positions, jeux Hyland-Ong, jeux concurrents à pointeurs.
λ-calcul → calcul à resources.
Sémantique catégorique → catégories à ressources.

Abstract and keywords

This thesis presents the *Pointer Concurrent Games* model. We study the links between game semantics and resource λ-calculus.

First, we focus on the links between game semantics and relational semantics. We introduce a new game model, *pointer concurrent games* (or PCG), inspired by traditional HO games and by concurrent games. There is a bijection bewteen *augmentations* (up to isomorphism) in PCG and *plays* (up to homotopy) of *innocent strategies* in HO. We obtain a first result of positional injectivity in PCG, which translates to a result of positional injectivity for *total finite* innocent strategies in HO. We also prove that *partial infinite* innocent strategies are *not* positionaly injective.

Next we introduce the *extensional* resource calculus, *i.e.* a typed resource calculus where typing rules ensure that terms in normal form are also in η-long form. These terms are in bijection with the augmentations (up to isomorphism) in PCG.

We can now consider the *dynamic* aspect of the semantics. We define the *composition* in PCG, and we show that PCG is a closed symetric monoidal category. The correspondance between PCG and HO is extended to a strict cartesian closed functor.

Finally, in order to study the interpretation of the resource calculus in PCG, we try and describe more precisely its categorical structure by introducing *resource categories* – inspired by differential categories. We construct a *sound* interpretation of the resource calculus in a resource category, and we show that PCG is indeed a resource category. Moreover, this interpretation coincides with the bijection for *normal* resource terms.

Keywords: Denotational semantics.
Game semantics → innocence, positions, Hyland-Ong games, pointer concurrent games.
λ-calculus → resource calculus.
Categorical semantics → resource categories.

Contents

Introduction

A brief outline. Game semantics is a formal model of the execution of programs. It may be more correct to write "are formal models", as there are many different game models; this thesis focuses both on the standard HO games and on a new model, *Pointer Concurrent Games* (or PCG for short). We study correspondences between several ways of approximating the behavior of programs, trying to better understand how they are related; in particular we will describe some links between PCG, HO, the relational model, and the resource calculus – as well as expose the categorical structure of PCG.

What is *(game) semantics* anyway?

What is this thesis about? Despite the fact that "Mathematics" is written on the title page of this document, it is at heart as much about computer science as about mathematics – or rather, it is about the mathematical structures of programs and computations. The *semantics* of programming languages is the study of formal properties of the executions of programs, by modeling executions with mathematical objects in order to better understand their structure and properties.

Consider for instance the interaction between a user and their calculator presented in Figure 1.

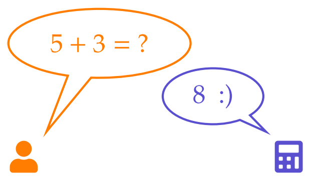


Figure 1: A simple example.

They are several ways to model this computation.

Operational semantics focuses on the *operations* performed during the computation by giving formal rules on syntax, modeling for example the step-by-step execution of a program.

For instance, we can represent natural numbers with:

$$\mathtt{n},\mathtt{m} \in \mathtt{Nat} ::= \mathtt{0} \mid \mathtt{S}\,\mathtt{n}\,.$$

Notation: The line above is an inductive definition of the elements of `Nat` (given in *Backus–Naur form*, or BNF for short). It means that the elements of `Nat` are written `n` or `m`, and that they are either `0` or of the form `S n` where `n` is an element of `Nat`.

Here `0` represents the natural number 0, and `S n` is the *successor* of the natural number n represented by `n` (*i.e.* `S n` represents $n+1$).

Notation: For any $n \in \mathbb{N}$, we write $\underline{n}$ as a shortcut for $\underbrace{\mathtt{S}\ldots\mathtt{S}}_{n \text{ times}}\ \mathtt{0}$.

Then $\underline{n}$ represents $0 \underbrace{+1+\ldots+1}_{n \text{ times}}$, *i.e.* the number n.

Going back to our addition example, the calculator uses the rules presented in Figure 2 to perform the step-by-step computation of Figure 3.

```
n+0      =  n
n+(S m)  =  (S n)+m
```

Figure 2: Formal rules for the addition in `Nat`.

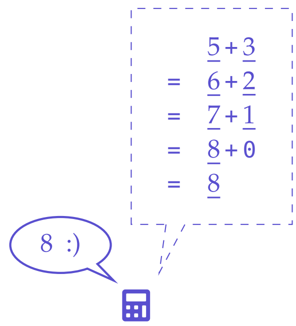


Figure 3: Step-by-step computation.

Denotational semantics is the translation of a program M to a mathematical object $[\![M]\!]$, its *denotation*. Here a program is seen as a function, whose arguments are the inputs of the program.

For the addition example, we could write for instance:

$$[\![\_+\_]\!] : \begin{array}{ccc} \mathbb{N} \times \mathbb{N} & \rightarrow & \mathbb{N} \\ (n, m) & \mapsto & n + m \end{array}$$

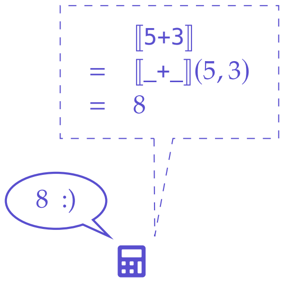


Figure 4: Denotational point of view.

and represent the computation $5 + 3 = 8$ as in Figure 4.

This way of modelling execution has several advantages; it does not depend as much on a specific syntax, and it allows us to study *compositional properties* of programs (for instance, we might want $[\![f(x)]\!] = [\![f]\!]([\![x]\!])$).

Historical context. Game semantics is a denotational semantics that arose in the early 90's from the problem of "full abstraction for PCF", *i.e.* the question of whether all observationally equivalent programs in PCF (*Programming Computable Functions*, a typed fonctional language – see [37] or [36]) have equal denotations.

[37]: Scott (1993), 'A Type-Theoretical Alternative to ISWIM, CUCH, OWHY'

[36]: Plotkin (1977), 'LCF Considered as a Programming Language'

Some of the first fully abstract models[1] for PCF are game models: HO/N games, or HO games, independently introduced in [27] and in [35]; and AJM games, introduced in [1]. Since then there have been lots of developpements in this line of work, and there are numerous other models – involving non-determinism, concurrency, *etc.* – but in this thesis, we focus only on HO games (with inspiration from concurrent games).

1: **Remark:** We use *model* as a synonym for *semantics*, most of the time implicitely *denotational semantics*.

[27]: Hyland and Ong (2000), 'On Full Abstraction for PCF: I, II, and III'

[35]: Nickau (1994), 'Hereditarily Sequential Functionals'

[1]: Abramsky, Jagadeesan, and Malacaria (2000), 'Full Abstraction for PCF'

Intuitions on games. Game semantics models programs as processes, focusing on the interactions between the program and its environment. These interactions are represented as plays in a *game* between two protagonists, one of them called **Player** representing the program and the other called **Opponent** representing the environment.

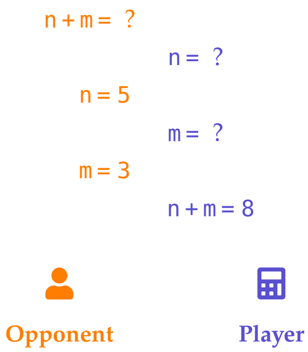


Figure 5: A *play* for "5+3 = 8"

In our example from Figure 1, Player would be the calculator and Opponent the user. Their interaction could be represented by the dialogue in Figure 5

Information tokens exchanged between Opponent and Player are called *moves*. The "rules" of the game, *i.e.* which moves are available and when, are given by an annotated tree structure called the *arena* – it corresponds to the *type* of the program. An example of such a rule could be: *when Player asks* n = ?, *Opponent can only respond with an integer value*, like n = 5, because the type of the addition program here is $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$.

A *play* represents one possible execution of a program; for instance Figure 5 represents one execution of the addition program, where Opponent wants to compute 5 + 3.

Programs themselves are represented by *strategies*, which are sets of plays corresponding to every possible execution of the program. The strategy representing the addition program would include plays corresponding to any computation of n+m for any n, m; as well as executions where Opponent decides to stop the computation, or to repeat (part of) it.

About Calculi

λ-calculus. Introduced by Church in the 30's (see [14] for the historical reference, or [2] for a more detailed introduction), the λ-calculus is a formal programming language, where programs are *terms* of the form:

[14]: Church (1940), 'A Formulation of the Simple Theory of Types'

[2]: Barendregt (1984), *The lambda calculus - its syntax and semantics*

$$\begin{array}{llll} M, N, L, \ldots & ::= & x & \textit{(variable)} \\ & \mid & \lambda x.M & \textit{(abstraction)} \\ & \mid & M\ N & \textit{(application)} \end{array}$$

An abstraction $\lambda x.M$ represents a function "$x \mapsto M$" – *i.e.* a program which asks for some input x and then executes a subprogram M.
An application $M\ N$ represents a function application "$M(N)$" – *i.e.* a subprogram M called with the subprogram N as its input.

Given a program $\lambda x.M$ and an input N, one step of the execution of the program $(\lambda x.M)\ N$ is to compute "$M[N/x]$", the λ-term written like M but where each occurrence of x is substituted by a copy of N. This operation is the *β-reduction*, written $(\lambda x.M)\ N \to_\beta M[N/x]$.

Example: $(\lambda x.y\ x\ x)\ N \to_\beta y\ N\ N$.

Notation (priority rules): The abstraction always captures the largest possible term, *i.e.* the term $\lambda x.M\ N$ is to be read as $\lambda x.(M\ N)$. The application is left-associative, *i.e.* the term $M\ N\ L$ is to be read as $(M\ N)\ L$. Hence the term $(\lambda x.y\ x\ x)\ N$ is $(\lambda x.((y\ x)\ x))\ N$.

Resource calculus. The *resource calculus*, on the other hand, arose from linear logic (introduced in [24]) and quantitative semantics ([25]). Unlike the usual λ-calculus, the resource calculus sees terms as *resources* which can each be used exactly once. Hence, the substitution is not defined with a single argument term N anymore, but rather with a multiset of terms $[N_1, \ldots, N_n]$, which will each replace exactly one occurence of x in M – the exact bijection being chosen non-deterministically, *via* a sum of resource terms corresponding to all possible substitutions.

[24]: Girard (1987), 'Linear logic'

[25]: Girard (1988), 'Normal functors, power series and λ-calculus'

Example: $(\lambda x.y\,[x]\,[x])\ [N_1, N_2] \to_\beta y\,[N_1]\,[N_2] + y\,[N_2]\,[N_1]$.

This allows for a control of the number of copies of N, and for example ensures that the reduction terminates. Replacing arguments with multisets of terms in λ-calculi first emerged with the λ-calculus with multiplicities [10], the term *resource* appearing a few years later in [11].

[10]: Boudol (1993), 'The lambda-calculus with multiplicities'

[11]: Boudol, Curien, and Lavatelli (1999), 'A semantics for lambda calculi with resources'

Bridging the gap between models

Motivation. Both game semantics and resource calculus have been well-studied lines of work for years, and both of them consider (sets of) finite executions of programs to represent programs with possibly infinite behavior. It is only natural to try and formalize a correspondence between the two of them.

We are also interested in the connections between games and the relational model, another semantics of programming languages.

Of course, there is a practical motivation for this work. The more we know about how to go from one model to another, the easier it becomes to translate well-known properties from one model to the other without having to prove them from scratch. It allows us to choose which setting to work in when trying to prove new results, in order to work with the model that is best suited for the proof techniques we want to use.

But above all, understanding links between models gives us a better understanding of the models themselves, and of the actual programs they seek to represent. Each model showcases different particularities of the computational behavior of programs, so if we gain a better understanding of their similarities and of their divergences, we come closer to grasping the "true" computational behavior of programs.

State of the art

Relational model. While game semantics focuses on the dynamic aspect of programs and their composition, there exist more static models, such as *relational semantics* (see [25] for one of its first (implicit) appearances).

[25]: Girard (1988), 'Normal functors, power series and λ-calculus'

In the relational model $\mathsf{Rel}_!$, types are sets and programs are relations between those sets – more precisely, between finite multisets over the set for the input type and subsets of the set for the output type.

Going back to the addition example, we might write:

$$[\![\_+\_]\!] \subseteq \mathcal{M}_f(\mathbb{N}) \times \mathcal{M}_f(\mathbb{N}) \times \mathbb{N}$$

where $\mathcal{M}_f(X)$ is the set of finite multisets over X. In particular, we have:

$$([5],[3],8) \in [\![\_+\_]\!] \,.$$

Remark that we recognize the moves `n=5`, `m=3` and `n+m=8` of Figure 5.

More generally, the elements of the relational model can be seen as desequentialized plays (see [9]), *i.e.* plays where the chronological information is forgotten. We call the operation from plays in HO to positions in $\mathsf{Rel}_!$ the *relational collapse*.

[9]: Boudes (2009), 'Thick Subtrees, Games and Experiments'

Surely this temporal information, once forgotten, is gone and cannot be recovered? It is true in general if you just look at plays, but when one looks at strategies things are a bit different.

Innocent strategies. *Innocence* is a key notion in game semantics: innocent strategies are strategies whose behavior does not change if Opponent duplicates moves, or chooses to perform some moves in a different order. They correspond to λ-terms, or programs, without mutable states.

Example: Consider a mystery program , of type $\mathbb{N} \to \mathbb{N}$. We know the execution from Figure 6 happened. Now, if is innocent*, then it must always return `res=1` no matter how many times Opponent plays the move `n=3` – because without mutable states, it has no way of storing the information "this is the second time Opponent plays `n=3`".

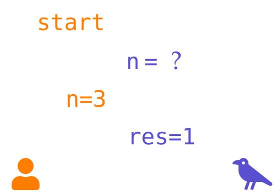


Figure 6: One execution of .

* Of course, given that three crows make a murder, their innocence is unlikely.

In a way, innocent strategies already “forget” part of the temporal information in a play, as they act only according to the current thread. Hence it makes sense to focus on the connections between *innocent* strategies in games and the relational model.

Positionality in asynchronous games. A strategy is *positional* if its behavior only depends on the current position (the moves that have been played so far, without chronological order), and not the sequence of positions reached in the computation. In Melliès’ asynchronous games, innocent strategies are positional [33, Theorem 2]. However, this correspondence is made possible by the fact that events carry explicit copy indices, that help distinguish duplications of the same move.

[33]: Melliès (2006), ‘Asynchronous games 2: The true concurrency of innocence’

What about HO games? In HO games, collapsing strategies to $\mathsf{Rel}_!$ gives us a set of positions; but it is not clear if innocent strategies in HO are still positional without the help of copy indices.

We can also consider the weaker property of positional injectivity: is an innocent strategy characterized by its set of positions? Results on the injectivity of the relational model for linear logic presented in [18] suggest that some temporal information can be recovered from the structure; and indeed Tsukada and Ong show an injective collapse from a category of innocent strategies onto the relational model in [40]. However, their interpretation of the base type α in $\mathsf{Rel}_!$ is a countably infinite set X, which allows them to label moves in each play in order to encode some causal links in the interpretation – but then we lose the correspondence between plays and points of the web in relational semantics.

[18]: de Carvalho (2016), ‘The Relational Model Is Injective for Multiplicative Exponential Linear Logic’

[40]: Tsukada and Ong (2016), ‘Plays as Resource Terms via Non-idempotent Intersection Types’

Question 1: Can we obtain a similar result of positional injectivity *without* this labeling, interpreting α with a singleton?

Resource terms and HO games. As stated previously, HO games and resource terms both consider (sets of) finite executions of programs to represent programs with possibly infinite behavior.

Tsukada and Ong showed in [40] that (β-normal, η-long) resource terms are in bijection with plays of HO games up to *Melliès’* ***homotopy*** *equivalence*. This homotopy relation, defined in [33], equates plays quotiented by Opponent’s scheduling, *i.e.* the order in which Opponent duplicates moves or starts a new thread.

[33]: Melliès (2006), ‘Asynchronous games 2: The true concurrency of innocence’

Example: See the plays from Figure 7: Opponent can choose to play `n=3` first and `n=4` second, or the reverse – and if is innocent then Player’s reaction to both of these moves does not depend on the order of Opponent’s inputs. The second play is just the first one with the pairs of moves (`n=3`, `res=1`) and (`n=4`, `res=42`) switched.

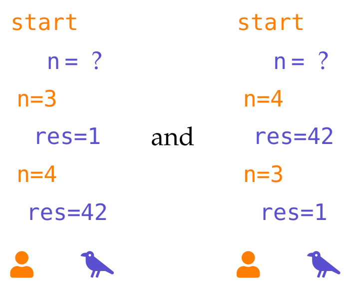


Figure 7: Two homotopic plays.

However, Tsukada and Ong’s correspondence is not direct, going through their aforementioned relational collapse; moreover it does not detail the dynamical aspect of the interpretation.

Question 2: Can we understand the correspondence between games and normal resource terms in a more direct way?

Question 3: What about interpreting *non* normal resource terms?

Contributions

Static Pointer Concurrent Games. PCG originated from the wish to better understand the links between $\mathsf{Rel}_!$ and HO. When working with innocent strategies, we often look at plays *quotiented by homotopy*, because Opponent's scheduling is not relevant to the behavior of the strategy. To focus on the "relevant" part of the chronological information, we defined **augmentations** as the main object of PCG, rather than plays. They correspond to plays quotiented by homotopy, and they encode only causal links that informs us on the causal structure of the strategy.

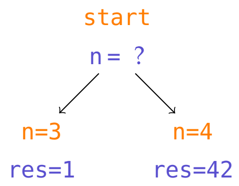


Figure 8: Augmentation from Figure 7.

Example: The two plays from Figure 7 correspond to the same augmentation, drawn in Figure 8.

This construction is inspired by the modern approach of *concurrent games* – see [15] for a detailed presentation. Augmentations retain the causal structure of the strategy, as well as *pointers* following the rules of the arena – one can think of pointers as the links between bound variables and abstractions in λ-calculus. In Figure 8 for instance, there are pointers from n=3 to n= ?, and from n=4 to n= ?.

[15]: Clairambault (2024), 'Causal Investigations in Interactive Semantics'

Main results [Chapter 3]**:**

- A new, synthetic formulation of plays up to Melliès' homotopy, which offers a nice framework to work with innocent strategies and causal structures;
- The detailed correspondence between innocent strategies in HO and sets of augmentations in PCG.

Positional injectivity. This framework allows us to answer **Question 1** – or at least part of it. Using a proof technique inspired from [18], we show that *total finite* innocent strategies are positionally injective. We also show that in general, *partial infinite* innocent strategies are *not* positionally injective, as we exhibit a counter-example. These results are both obtained in PCG and then translated to HO games.

[18]: de Carvalho (2016), 'The Relational Model Is Injective for Multiplicative Exponential Linear Logic'

Main result [Chapter 4]**:**

- Total finite innocent strategies in HO games (and their counterparts in PCG) are positionally injective.

Augmentations and normal resource terms. Since we wish to study the links between the resource calculus and PCG, we start by looking at augmentations and normal resource terms. The correspondence between plays and terms featured in [40] relied on the relational collapse; we show a more direct construction of augmentations from β-normal η-long resource terms, answering **Question 2**.

[40]: Tsukada and Ong (2016), 'Plays as Resource Terms via Non-idempotent Intersection Types'

Main result [Chapter 5]**:**

- Normal, η-long resource terms are isomorphic to (some isomorphism classes of) augmentations.

Dynamics of PCG. Before answering **Question 3**, we need to extend our game model with a *composition* – otherwise, PCG is barely a denotational semantics at all! Suppose we want to compose an augmentation q on an arena $\mathsf{A} \vdash \mathsf{B}$ with an augmentation p on $\mathsf{B} \vdash \mathsf{C}$ (if these notations make no sense for now, think of the usual composition of functions $g \circ f$, with $f : A \to B$ and $g : B \to C$ for A, B, C some sets). We study the *interactions* between q and p occuring in B the shared arena component. Because augmentations are not chronologically ordered, there can be several ways to synchronize q and p so that they agree on what is happening in B. Hence, the composition $p \odot q$ is not a single augmentation, but a *sum* of augmentations over every possible synchronizations. This is reminiscent of the sum produced by the substitution in the resource calculus.

In HO games, strategies are sets of plays, and we know that augmentations are related to plays. Hence it makes sense that strategies in PCG would be some kind of objects representing "gathering several augmentations together". Unlike in HO however, strategies in PCG are not sets of augmentations but rather sums with coefficients. This quantitative aspect is important for future works: by taking coefficients into account in our setting, we lay out the foundations to link PCG with other quantitative models, for instance ones with probabilities. Moreover, resource terms are obtained from usual λ-terms *via* the *Taylor expansion*, an operation which transforms a λ-term into a sum of resource terms; we want coefficients in PCG to be able to match those obtained *via* the Taylor expansion.

We now have a model with strategies and composition, which allows us to study the categorical structure of PCG. Models of linear logic are symmetric monoidal closed categories (or smcc's for short), and the resource calculus is inspired by linear logic; so we might expect PCG to be at least a smcc if we are to find an interpretation of the resource calculus in PCG – and indeed, we prove that it is a smcc.

Forgetting coefficients for a short time, we check that our correspondence between innocent strategies in HO and (sets of) augmentations in PCG is compatible with the composition – yielding a strict cartesian closed functor between (the quantitative fragment of) PCG and HO.

Main results [Chapter 6]**:**

- A notion of composition taking into account coefficients and the different ways to synchronize two augmentations;
- PCG is a category (with arenas as objects and strategies as morphisms) with a closed symmetric monoidal structure;
- The correspondence between PCG and HO defined in Chapter 3 is extended into a strict cartesian closed functor.

Categorical Structure and Resource Categories. In order to study the interpretation of resource terms in PCG, we want to gain a better understanding of its structure. We would like a categorical framework enabling the characterization of morphisms behaving "linearly" in PCG, to show that these morphisms are in bijection with resource terms.

A first candidate would be the symmetric monoidal closed structure, but this is not enough to properly define which morphisms in PCG correspond to resource terms.

A second idea is to look as *differential categories*. Indeed, another extension of λ-calculus inspired by linear logic is the *differential λ-calculus*, defined in [20]. The name differential comes from the *differentiation* operation in analysis. There, functions are approximated by their *derivatives*, obtained through differentiation – the *Taylor expansion* of a function is the sum of its derivatives. By analogy with analysis, the Taylor expansion of a λ-terms in the differential λ-calculus is the sum of its approximants, obtained thanks to a formal differential operator. In [22], the authors present the resource calculus as a sub-language of differential calculus.

[20]: Ehrhard and Regnier (2003), 'The differential lambda-calculus'

[22]: Ehrhard and Regnier (2008), 'Uniformity and the Taylor expansion of ordinary lambda-terms'

[7]: Blute, Cockett, and Seely (2006), 'Differential categories'

Differential categories were introduced in [7] as a categorical framework for differential linear logic. Does that mean we need to look for a differential categorical structure in PCG? Actually, strategies in games *do not* have a linear behavior in general – so this approach is doomed to failure.

Our solution was to define **resource categories**, a categorical structure built using similar constructions to differential categories (as presented in [8]). Resource categories allow both for "non-linear" morphisms in general (which are needed if we want PCG to be a resource category) and for the characterisation of some "linear" morphisms (which will be the target of the interpretation of resource terms).

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Main results [Chapter 7]**:**

- A new categorical framework, featuring a few equations expressing the behavior of morphisms corresponding to resource terms;
- A *sound* interpretation for the resource calculus;
- The links between resource categories and the notion of codereliction in differential categories.

PCG and the resource calculus. We check that PCG is indeed a closed resource category. We can finally answer **Question 3**: resource terms in general can be interpreted in PCG, following the interpretation for resource categories. We show that the interpretation is compatible with the isomorphism from Chapter 5 in the case of normal resource terms.

Main results [Chapter 8]**:**

- PCG is a closed resource category;
- In particular, we obtain a sound interpretation of resource terms in PCG;
- This interpretation is compatible with the isomorphism for normal terms defined in Chapter 5 (see Figure 9).

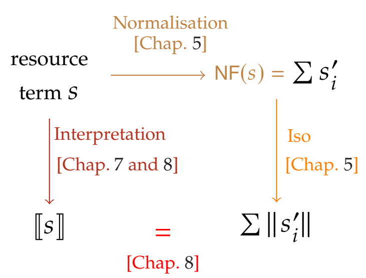


Figure 9: The interpretation behaves nicely!

Outline of the thesis

Preliminaries presents some useful mathematical notions:

Chapter 1 quickly summarizes some definitions on category theory and calculi;
Chapter 2 gives a more detailed presentation of HO games.

An introduction to PCG showcases the first steps of the PCG model:

Chapter 3 defines positions and augmentations, as well as the link with standard HO games;
Chapter 4 presents our result on positional injectivity.

Composition and resource calculus semantics studies the dynamical aspects of PCG as well as the links with resource calculus:

Chapter 5 presents the static bijection between PCG augmentations and normal resource terms;
Chapter 6 defines the composition in PCG. We show that PCG is a smcc, with a strict cartesian closed functor between PCG and HO, compatible with the bijection from Chapter 3;
Chapter 7 introduces resource categories. We define the interpretation of resource terms and prove its soundness. We also study the links with differential categories;
Chapter 8 proves that PCG is a resource category, giving us an interpretation of resource calculus in PCG. It coincides with the bijection of Chapter 5 for normal terms.

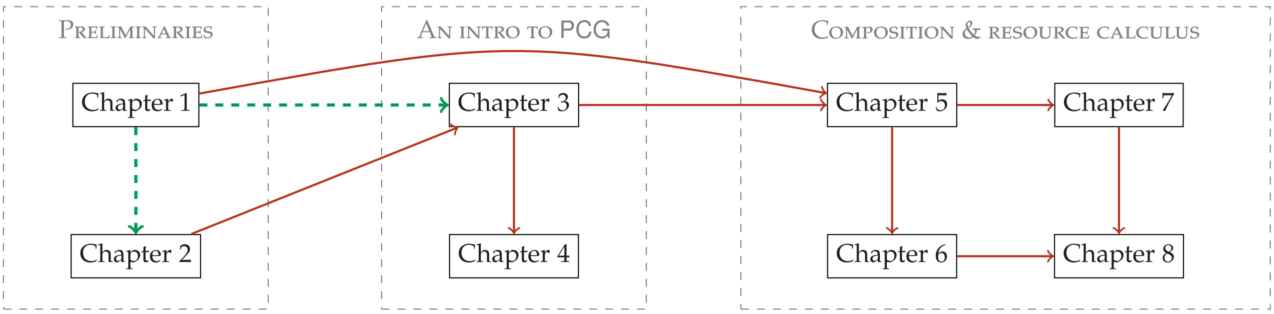


Figure 10: Chapter dependency diagram.
Dashed green arrows ⇢ indicate chapters that might be useful; full red arrows → indicate chapters that are definitely needed.

Preliminaries

In this part, we present several mathematical notions.

In Chapter 1, we give some categorical definitions and we introduce λ*-calculus and resource calculus. This chapter is not intended to be a tutorial and is mostly meant for clarifying notations and definitions.*

In Chapter 2, we present the traditional Hyland Ong games. This chapter intends to be pedagogical and to explain games semantics from the start.

Reminders: Categories, λ-calculus and Resource calculus 1

This chapter summarizes a few notions that are important for this body of work. We expect the reader to be already somewhat familiar with category theory and λ-calculus, as this chapter is not meant to be a complete introduction to these.

First, we remind some useful categorical notions in Section 1.1. Then we state a few definitions and properties of λ-calculus – first the usual λ-calculus in Section 1.2, then the *resource* λ-calculus in Section 1.3.

1.1 Categorical Preliminaries

This section presents some categorical notions which are used throughout this document: symmetric monoidal closed categories, string diagrams, and (co)monoids. We direct the interested reader to [32] for an introduction to category theory.

[32]: Mac Lane (1971), *Categories for the Working Mathematician*

1.1.1 Symmetric Monoidal Closed Categories

As mentionned in the introduction, we will be particularly interested in smcc's. A *monoidal category* is a category equipped with a *tensor*.

Definition 1.1 – Monoidal Category

A **monoidal** category is a category $\mathscr{C}$ equipped with:

- a functor $\otimes : \mathscr{C} \times \mathscr{C} \longrightarrow \mathscr{C}$ called the **tensor**;
- an object $I \in \mathscr{C}$ called the **unit**;
- the following isomorphisms natural in A, B, C:

associator:	$\alpha_{A,B,C} : (A \otimes B) \otimes C \longrightarrow A \otimes (B \otimes C)$
left-unitor:	$\lambda_A : I \otimes A \longrightarrow A$
right-unitor:	$\rho_A : A \otimes I \longrightarrow A$

such that for any objects A, B, C, D, we have:

triangle identity: $(\mathsf{id}_A \otimes \lambda_B) \circ \alpha_{A,I,B} = \rho_A \otimes \mathsf{id}_B$

and the diagram of Figure 1.1 commutes.

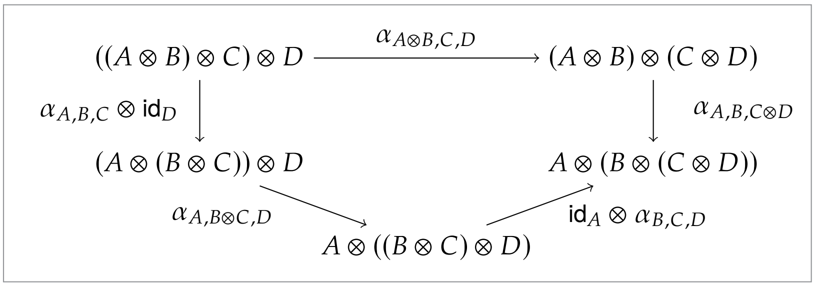


Figure 1.1: Pentagon identity.

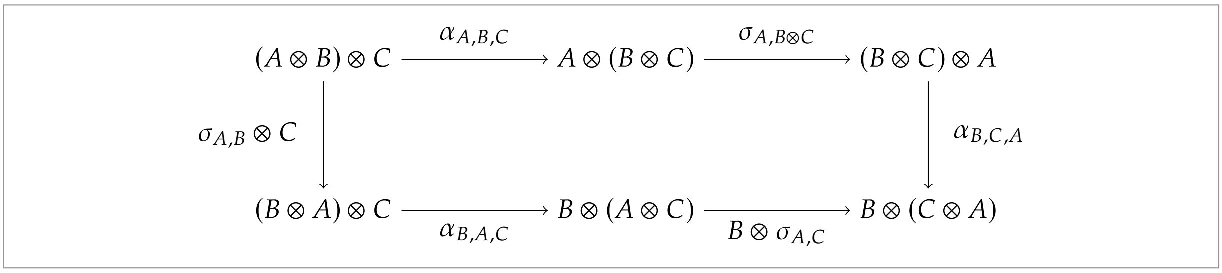


Figure 1.2: Hexagon identity.

If a monoidal category comes with a notion of commutativity of this tensor, it is additionally *symmetric*.

Definition 1.2 – Symmetric Monoidal Category

A **symmetric monoidal category** (or **smc** for short) is a monoidal category $(\mathscr{C}, \otimes, I)$ equipped with an isomorphism natural in A, B:

symmetry: $\sigma_{A,B} : A \otimes B \to B \otimes A$

such that for any objects A, B, C, we have:

symmetry with unit: $\lambda_A \circ \sigma_{A,I} = \rho_A$
symmetry with tensor: $\sigma_{B,A} \circ \sigma_{A,B} = \mathrm{id}_{A\otimes B}$

and the diagram of Figure 1.2 commutes.

In this chapter, all categories are assumed equipped with a symmetric monoidal structure (using $\otimes$ for the tensor and I for the unit), unless stated otherwise.

[31]: Mac Lane (1963), 'Natural Associativity and Commutativity'

[32]: Mac Lane (1971), *Categories for the Working Mathematician*

For the sake of readability, we mostly treat associator and unitors as identities, as justified by Mac Lane's coherence theorem (see [31, Theorem 5.2] for the historical statement and [32, Chapter 7] for the more standard, textbook version).

Finally, we define *symmetric monoidal closed categories*.

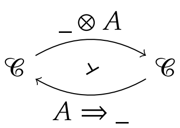


Figure 1.3: SMCC.

Definition 1.3 – Symmetric Monoidal Closed Category

A **symmetric monoidal closed category** (or **smcc**) is an smc $(\mathscr{C}, \otimes, I)$ such that for all A, the functor $\_ \otimes A : \mathscr{C} \to \mathscr{C}$ has a right adjoint (as in Figure 1.3).

[28]: Joyal and Street (1991), 'The geometry of tensor calculus, I'

[39]: Selinger (2010), 'A Survey of Graphical Languages for Monoidal Categories'

1.1.2 String diagrams

We use string diagrams, read from top to bottom, for a graphical representation of some categorical equations (see [28] for a historical introduction and [39] for a survey of graphical languages).

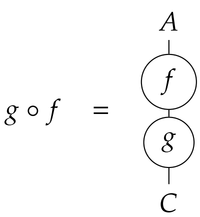


Figure 1.4: Composition.

Composition. Given two morphisms $f : A \to B$ and $g : B \to C$, the composition $g \circ f : A \to C$ is presented in Figure 1.4.

Tensor. The tensor of $f : A \longrightarrow B$ and $g : C \longrightarrow D$ is represented using two wires side by side as in Figure 1.5.

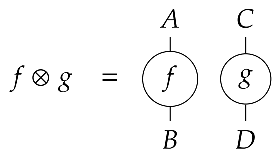


Figure 1.5: Tensor.

Symmetry. The symmetry morphism is represented by crossing the wires as in Figure 1.6.

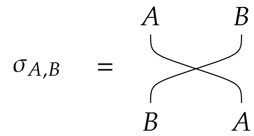


Figure 1.6: Symmetry.

We often omit the labels on wires if they are clear from the context; we also omit I wires because we treat unitors as identities.

Exponential. The last section of Chapter 7 features differential categories, equipped with the endofunctor !. Following [7], for any morphism $f : A \longrightarrow B$ we represent $!f : !A \longrightarrow !B$ with a squared box around f, as in Figure 1.7.

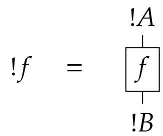


Figure 1.7: $!f$.

[7]: Blute, Cockett, and Seely (2006), 'Differential categories'

1.1.3 Monoids and Comonoids

Finally, the construction of resource categories (Chapter 7) will involve *commutative (co)monoids*.

Definition 1.4 – Monoid

A **monoid** in an smc $\mathscr{C}$ is an object A equipped with:

multiplication:	$\mu_A : A \otimes A \longrightarrow A$
unitor:	$\eta_A : I \longrightarrow A$

satisfying the following equations:

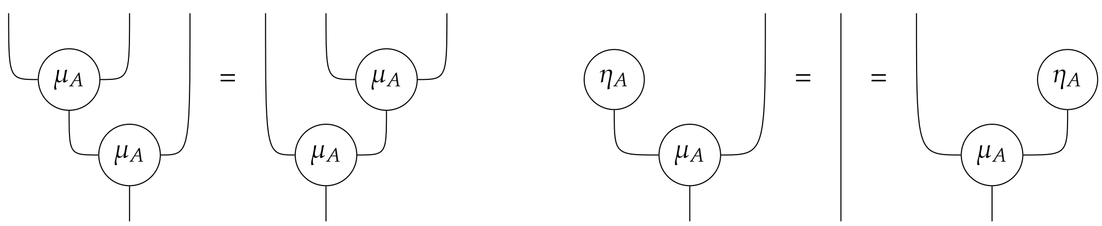
associativity of μ:	$\mu_A \circ (\mu_A \otimes \mathsf{id}_A) = \mu_A \circ (\mathsf{id}_A \otimes \mu_A)$
neutrality of η:	$\mu_A \circ (\eta_A \otimes \mathsf{id}_A) = \mathsf{id}_A = \mu_A \circ (\mathsf{id}_A \otimes \eta_A)$

which are presented in the string diagrams of Figure 1.8.

Definition 1.5 – Commutative monoid

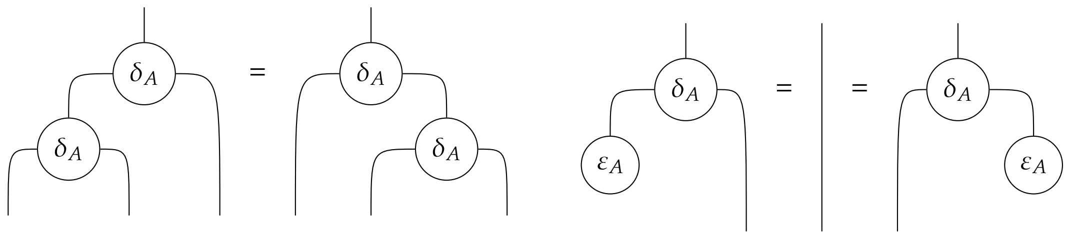
A monoid (A, μ_A, η_A) is **commutative** if it satisfies:

commutativity: $\mu_A \circ \sigma_{A,A} = \mu_A$.



(a) Associativity of μ. **(b)** Neutrality of η.

Figure 1.8: Monoid laws.



(a) Associativity of δ. **(b)** Neutrality of ε.

Figure 1.9: Comonoid laws.

Dually, we define (commutative) comonoids.

Definition 1.6 – Comonoid

A **comonoid** in an smc $\mathscr{C}$ is an object A equipped with:

$$\begin{array}{ll} \textit{co-multiplication:} & \delta_A : A \longrightarrow A \otimes A \\ \textit{co-unitor:} & \varepsilon_A : A \longrightarrow I \end{array}$$

satisfying the equations of Figure 1.9.

Definition 1.7 – Commutative comonoid

A comonoid $(A, \delta_A, \varepsilon_A)$ is **commutative** if it satisfies:

$$\textit{commutativity:} \qquad \sigma_{A,A} \circ \delta_A = \delta_A \,.$$

1.2 Lambda-calculus

[2]: Barendregt (1984), *The lambda calculus - its syntax and semantics*

We briefly state a few definitions and properties of λ-calculus; we direct the interested reader to [2] for a detailed introduction.

1.2.1 Terms of λ-calculus

Terms represent programs – or proofs of propositions, following Curry-Howard isomorphism.

Definition 1.8 – λ-terms

Consider a (countable) set of **variables** $x, y, z, \ldots \in \mathscr{V}$.

We define λ-**terms**, written $M, N, L, \ldots \in \Lambda$, with:

$$\begin{array}{llll} M, N, L, \ldots & ::= & x & \textit{(variable)} \\ & | & \lambda x.M & \textit{(abstraction)} \\ & | & M\ N & \textit{(application)} \end{array}$$

Intuitively, an **abstraction** "$\lambda x.M$" is to be understood as "$x \mapsto M$", *i.e.* "the program which asks for an argument x and then executes the subprogram M". A term of the form $M\ N$ is an **application**, *i.e.* "the subprogram M is called with the subprogram N as its argument".

1.2.2 Free and bound variables

Consider a term M and a variable x occurring in M.

- If x appears in a subprogram of M starting with the abstraction λx, we say x is **bound**.
- Otherwise, if x is unbound, we say that x is a **free variable** in M, noted $x \in FV(M)$.

Example: Consider the term

$$M := \lambda x.(x\ y)\,.$$

It has two variables

$$V(M) = \{x, y\}\,,$$

including one free variable

$$FV(M) = \{y\}\,.$$

The notion of free and bound variables is a key one: intuitively, the name of bound variables should not matter. Consider for instance the functions:

$$f_1 : x \mapsto x \qquad f_2 : y \mapsto y$$

then surely we want our calculus model to express that f_1 and f_2 have the same behavior. These functions translate to the following λ-terms:

$$M_1 = \lambda x.x \qquad M_2 = \lambda y.y$$

so we would like an equivalence relation equating M_1 and M_2.

Definition 1.9 – α-equivalence

The α**-equivalence** is the least congruence relation $=_\alpha$ such that:

$$\lambda x.M =_\alpha \lambda y.M'$$

where $M, M' \in \Lambda$, $x, y \in \mathcal{V}$, and we ask

- $y \notin V(M)$ (we say y is a **fresh** variable),
- M' is the λ-term written like M where each occurrence of x is replaced by y.

Unless specified otherwise, we consider terms up to α-equivalence.

1.2.3 Substitution

The dynamics of λ-calculus relies on the notion of substitution: given a term M using the argument x and another term N, one can **substitute** every occurrence of x in M by a copy of N.

Example: Consider the terms

$$M = \lambda y.x\ y \text{ and } N = z\,.$$

Then the substitution $M[N/x]$ is

$$M[N/x] = \lambda y.z\ y\,.$$

Definition 1.10 – Substitution

Consider two terms $M, N \in \Lambda$ and a variable $x \in \mathcal{V}$.

The **substitution** $M[N/x]$ is the λ-term written like M but where each *free* occurrence of x is replaced by N.

$$\frac{}{(\lambda x.\, M)\, N \to_\beta M[N/x]} \qquad \frac{M \to_\beta L}{\lambda x.M \to_\beta \lambda x.L} \qquad \frac{M \to_\beta L}{M\, N \to_\beta L\, N} \qquad \frac{N \to_\beta L}{M\, N \to_\beta M\, L}$$

Figure 1.10: β-reduction rules.

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\, N : B}$$

Figure 1.11: Typing rules (for simply typed λ-calculus).

1.2.4 Reduction

The other fundamental rule of λ-calculus is the β-reduction: given an abstraction $\lambda x.M$ and a term N, the operation "applying $\lambda x.M$ to N" is **substituting** every occurrence of x in M by a copy of N.

Definition 1.11 – β-reduction

Consider $\lambda x.M, N \in \Lambda$. Then we define the **$\beta$-reduction** with:

$$(\lambda x.M)\, N \to_\beta M[N/x]$$

and the relation is extended with the rules of Figure 1.10.

Example: Consider $(\lambda x.x\ x)\ (\lambda y.y)$. We have the following β-reductions:

$$\begin{array}{ll} & (\lambda x.x\ x)\ (\lambda y.y) \\ \to_\beta & (\lambda y.y)\ (\lambda y.y) \\ \to_\beta & \lambda y.y \end{array}$$

The β-reduction gives us a notion of execution: one reduction represents one step of computation.

Additionnally, the β-reduction is *confluent*, ensuring the *unicity of the normal form* (if it exists).

Proposition 1.12 – Confluence of β-reduction

For any λ-terms M, N_1, N_2, if $M \to_\beta^* N_1$ and $M \to_\beta^* N_2$, then there exists L such that $N_1 \to_\beta^* L$ and $N_2 \to_\beta^* L$.

1.2.5 Simple types

Types in λ-calculus act as programming types: they inform on the nature of the program (or term). In this thesis, we only consider *simple types*.

Definition 1.13 – Simple types

Simple types are given by a **base type** α and the following grammar:

$$A, B, \ldots ::= \alpha \mid A \to B\,.$$

The type $A \to B$ represents "functions from A to B".

Terms are typed following **typing rules** from Figure 1.11, where Γ is a **typing context**, *i.e.* a set of typed variables of the form $x : A$.

1.3 Resource calculus

We now present a few notions regarding *resource calculus*. The definitions mostly follow the ones from [22], but we will be using a sligthly different type system, which will be introduced in Chapter 5.

[22]: Ehrhard and Regnier (2008), 'Uniformity and the Taylor expansion of ordinary lambda-terms'

1.3.1 Preliminaries on tuples and bags

Tuples. If X is a set, we write X^* for the set of finite lists, or tuples, of elements of X, ranged over by $\vec{a}, \vec{b}$, *etc.* We write $\vec{a} = \langle a_1, \dots, a_n \rangle$ to list the elements of $\vec{a}$, of length $|\vec{a}| = n$. The empty list is $\langle\rangle$, and concatenation is simply juxtaposition, *e.g.*, $\vec{a}\vec{b}$.

Multisets. We write $\mathcal{M}_f(X)$ for the set of finite multisets of elements of X, which we call **bags**, ranged over by $\bar{a}, \bar{b}$, *etc.* We write $\bar{a} = [a_1, \dots, a_n]$ for the bag induced by the list $\vec{a} = \langle a_1, \dots, a_n \rangle$ of elements: we then say $\vec{a}$ is an **enumeration** of $\bar{a}$ in this case. We write $[]$ for the empty bag, and use $*$ for bag concatenation. We also write $|\bar{a}|$ for the size of $\bar{a}$: $|\bar{a}|$ is the length of any enumeration of $\bar{a}$.

Partitions. We shall often need to *partition* bags, which requires some care because of the possible duplications. For $\bar{a} \in \mathcal{M}_f(X)$ and $k \in \mathbb{N}$, a k**-partitioning** of $\bar{a}$, written $p : \bar{a} \lhd k$, is a function

$$p : \{1, \dots, |\bar{a}|\} \to \{1, \dots, k\} .$$

Given an enumeration $\langle a_1, \dots, a_n \rangle$ of $\bar{a}$, the associated k**-partition** is the tuple $\langle \bar{a} \upharpoonright_p 1, \dots, \bar{a} \upharpoonright_p k \rangle$, where we set

$$\bar{a} \upharpoonright_p i = [a_j \mid p(j) = i] \text{ for } 1 \leq i \leq k$$

so that $\bar{a} = \bar{a} \upharpoonright_p 1 * \cdots * \bar{a} \upharpoonright_p k$. The obtained k-partition does depend on the chosen enumeration of $\bar{a}$ but, for any function $f : \mathcal{M}_f(X)^k \to \mathbb{M}$ with values in a commutative monoid $\mathbb{M}$ (noted additively), the sum

$$\sum_{\bar{a} \lhd \bar{a}_1 * \cdots * \bar{a}_k} f(\bar{a}_1, \dots, \bar{a}_k) \stackrel{\text{def}}{=} \sum_{p : \bar{a} \lhd k} f(\bar{a} \upharpoonright_p 1, \dots, \bar{a} \upharpoonright_p k)$$

is independent from the enumeration. When indexing a sum with $\bar{a} \lhd \bar{a}_1 * \cdots * \bar{a}_k$ we thus mean to sum over all partitionings $p : \bar{a} \lhd k$, using $\bar{a}_i$ as a shorthand for $\bar{a} \upharpoonright_p i$ in each summand.

Sequences. We will also consider tuples of bags: we write $\mathcal{S}[X]$ for $\mathcal{M}_f(X)^*$. We denote elements of $\mathcal{S}[X]$ as $\vec{a}, \vec{b}$, *etc.* just like for plain tuples, but we reserve the name **sequence** for such tuples of bags.

1.3.2 Terms of the resource calculus

The terms of the resource calculus, as presented in [22], are called **resource terms**. They are just like ordinary λ-terms, except that the argument in an application is a bag of terms instead of just one term.

Definition 1.14 – Resource terms

Consider a (countable) set of variables $x, y, z, \ldots \in \mathcal{V}$. We define **resource terms**, written $s, t, u, \ldots \in \Delta$, and **resource bags**, written $\bar{s}, \bar{t}, \bar{u}, \ldots \in \bar{\Delta}$, with:

$$s, t, u, \ldots ::= x \mid \lambda x.s \mid s\ \bar{t}$$
$$\bar{s}, \bar{t}, \bar{u}, \ldots ::= [s_1, \ldots, s_n]\,.$$

Remark: Resource terms are also considered up to α-equivalence.

1.3.3 Substitution

The dynamics relies on a multilinear variant of substitution, that we will call **resource substitution**: a redex $(\lambda x.s)\ \bar{t}$ reduces to a formal finite sum $s\langle \bar{t}/x\rangle$ of terms, each summand being obtained by substituting each occurrence of x in s with exactly one element of $\bar{t}$.

Notation: We use an extension of syntactic constructs to finite sums of expressions: for $S = \sum_{i\in I} s_i$ and $\bar{T} = \sum_{j\in J} \bar{t}_j$, we set:

$$\lambda x.S \stackrel{\text{def}}{=} \sum_{i\in I} \lambda x.s_i\,,$$
$$[S] * \bar{T} \stackrel{\text{def}}{=} \sum_{i\in I}\sum_{j\in J} [s_i] * \bar{t}_j\,,$$
$$S\ \bar{T} \stackrel{\text{def}}{=} \sum_{i\in I}\sum_{j\in J} s_i\ \bar{t}_j\,.$$

Definition 1.15 – Resource substitution

Resource substitution is defined inductively with:

$$y\langle \bar{t}/x\rangle \stackrel{\text{def}}{=} \begin{cases} t & \text{if } y = x \text{ and } \bar{t} = [t] \\ y & \text{if } y \neq x \text{ and } \bar{t} = [] \\ 0 & \text{otherwise} \end{cases}$$
$$(\lambda z.s)\langle \bar{t}/x\rangle \stackrel{\text{def}}{=} \lambda z.(s\langle \bar{t}/x\rangle)$$
$$(s\ \bar{u})\langle \bar{t}/x\rangle \stackrel{\text{def}}{=} \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} (s\langle \bar{t}_1/x\rangle)\,(\bar{u}\langle \bar{t}_2/x\rangle)$$
$$[s_1, \ldots, s_n]\langle \bar{t}/x\rangle \stackrel{\text{def}}{=} \sum_{\bar{t} \triangleleft \bar{t}_1 * \cdots * \bar{t}_n} [s_1\langle \bar{t}_1/x\rangle, \ldots, s_n\langle \bar{t}_n/x\rangle]$$

where z is chosen fresh in the abstraction case.

The actual protagonists of the calculus are thus sums of terms rather than single terms. We will generally write $\Sigma[X]$ for the set of finite formal sums on set X – those may be considered as finite multisets, but we adopt a distinct additive notation to avoid confusion with bags.

Resource substitution is in turn extended by linearity, setting

$$S\langle \bar{T}/x\rangle \stackrel{\text{def}}{=} \sum_{i\in I}\sum_{j\in J} s_i\langle \bar{t}_j/x\rangle$$

with the same notations as above.

1.3.4 Resource reduction

The **reduction of resource terms** $\rightsquigarrow\ \subseteq \Delta \times \Sigma[\Delta]$ is defined inductively by the rules of Figure 1.12 – simultaneously with the reduction of resource bags $\rightsquigarrow\ \subseteq \bar{\Delta} \times \Sigma[\bar{\Delta}]$. It is extended to $\rightsquigarrow\ \subseteq \Sigma[\Delta] \times \Sigma[\Delta]$ by setting $S \rightsquigarrow S'$ whenever $S = t + U$ and $S' = T' + U$ with $t \rightsquigarrow T'$.

Unlike the reduction in the usual λ-calculus, the reduction $\rightsquigarrow$ is *strongly normalizing*, *i.e.* there is no infinite sequence of redutions.

$$\frac{}{(\lambda x.s)\,\bar{t} \rightsquigarrow s\langle \bar{t}/x\rangle} \qquad \frac{s \rightsquigarrow S'}{\lambda x.s \rightsquigarrow \lambda x.S'} \qquad \frac{s \rightsquigarrow S'}{s\,\bar{t} \rightsquigarrow S'\,\bar{t}} \qquad \frac{s \rightsquigarrow S'}{[s] * \bar{t} \rightsquigarrow [S'] * \bar{t}} \qquad \frac{\bar{t} \rightsquigarrow \bar{T}'}{s\,\bar{t} \rightsquigarrow s\,\bar{T}'}$$

Figure 1.12: Rules of resource reduction.

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, x : A \vdash s : B}{\Gamma \vdash \lambda x.s : A \to B} \qquad \frac{\Gamma \vdash s : A \to B \quad \Gamma \vdash \bar{t} : A}{\Gamma \vdash s\,\bar{t} : B} \qquad \frac{\Gamma \vdash t_1 : A \quad \dots \quad \Gamma \vdash t_n : A}{\Gamma \vdash [t_1, \dots, t_n] : A}$$

Figure 1.13: Typing rules for resource calculus.

Theorem 1.16 – (see [22, Theorem 9])

The reduction $\rightsquigarrow$ on $\Sigma[\Delta]$ is confluent and strongly normalizing.

[22]: Ehrhard and Regnier (2008), 'Uniformity and the Taylor expansion of ordinary lambda-terms'

1.3.5 Typing rules

We use simple types as in Definition 1.13.

The usual typing rules are given in Figure 1.13.

Notation (priority rules): We write $A \to B \to C$ for $A \to (B \to C)$.

When constructing a bijection between resource calculus and games, we consider terms that are in normal form, and that are η-*long*.

Definition 1.17 – η-expansion

Consider a normal resource term s of type $A_1 \to \dots \to A_n \to \alpha$.

We say s is **η-long** if it has the shape:

$$\lambda x_1 \dots . \lambda x_n . t$$

with t a (normal) term of type α, and each subterm of t is η-long, recursively.

In Chapter 5, we consider a modified version of the typing rules, constructed to ensure that normal terms are always η-long. Nonetheless, we give the rules of Figure 1.13 in order to describe the bijection between resource calculus and HO games from [40].

[40]: Tsukada and Ong (2016), 'Plays as Resource Terms via Non-idempotent Intersection Types'

Introduction to Hyland-Ong Games

2

In this chapter, we introduce HO games, which will be our starting point for the question of positional injectivity in the next chapter.

If the reader is not familiar with game semantics, it might help to keep in mind the following correspondences between games and programs:

Arenas *(the rules of the game)*	$\longleftrightarrow$	**Types**
Plays *(one iteration of the game)*	$\longleftrightarrow$	**Executions**
Strategies *(a set of several plays)*	$\longleftrightarrow$	**Programs**

2.1 Arenas

Our first objects of interest in game semantics are **arenas**, representing *types* – they set "the rules" of the game, *i.e.* they list all computational events available to Player and Opponent given the type of the program being computed.

2.1.1 Definition

An arena is a set of **moves** (the possible events) which are **polarized** (indicating if the move is playable by Opponent or by Player) and partially ordered (following causal dependencies of interactions). More formally:

Definition 2.1 – Arena

An **arena** is $A = \langle |A|, \leq_A, \mathrm{pol}_A \rangle$ with:

- $|A|$ is a countable set of **moves**,
- $\leq_A$ is a partial order over $|A|$,
- $\mathrm{pol}_A : |A| \longrightarrow \{-, +\}$ is a **polarity function**.

Moreover, these data must satisfy the following conditions:

finitary: for all $a \in |A|$, $[a]_A = \{a' \in |A| \mid a' \leq_A a\}$ is finite,
forestial: for all $a_1, a_2 \leq_A a$, either $a_1 \leq_A a_2$ or $a_2 \leq_A a_1$,
alternating: for all $a_1 \rightarrowtriangle_A a_2$, $\mathrm{pol}_A(a_1) \neq \mathrm{pol}_A(a_2)$.

where $a_1 \rightarrowtriangle_A a_2$ means $a_1 <_A a_2$ with no move strictly in between.

Notation: We define the **immediate causality** relation $\rightarrowtriangle_A$ with:

For all $a, b \in |A|$, $a \rightarrowtriangle_A b$ iff:
- $a <_A b$,
- for any $c \in |A|$, if $a \leq_A c \leq_A b$, then $a = c$ or $b = c$.

Though our notations differ superficially, our arenas are similar to those presented in [27]. As in concurrent games, we use $+$ and $-$ for polarities instead of O and P: positive moves are due to Player / the program, and negative moves to Opponent / the environment.

[27]: Hyland and Ong (2000), 'On Full Abstraction for PCF: I, II, and III'

For any arena A and move $\mathsf{a} \in |\mathsf{A}|$, we write a^- (respectively a^+) as a shortcut for "a s.t. $\mathsf{pol}_\mathsf{A}(\mathsf{a}) = -$" (respectively $\mathsf{pol}_\mathsf{A}(\mathsf{a}) = +$).

Thanks to finitarity, $\leq_\mathsf{A}$ can be recovered from $\rightarrowtriangle_\mathsf{A}$, so we draw only the immediate causality in graphical representations of arenas. Consider as an example the arena A from Figure 2.1. We read this diagram in the following way:

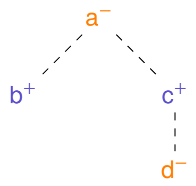


Figure 2.1: An arena A.

- $|\mathsf{A}| = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$,
- $\rightarrowtriangle_\mathsf{A}$ is represented by dashed lines, read from top to bottom,
- for all $\mathsf{e} \in |\mathsf{A}|$, $\mathsf{pol}_\mathsf{A}(\mathsf{e})$ is indicated by the superscript of e.

In most diagrams, we also use the convention orange for Opponent / negative moves, and blue for Player / positive moves.

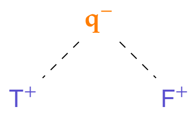


Figure 2.2: Arena bool.

We show in Figure 2.2 the representation of the data type bool as an arena: Opponent initiates the execution with $\mathbf{q}^-$, the initial *query* requesting the value of a boolean. Player may respond with T^+ (true) or F^+ (false).

Another example is presented in Figure 2.3: given a program of type nat, its possible interactions with its environment are:

- being called by the environment ($\mathbf{q}^-$),
- reacting with its value (any n^+ with $n \in \mathbb{N}$).

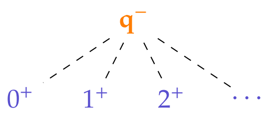


Figure 2.3: Arena nat.

Notation: We write I for the empty arena and o for the arena with exactly one negative move $\mathbf{q}^-$.

We define additional conditions on arenas:

Definition 2.2 – Well-opened arena

An arena A is **well-opened** if it has exactly one minimal move, *i.e.*:

$$\min(\mathsf{A}) = \{\mathsf{a} \in |\mathsf{A}| \mid \mathsf{a} \text{ is minimal for } \leq_\mathsf{A}\}$$

is a singleton.

If A is well-opened, its only minimal move is the **initial move**, written $\mathsf{init}(\mathsf{A})$.

Definition 2.3 – Negative (and positive) arenas

An arena A is **negative** if $\mathsf{pol}_\mathsf{A}(\min(\mathsf{A})) = \{-\}$.

Likewise, an arena A is **positive** if $\mathsf{pol}_\mathsf{A}(\min(\mathsf{A})) = \{+\}$.

Figure 2.4: A non negative (nor positive) and non well-opened arena.

All arenas presented so far were negative and well-opened. This is not a requirement for arenas: Figure 2.4 for example shows an arena that is neither negative nor well-opened.

However, arenas in HO games are usually negative, hence **in the rest of this chapter, all arenas are assumed to be negative, unless stated otherwise**.

Since this not the case for PCG arenas, we did not ask for negativity in the definition of arenas, to allows us to use the same arena definition for both game models.

2.1.2 Constructors on arenas

More elaborate types involve matching constructions: the **product** and the **arrow**.

The product of two arenas simply places both arenas side by side.

Notation: For any two sets E_1 and E_2, their **disjoint union** is:

$$E_1 + E_2 = \{(i, e) \mid i = 1, 2 \text{ and } e \in E_i\} .$$

Definition 2.4 – Product of arenas

Consider A_1 and A_2 two arenas. Their **product** $\mathsf{A}_1 \otimes \mathsf{A}_2$ is the arena defined with:

$$\begin{aligned} |\mathsf{A}_1 \otimes \mathsf{A}_2| &= |\mathsf{A}_1| + |\mathsf{A}_2| , \\ (i, \mathsf{a}) \leq_{\mathsf{A}_1 \otimes \mathsf{A}_2} (i, \mathsf{b}) &\Leftrightarrow \mathsf{a} \leq_{\mathsf{A}_i} \mathsf{b} , \\ \mathsf{pol}_{\mathsf{A}_1 \otimes \mathsf{A}_2}((i, \mathsf{a})) &= \mathsf{pol}_{\mathsf{A}_i}(\mathsf{a}) . \end{aligned}$$

It is immediate that $\mathsf{A}_1 \otimes \mathsf{A}_2$ also is an arena.

Remark: In the HO games category, the product $\mathsf{A} \otimes \mathsf{B}$ *is* a cartesian product – hence its name. However, we use the notation $\mathsf{A} \otimes \mathsf{B}$, instead of the usual notation $\mathsf{A} \times \mathsf{B}$, because this construction will be shown to be a tensor in the (symmetric monoidal) category of PCG.

For any family $(\mathsf{A}_i)_{i \in I}$ of arenas, this extends to $\prod_{i \in I} \mathsf{A}_i$ in the obvious way. Any arena A decomposes (up to forest isomorphism) as $\mathsf{A} \cong \prod_{i \in I} \mathsf{A}_i$ for some family $(\mathsf{A}_i)_{i \in I}$ of well-opened arenas.

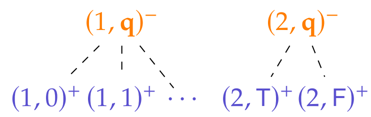


Figure 2.5: Arena $\mathsf{nat} \otimes \mathsf{bool}$.

We now define the **arrow** constructor: given two arenas A and B with B well-opened, the arrow arena $\mathsf{A} \Rightarrow \mathsf{B}$ is similar to the product, but we invert polarities of moves from A and add a causal dependency from the moves of A to the initial move of B.

Definition 2.5 – Arrow

Consider $\mathsf{A}_1, \mathsf{A}_2$ two arenas with A_2 well-opened.

We define $\mathsf{A}_1 \Rightarrow \mathsf{A}_2$ with:

$$\begin{aligned} |\mathsf{A}_1 \Rightarrow \mathsf{A}_2| &= |\mathsf{A}_1| + |\mathsf{A}_2| , \\ (i, \mathsf{a}) \leq_{\mathsf{A}_1 \Rightarrow \mathsf{A}_2} (j, \mathsf{b}) &\Leftrightarrow i = j \text{ and } \mathsf{a} \leq_{\mathsf{A}_i} \mathsf{b} , \\ & \quad\;\; \text{or } (j, \mathsf{b}) = (2, \mathsf{init}(\mathsf{A}_2)) , \\ \mathsf{pol}_{\mathsf{A}_1 \Rightarrow \mathsf{A}_2}((i, \mathsf{a})) &= -\mathsf{pol}_{\mathsf{A}_1}(\mathsf{a}) \text{ if } i = 1 , \\ & \quad\;\; \mathsf{pol}_{\mathsf{A}_2}(\mathsf{a}) \text{ if } i = 2 . \end{aligned}$$

Again, it is clear that $\mathsf{A}_1 \Rightarrow \mathsf{A}_2$ is a (well-opened) arena.

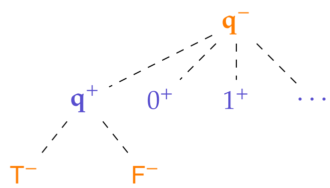


Figure 2.6: Arena $\mathsf{bool} \Rightarrow \mathsf{nat}$.

Figure 2.6 displays the arena $\mathsf{A} = \mathsf{bool} \Rightarrow \mathsf{nat}$. Once Opponent initiates the computation with $\mathbf{q}^-$, two types of Player moves become available: Player may react by giving directly an integer (n^+), or they can choose to evaluate their argument ($\mathbf{q}^+$), which in turn allows Opponent to react with a boolean (T^- or F^-). Remark that we represent moves as the moves from their arena component, without the tags coming from the disjoint union. The moves of A are actually:

$$|\mathsf{A}| = \{(1, \mathbf{q}), (1, \mathsf{T}), (1, \mathsf{F}), (2, \mathbf{q}), (2, 0), (2, 1), (2, 2), \ldots\},$$

but we often omit tags in graphical representations for readability.

$$(\alpha \longrightarrow \alpha) \longrightarrow \alpha \longrightarrow \alpha$$

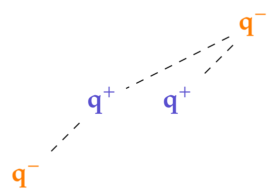


Figure 2.7: Arena $(\mathsf{o} \Rightarrow \mathsf{o}) \Rightarrow \mathsf{o} \Rightarrow \mathsf{o}$.

For any arenas A, B and C, we read $\mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C}$ as $\mathsf{A} \Rightarrow (\mathsf{B} \Rightarrow \mathsf{C})$. We can now interpret any simple type, using o for the base type α and the arrow constructor for higher-order types. For instance, Figure 2.7 displays the arena $(\mathsf{o} \Rightarrow \mathsf{o}) \Rightarrow \mathsf{o} \Rightarrow \mathsf{o}$, matching the simple type $(\alpha \to \alpha) \to \alpha \to \alpha$

with atomic type α – where each move is placed under the atom of the type it comes from.

Let us go back to our introductory example (Figure 5). The program corresponding to the addition of two integers lives in the following arena:

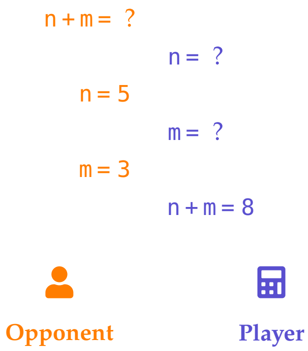


Figure 5: A *play* for "5+3 = 8"

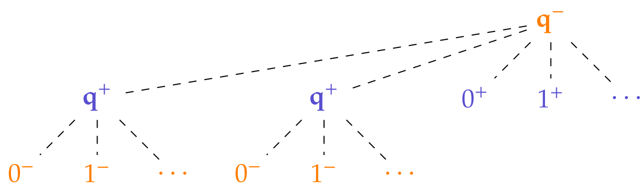


Figure 2.8: Arena nat ⇒ nat ⇒ nat.

Remark that this arena $\mathsf{A} = \mathsf{nat} \Rightarrow \mathsf{nat} \Rightarrow \mathsf{nat}$ is isomorphic to the arena $\mathsf{A}' = (\mathsf{nat} \otimes \mathsf{nat}) \Rightarrow \mathsf{nat}$; the only difference being the tags of moves, which are not represented in Figure 2.8. For instance, $(2, (2, \mathbf{q}))$ the initial move of A would become $(2, \mathbf{q})$ in A'.

2.2 Plays

Next we define **plays**, corresponding to program executions: a play is a particular iteration of the game, following the rules given by the arena.

2.2.1 Definition

In Hyland-Ong games, players are allowed to *backtrack*, and resume the play from any earlier stage. This is made formal by the notion of **pointing strings**, which are sequences of moves with optional pointers from moves to earlier moves (representing causal dependencies).

Definition 2.6 – Pointing String

A **pointing string** over a set of moves Σ is a string $s \in \Sigma^*$, where each move may additionally come equipped with a **pointer** to an earlier move.

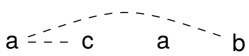


Figure 2.9: A pointing string.

Figure 2.9 shows a pointing string over $\{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$, read from left to right. Pointers are indicated by dashed lines:

- the first move a has no pointer,
- the second move c points to the first move a,
- the third move a has no pointer,
- the fourth move b points to the first move a.

For any pointing string s, we often write $s = s_1 \ldots s_n$ where s_i is the i-th move of s, and pointers are left implicit. The **length** of s, denoted by $|s|$, is the number n of moves in s. We write ε for the pointing string of length 0. For any s of length n, for any $k \leq n$, we define $s[1 : k] = s_1 \ldots s_k$ (where we keep the pointers). We say $s[1 : k]$ is a **prefix** of s, written $s[1 : k] \sqsubseteq s$.

For instance, the prefixes of the pointing string s from Figure 2.9 are:

$$s[1:0] = \varepsilon,$$
$$s[1:1] = \mathsf{a},$$
$$s[1:2] = \mathsf{a} \text{ - - - } \mathsf{c},$$
$$s[1:3] = \mathsf{a} \text{ - - - } \mathsf{c} \quad \mathsf{a},$$
$$s[1:4] = \mathsf{a} \text{ - - - } \mathsf{c} \quad \mathsf{a} \quad \mathsf{b}.$$

Executions of programs will be represented by pointing strings over the arena moves, with additional conditions.

Definition 2.7 – Play

Consider an arena A. A **play** on A is a pointing string $s = s_1 \dots s_n$ over A with the following properties:

rigid:	if s_i points to s_j, then $s_j \multimap_{\mathsf{A}} s_i$,
alternating:	for any $1 \leq i < n$, $\mathsf{pol}_{\mathsf{A}}(s_i) \neq \mathsf{pol}_{\mathsf{A}}(s_{i+1})$,
negative:	if $n \geq 1$, then $\mathsf{pol}_{\mathsf{A}}(s_1) = -$,
legal:	for all $1 \leq i \leq n$, $s_i \in \min(\mathsf{A})$ or s_i has a pointer.

We write Plays(A) the set of plays on A.

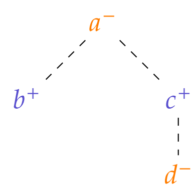


Figure 2.1: An arena A.

Recall the arena A from Figure 2.1. Then the pointing string presented in Figure 2.9 is actually a play on A.

Given a play s and $k \leq |s|$, we say that $s[1:k]$ is a **positive prefix** (respectively a **negative prefix**), noted

$$s[1:k] \sqsubseteq^+ s \qquad \text{(respectively } s[1:k] \sqsubseteq^- s\text{)},$$

if $\mathsf{pol}_{\mathsf{A}}(s_k) = +$ (respectively $\mathsf{pol}_{\mathsf{A}}(s_k) = -$). We extend this definition to the empty play by stating that ε is a positive prefix of any play s. More generally, a play s on an arena A is **positive** if it is empty or if its last move is positive, *i.e.* $\mathsf{pol}_{\mathsf{A}}(s_{|s|}) = +$. We write $\mathsf{Plays}^+(\mathsf{A})$ for the set of positive plays on A.

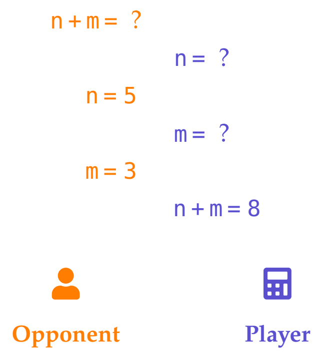


Figure 5: A "play" for "5+3 = 8"

Definition 2.8 – Well-opened play

A play $s \in \mathsf{Plays}(\mathsf{A})$ is **well-opened** if and only if it has exactly one move minimal in A.

We write $\mathsf{Plays}_\bullet(\mathsf{A})$ for the set of well-opened plays on A.

Combining the two notations, we write $\mathsf{Plays}^+_\bullet(\mathsf{A})$ for the set of positive, well-opened plays on A.

Going back to Figure 5 again, the execution "5 + 3 = 8" corresponds to the (positive, well-opened) play $s \in \mathsf{Plays}^+_\bullet(\mathsf{nat} \Rightarrow \mathsf{nat} \Rightarrow \mathsf{nat})$ presented in Figure 2.10, where we use indices for q_1 and q_2 to distinguish them[1].

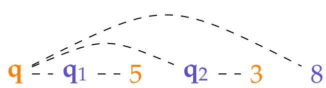


Figure 2.10: "5+3 = 8" as an actual play

1: Recall that these moves are actually $(1, \mathsf{q})$ and $(2, (1, \mathsf{q}))$ – we simply write q_1 and q_2, and drop the other tags, to avoid an overdecorated diagram.

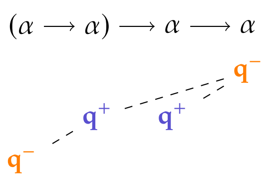


Figure 2.7: Arena $(o \Rightarrow o) \Rightarrow o \Rightarrow o$.

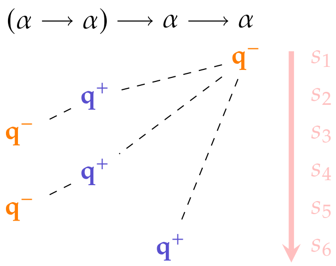


Figure 2.11: A play s in $[\![M]\!]_{\mathsf{HO}}$.

We are now able to write plays corresponding to the evaluation of simply-typed lambda-terms. Consider for example the term:

$$M = \lambda f^{\alpha \to \alpha}.\lambda x^{\alpha}.f\,(f\,x) \qquad \text{of type } A = (\alpha \to \alpha) \to \alpha \to \alpha.$$

We interpret A as an arena in Figure 2.7. Then Figure 2.11 shows a play $s \in [\![M]\!]_{\mathsf{HO}}$, where $[\![M]\!]_{\mathsf{HO}}$ is the interpretation (which is yet to be defined) of M as a strategy $\sigma \subseteq \mathsf{Plays}^+((o \Rightarrow o) \Rightarrow o \Rightarrow o)$. We shall develop this interpretation in the following section; for now we focus on understanding s. The diagram presented in Figure 2.11 is to be read in the following way: moves are sequentially ordered from top to bottom, and as for arenas each move is placed under its corresponding type component, with dashed lines for the justification pointers (matching immediate causality). To help the unfamiliar reader, we wrote the corresponding s_i's, horizontally aligned, to the right of the diagram. Figure 2.11 corresponds to the following execution:

1. First, Opponent asks for something of type α – knowing M is of type $(\alpha \to \alpha) \to \alpha \to \alpha$. This corresponds to $s_1 = \mathbf{q}^-$.
2. The variable in head position at this step of the computation is f of type $\alpha \to \alpha$, so Player reacts with $s_2 = \mathbf{q}^+$ (the initial move of the subpart of the arena coming from $\alpha \to \alpha$).
3. Opponent then wants to evaluate the argument of f (expecting a subterm of type α).
4. Player reacts with another copy of the move corresponding to the variable f (of type $\alpha \to \alpha$).
5. Opponent asks for the argument of this second f.
6. Player responds with the move corresponding to x of type α.

Intuitively, negative moves correspond to λ-abstractions / evaluating the argument of an application, and positive moves correspond to variable occurrences. The pointers indicate both the link between a variable and the λ-abstraction it came from, and the link between an evaluation of an argument and the function waiting for this argument.

2.2.2 Views

We saw several examples of plays in the previous subsection. Some, like the play (from Figure 2.9):

$$s \quad = \quad \mathsf{a}^- \quad \mathsf{c}^+ \quad \mathsf{a}^- \quad \mathsf{b}^+$$

feature repetition and duplication of moves: s_1 and s_3 both correspond to the move a^-. In HO games, both players are allowed to evaluate again part of the program they already evaluated; here for instance Opponent starts the evaluation with a^-, Player reacts with c^+, and then Opponent decides to start the evaluation again by playing a^- a second time.

This means that Opponent is allowed to "open several threads" corresponding to "several program phrases". However, when interpreting terms of the simply-typed λ-calculus, we do not want Player to be able to react differently to moves that are duplications of the same Opponent move: simply-typed terms have no mutable references and thus no way of storing the information "this is the n-th time I'm being evaluated". This corresponds to the key notion of *innocence*: an innocent strategy only uses the information from the "current program phrase" to decide their next move. This "current program phrase" is captured by the **P-view**.

Definition 2.9 – P-view

For any arena A, we set a partial function $\ulcorner - \urcorner : \mathsf{Plays}(\mathsf{A}) \rightharpoonup \mathsf{Plays}(\mathsf{A})$ with $\ulcorner s \urcorner$ defined inductively on s by:

$$\begin{array}{lcll} \ulcorner s\ \mathsf{a} \urcorner & = & \mathsf{a} & \text{if } \mathsf{a} \in \min(\mathsf{A}), \\ \ulcorner s\ \mathsf{a}^-\ \mathsf{b}^+ \urcorner & = & \ulcorner s\ \mathsf{a}^- \urcorner\ \mathsf{b}^+ & \text{if the pointer of } \mathsf{b} \text{ is in } \ulcorner s\ \mathsf{a}^- \urcorner, \\ \ulcorner s\ \mathsf{a}^+\ s'\ \mathsf{b}^- \urcorner & = & \ulcorner s\ \mathsf{a}^+ \urcorner\ \mathsf{b}^- & \text{if } \mathsf{b} \text{ points to } \mathsf{a}, \end{array}$$

undefined otherwise. In the last two cases, b keeps its pointer in the resulting play.

If defined, $\ulcorner s \urcorner$ is called the **P-view** of s.

By induction, the P-view of a play s is always a play itself if it exists: all moves remaining in $\ulcorner s \urcorner$ are either minimal (by the first case of the definition) or justified (by the last two cases of the definition). Whenever constructing the P-view would involve "jumping over a pointer" and forgetting it, $\ulcorner s \urcorner$ is undefined. For instance:

$$\begin{array}{lclcl} \ulcorner \mathsf{a}^- \urcorner & = & \mathsf{a}^-, & & \\ \ulcorner \mathsf{a}^- \text{ -- } \mathsf{c}^+ \urcorner & = & \ulcorner \mathsf{a}^- \urcorner\ \ \mathsf{c}^+ & = & \mathsf{a}^- \text{ -- } \mathsf{c}^+, \\ \ulcorner \mathsf{a}^- \text{ -- } \mathsf{c}^+\quad \mathsf{a}^- \urcorner & = & \mathsf{a}^-, & & \\ \ulcorner \mathsf{a}^- \text{ -- } \mathsf{c}^+\quad \mathsf{a}^-\quad \mathsf{b}^+ \urcorner & & \text{undefined}, & & \end{array}$$

where the last P-view is undefined since $\ulcorner\ \mathsf{a}^- - \mathsf{c}^+\ \mathsf{a}^-\ \urcorner$ only keeps $s_3 = \mathsf{a}^-$ and the pointer of $s_4 = \mathsf{b}^+$ is $s_1 = \mathsf{a}^-$. We say that such a play is non-**P-visible**.

Definition 2.10 – P-Visibility

Consider an arena A. A play $s \in \mathsf{Plays}(\mathsf{A})$ is **P-visible** if and only if for all prefixes $t \sqsubseteq s$, the P-view $\ulcorner\ t\ \urcorner$ is defined.

Constructing a P-view (if it exists) is idempotent: for any $s \in \mathsf{Plays}(\mathsf{A})$,

$$\ulcorner s \urcorner = \ulcorner \ulcorner s \urcorner \urcorner .$$

Remark: By *legality* of plays, we have:

- if $s \neq \varepsilon$, s_1 is initial;
- if $1 \leq i \leq |s|$ is odd, s_i is negative.

For any $s \in \mathsf{Plays}(\mathsf{A})$, we say s **is a P-view** if $\ulcorner s \urcorner = s$ – remark that by construction, a P-view is always P-visible. The P-views of A are exactly the plays $s \in \mathsf{Plays}(\mathsf{A})$ such that:

- for any odd $1 < i \leq |s|$, s_i points to s_{i-1}.

2.3 Strategies

If plays are particular executions of a program, then a program as a whole is represented by a **strategy**, a set of plays corresponding to every possible execution of that program.

2.3.1 Definition

This set of plays must follow some conditions: in a deterministic setting, Player should always react in the same way to a given play, for instance.

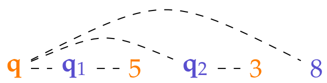
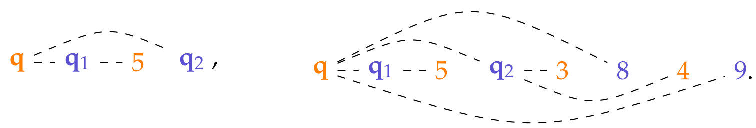


Figure 2.10: "5 + 3 = 8" – as a play

Consider the "addition" program: Figure 2.10 shows a possible play in the strategy corresponding to this program. But Opponent might want to compute other sums than just "5 + 3" – our strategy should also include, amongst (many) others, the following plays:

What about duplications? Opponent is also allowed to duplicate some moves, or to stop the computation at any point. Thus our strategy will also include plays such as:



In the general case, all we ask of strategies is that they be *deterministic* and *closed under taking prefixes*.

Definition 2.11 – Strategy

A **strategy** $\sigma : \mathsf{A}$ on an arena A is a set $\sigma \subseteq \mathsf{Plays}^+(\mathsf{A})$, satisfying:

$$\begin{aligned} \text{non-empty:}\quad & \varepsilon \in \sigma\,, \\ \text{prefix-closed:}\quad & \forall s \in \sigma,\ \text{if } t \sqsubseteq^+ s,\ \text{then } t \in \sigma\,, \\ \text{deterministic:}\quad & \forall s \in \sigma,\ \text{if } s\mathsf{ab}, s\mathsf{ab}' \in \sigma,\ \text{then } s\mathsf{ab} = s\mathsf{ab}'\,. \end{aligned}$$

Remark: Implicit in the last clause is that $s\mathsf{ab}$ and $s\mathsf{ab}'$ also have the same pointers.

We say that a strategy $\sigma : \mathsf{A}$ is **P-visible** if all plays $s \in \sigma$ are P-visible.

2.3.2 Innocence

Innocence captures the fact that some strategies only react to the "current program phrase" and not the whole history of moves. In other words, the behavior of Player entirely depends on the current P-view.

Definition 2.12 – Innocence

A strategy $\sigma : \mathsf{A}$ is **innocent** if it is P-visible and satisfies:

for all $s\mathsf{ab}$, $t \in \sigma$, if $t\mathsf{a} \in \mathsf{Plays}(\mathsf{A})$ and $\ulcorner s\mathsf{a} \urcorner = \ulcorner t\mathsf{a} \urcorner$, then $t\mathsf{ab} \in \sigma$,

where $\ulcorner s\mathsf{ab} \urcorner = \ulcorner t\mathsf{ab} \urcorner$ (informally, b points "as in $s$ab" in $t$ab).

An innocent strategy $\sigma : \mathsf{A}$ is determined by its **P-view forest**:

$$\ulcorner\!\ulcorner \sigma \urcorner\!\urcorner = \{\ulcorner s \urcorner \mid s \in \sigma\} .$$

Hence, we actually have two characterizations of innocent strategies: the "fat" innocent strategy is the set of plays of the strategy σ, and the "meagre" innocent strategy is just the set of P-views $\ulcorner\!\ulcorner \sigma \urcorner\!\urcorner$.

Since P-views are well-opened, we might also characterize any innocent strategy $\sigma : \mathsf{A}$ by the subset of its well-opened plays:

$$\sigma_\bullet = \sigma \cap \mathsf{Plays}_\bullet(\mathsf{A}) .$$

2.3.3 Other properties of strategies: totality and finiteness

We might want our strategy to react to every possible action of Opponent – that would be a **total** strategy; or we might allow it to diverge sometimes – giving us a **partial** strategy.

Definition 2.13 – Total strategy

Consider a strategy $\sigma : \mathsf{A}$. We say σ is **total** if for all $s \in \sigma$, for all $\mathsf{a}^- \in \mathsf{A}$ such that $s\mathsf{a} \in \mathsf{Plays}(\mathsf{A})$, there exists $\mathsf{b}^+ \in \mathsf{A}$ such that $s\mathsf{ab} \in \sigma$. Otherwise, we say that σ is **partial**.

Finally, innocent strategies are "infinite" in the sense that given an innocent strategy σ and a play $s \in \sigma$, any play of the form $s^n = s \dots s$ with n copies of s also belongs in σ by innocence – so σ admits an infinite number of plays. However, since innocent strategies are characterized by their P-views, we can distinguish between strategies having a *finite* set of P-views, and strategies having an infinite set of P-views.

Reminder: The **P-views forest** of an innocent strategy σ is:

$$\ulcorner\!\ulcorner \sigma \urcorner\!\urcorner = \{\ulcorner s \urcorner \mid s \in \sigma\} .$$

Definition 2.14 – Finite innocent strategy

Consider an innocent strategy $\sigma : \mathsf{A}$. We say σ is **finite** if its P-view forest $\ulcorner\!\ulcorner \sigma \urcorner\!\urcorner$ is finite. Otherwise, we say that σ is **infinite**.

Total finite innocent strategies are already well-known in the litterature. For example, on arenas interpreting simple types with a single atomic type α, total finite innocent strategies exactly correspond to β-normal η-long simply-typed λ-terms [17, Theorem 5].

[17]: Danos, Herbelin, and Regnier (1996), 'Game semantics and abstract machines'

Remark: This result only holds for arenas with *one* atomic type.

2.4 Composition

As a denotational model, HO games include a notion of *composition*: how do two programs interact with each other?

Given strategies $\sigma : \mathsf{A} \Rightarrow \mathsf{B}$ and $\tau : \mathsf{B} \Rightarrow \mathsf{C}$, we wish to somehow define a strategy $\tau \odot^{\mathsf{HO}} \sigma : \mathsf{A} \Rightarrow \mathsf{C}$. Hence, we must construct a set of plays corresponding to possible executions of the program "$\tau \odot^{\mathsf{HO}} \sigma$".

Intuitively, the composition works in two steps:

- First, we choose two plays $s \in \sigma$ and $t \in \tau$ such that s and t "agree on the moves played in B", following an *interaction*.
- Then, the interaction of these plays induces a play on $\mathsf{A} \Rightarrow \mathsf{C}$, keeping only the moves from the outer arena components and adding pointers and sequential order "following those of s and t". We say we *hide* moves occurring in the shared arena component B.

The strategy $\tau \odot^{\mathsf{HO}} \sigma$ is the set of all the possible compositions of plays.

More formally, we start by defining *interactions*.

Definition 2.15 – Interaction

Consider arenas A, B and C, and a pointing string u on $|\mathsf{A}| + |\mathsf{B}| + |\mathsf{C}|$. We note $u \upharpoonright \mathsf{A}, \mathsf{B}$ the subsequence of u of the moves played in A and B, seen as a pointing string on $\mathsf{A} \Rightarrow \mathsf{B}$, and preserving pointers – and likewise for $u \upharpoonright \mathsf{B}, \mathsf{C}$ and $u \upharpoonright \mathsf{A}, \mathsf{C}$.

Then u is an **interaction**, noted $u \in \mathrm{I}(\mathsf{A}, \mathsf{B}, \mathsf{C})$, if:

- $u \upharpoonright \mathsf{A}, \mathsf{B} \in \mathsf{Plays}(\mathsf{A} \Rightarrow \mathsf{B})$,
- $u \upharpoonright \mathsf{B}, \mathsf{C} \in \mathsf{Plays}(\mathsf{B} \Rightarrow \mathsf{C})$,
- $u \upharpoonright \mathsf{A}, \mathsf{C}$ is alternating.

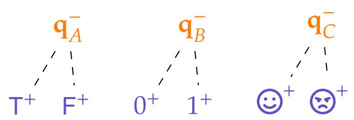


Figure 2.12: Arenas A := bool, B := bit and C := mood.

Remark that this definition ensures that for any $u \in \mathrm{I}(\mathsf{A}, \mathsf{B}, \mathsf{C})$, we have $u \upharpoonright \mathsf{A}, \mathsf{C} \in \mathsf{Plays}(\mathsf{A} \Rightarrow \mathsf{C})$. Consider for example the (very simple) arenas in Figure 2.12. Then the pointing string

$$u \quad := \quad \mathbf{q}_C \quad \mathbf{q}_B \quad \mathbf{q}_A \quad \mathsf{T} \quad 1 \quad ☺$$

is an interaction – notice how there is no indication of polarities – because its restrictions are all plays in the corresponding arenas:

$$u \upharpoonright A, B \;:=\; \mathbf{q}_B^- \;\; \mathbf{q}_A^+ \;\; \mathsf{T}^- \;\; 1^+ \quad \in \mathsf{Plays}(A \Rightarrow B),$$

$$u \upharpoonright B, C \;:=\; \mathbf{q}_C^- \;\; \mathbf{q}_B^+ \;\; 1^- \;\; ☺^+ \quad \in \mathsf{Plays}(B \Rightarrow C),$$

$$u \upharpoonright A, C \;:=\; \mathbf{q}_C^- \;\; \mathbf{q}_A^+ \;\; \mathsf{T}^- \;\; ☺^+ \quad \in \mathsf{Plays}(A \Rightarrow C).$$

Already we see some kind of compositional behavior: $u \upharpoonright A, C$ is the result of *hiding* the moves occuring in B when we "compose" the two plays $u \upharpoonright A, B$ and $u \upharpoonright B, C$.

We extend the definition of interaction to strategies:

Definition 2.16 – Interaction of strategies

Consider arenas A, B and C, with $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$.

We define the **interaction of** σ **and** τ as:

$$\tau || \sigma \stackrel{\text{def}}{=} \{u \in \mathrm{I}(A, B, C) \mid u \upharpoonright A, B \in \sigma \text{ and } u \upharpoonright B, C \in \tau\} .$$

The composition is then obtained by hiding the moves occuring in the shared arena component.

Definition 2.17 – Composition of strategies

Consider arenas A, B and C, with $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$.

We define the **composition of** σ **and** τ as:

$$\tau \odot^{\mathsf{HO}} \sigma \stackrel{\text{def}}{=} \{u \upharpoonright A, C \mid u \in \tau || \sigma\} .$$

Now, obviously we want the composition of two strategies to also be a strategy (see [27, Proposition 5.1] or [26, Proposition 2.5.3]).

[27]: Hyland and Ong (2000), 'On Full Abstraction for PCF: I, II, and III'

[26]: Harmer (2006), *Innocent game semantics*

Proposition 2.18 – Composition is well defined

Consider arenas A, B and C, with $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$.

Then $\tau \odot^{\mathsf{HO}} \sigma$ is a strategy on $A \Rightarrow C$.

Moreover, the composition of two *innocent* strategies is itself an innocent strategy (see [27, Proposition 5.3] or [26, Proposition 2.6.3]).

Proposition 2.19 – Composition preserves innocence

Consider arenas A, B and C, with innocent strategies $\sigma : A \Rightarrow B$ and $\tau : B \Rightarrow C$.

Then $\tau \odot^{\mathsf{HO}} \sigma$ is an innocent strategy on $A \Rightarrow C$.

2.5 HO and $\mathsf{HO}^{\mathsf{Inn}}$ as categories

Since composition behaves nicely both for strategies in general and for innocent strategies, it is natural to consider the *categorical structure* of arenas and strategies. We do not aim to give a detailed review on this subject here; we only state some results so as to better understand how the categorical structure of PCG – the model we focus on in this work – relates to HO and $\mathsf{HO}^{\mathsf{Inn}}$.

Theorem 2.20 – HO is a category

There is a category of strategies HO with arenas as objects and strategies as morphisms.

Again, we direct the reader to [27] or [26] for detailed statements.

Theorem 2.21 – $\mathsf{HO}^{\mathsf{Inn}}$ is a category

There is a category of *innocent* strategies $\mathsf{HO}^{\mathsf{Inn}}$ with arenas as objects and innocent strategies as morphisms.

Identity morphisms are called **copycat strategies** – they "copy" the behavior of Opponent, hence their name.

Definition 2.22 – Copycat strategy

Consider an arena A.

We define $\mathsf{cc}_{\mathsf{A}}^{\mathsf{HO}} : \mathsf{A}_\ell \Rightarrow \mathsf{A}_r$ the **copycat strategy** on A with:

for any $s \in \mathsf{Plays}(\mathsf{A}_\ell \Rightarrow \mathsf{A}_r)$, $s \in \mathsf{cc}_{\mathsf{A}}^{\mathsf{HO}}$ iff

1. $\forall t \sqsubseteq^+ s, t \upharpoonright \mathsf{A}_\ell = t \upharpoonright \mathsf{A}_r$,
2. if s_i^- and s_{i+1}^+ minimal in A, then s_{i+1}^+ points to s_i^-,

Notation: We write $\mathsf{A}_\ell \Rightarrow \mathsf{A}_r$ for $\mathsf{A} \Rightarrow \mathsf{A}$ to distinguish between the two copies of the arena A.

Cartesian structure. Both HO and $\mathsf{HO}^{\mathsf{Inn}}$ can be equiped with a cartesian structure, thanks to the product of arenas (Definition 2.4).

The projection $\pi_{\mathsf{A}}^{\mathsf{HO}} : \mathsf{A} \otimes \mathsf{B} \Rightarrow \mathsf{A}$ is the copycat-like strategy where Player duplicates every Opponent move played in one copy of A to the same move in the other copy of A (note that Opponent cannot change the arena component, so all moves stays in the two copies of A and no move is played in B). The projection $\pi_{\mathsf{B}}^{\mathsf{HO}}$ is defined in the same way.

Closed structure. Finally, recall the remark made about product and arrow constructions: it is immediate to check that for any arenas A, B and C, there is an isomorphism:

$$\Lambda^{\mathsf{HO}} : \quad (\mathsf{A} \otimes \mathsf{B}) \Rightarrow \mathsf{C} \quad \cong \quad \mathsf{A} \Rightarrow \mathsf{B} \Rightarrow \mathsf{C};$$

the only difference between the two construction being tags of moves. Applying Λ^{HO} to plays in strategies gives us the *currying isomorphism*:

$$\Lambda^{\mathsf{HO}} : \quad \mathsf{HO}(\mathsf{A} \otimes \mathsf{B}, \mathsf{C}) \quad \cong \quad \mathsf{HO}(\mathsf{A}, \mathsf{B} \Rightarrow \mathsf{C}) \,,$$

which in turn gives us the *evaluation morphism*:

$$\mathsf{ev}_{\mathsf{A},\mathsf{B}}^{\mathsf{HO}} \quad \stackrel{\text{def}}{=} \quad (\Lambda^{\mathsf{HO}})^{-1} \, (\mathsf{cc}_{\mathsf{A} \Rightarrow \mathsf{B}}^{\mathsf{HO}}) \quad \in \; \mathsf{HO}((\mathsf{A} \Rightarrow \mathsf{B}) \otimes \mathsf{A}, \mathsf{B}) \,.$$

These morphisms verify all the equations for a cartesian closed category. Moreover, Λ^{HO} preserves innocence.

Theorem 2.23 – HO and $\mathsf{HO}^{\mathsf{Inn}}$ are CCC's

HO and $\mathsf{HO}^{\mathsf{Inn}}$ are cartesian closed categories.

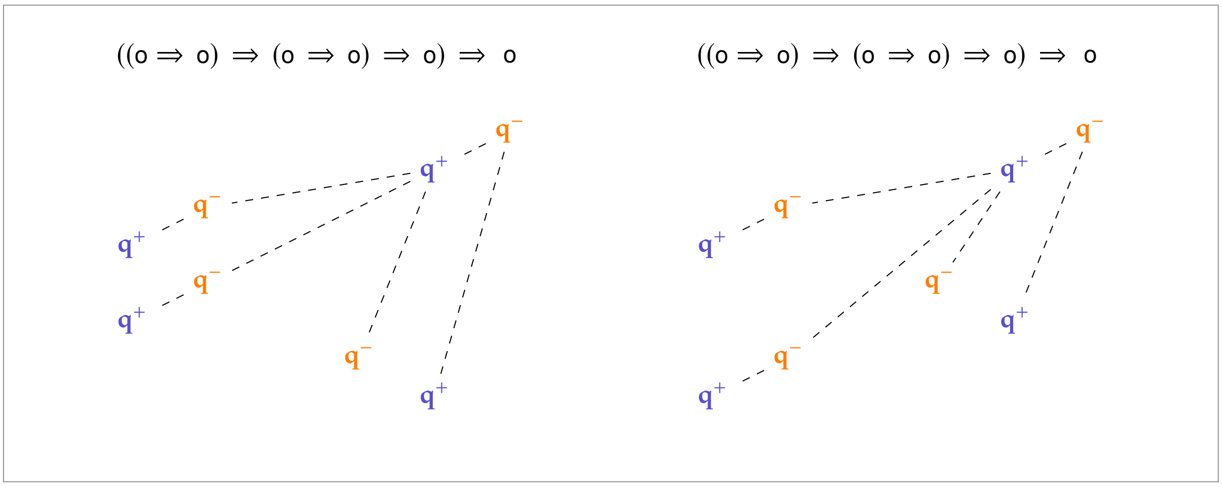


Figure 2.13: Two homotopic plays.

2.6 Links with the resource calculus

As mentionned in the introduction, the links between HO games and resource plays have already been investigated, for instance in [40].

[40]: Tsukada and Ong (2016), 'Plays as Resource Terms via Non-idempotent Intersection Types'

These links rely on Melliès' *homotopy relation*, introduced in [33]. We will define this homotopy relation more formally in the next chapter. Intuitively, it equates plays which only differ by Opponent's scheduling, as the two plays of Figure 2.13.

[33]: Melliès (2006), 'Asynchronous games 2: The true concurrency of innocence'

Theorem 2.24 – HO and the resource calculus [40]

There is a bijection between simply typed, β-normal, η-long resource terms, and HO plays up to homotopy.

Example: Consider the (simply typed, β-normal, η-long) resource term:

$$\vdash \lambda f.\, f\ [\lambda x.x, \lambda x.x]\ [\lambda y.f\ []\ []] : ((\alpha \to \alpha) \to (\alpha \to \alpha) \to \alpha) \to \alpha$$

Both plays of Figure 2.13 correspond to that resource term. In the first play, the first argument of the first call to f is evaluated twice, *and then* the second one is evaluated once; and none of the arguments to the second call to f is evaluated. The second play features the same moves, only in a different order.

An introduction to Pointer Concurrent Games

In this part, we introduce our Pointer Concurrent Games model. This model was motivated by the study of positional properties of innocent strategies in Hyland-Ong games: is the collapse of innocent strategies into the relation model injective? This question led us to design ***Pointer Concurrent Games****.*

In Chapter 3, we introduce the question of positional injectivity and we define ***configurations*** *and* ***augmentations****, our main mathematical objects. We show how this games model relates to traditional Hyland-Ong games.*

In Chapter 4, we present a first result obtained thanks to this games model: ***positional injectivity*** *for certain total augmentations in* PCG*, corresponding to a result of positional injectivity for total innocent strategies in* HO*.*
This chapter is more technical and is not needed to follow other parts.

Most of these results were presented in the article [4].

[4]: Blondeau-Patissier and Clairambault (2021), 'Positional Injectivity for Innocent Strategies'

Static PCG: Configurations and Augmentations

3

Before thoroughly defining pointer concurent games, we motivate our games model with a study of *positionality* / *positional injectivity* of innocent strategies. Indeed, at the core of pointer concurrent games are *positions*, which are moves and pointers without the sequential information given in a play. A desequentialized play induces a position, corresponding to its collapse in the relational model. But how much information about the play is preserved? Obviously, one cannot recover a play from any position. Consider for example the arena bool and the following plays:

$$s = \mathsf{q} - \mathsf{T} \quad \mathsf{q} - \mathsf{F} \qquad \text{and} \qquad t = \mathsf{q} - \mathsf{F} \quad \mathsf{q} - \mathsf{T}\,.$$

The only difference between s and t is the temporal order in which the pairs $\mathsf{q} - \mathsf{T}$ and $\mathsf{q} - \mathsf{F}$ occur, so once we forget that temporal order, we have no way of distinguishing them. But still, maybe positions of the plays of an innocent strategy can inform us on the strategy itself. This is what we investigate in this part. We start by defining positions and stating the problem of positional injectivity for HO games; then we introduce **augmentations** and we show how they relate to plays in HO games.

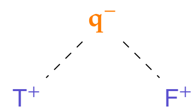


Figure 3.1: Arena bool

3.1 Relational Collapse

3.1.1 Configurations

We first define *configurations*, the actual mathematical objects we will be working with.

Definition 3.1 – Configuration

A **configuration** $x \in \mathsf{Conf}(\mathsf{A})$ of arena A is $x = \langle |x|, \leq_x, \partial_x \rangle$ such that $\langle |x|, \leq_x \rangle$ is a finite forest and ∂_x is a function $\partial_x : |x| \longrightarrow |\mathsf{A}|$ called the **display map**, subject to the conditions:

minimality-respecting: for any $a \in |x|$, a is $\leq_x$-minimal iff $\partial_x(a)$ is $\leq_{\mathsf{A}}$-minimal,

causality-preserving: for all $a_1, a_2 \in |x|$, if $a_1 \rightarrowtriangle_x a_2$ then $\partial_x(a_1) \rightarrowtriangle_{\mathsf{A}} \partial_x(a_2)$,

Reminder: $\langle |x|, \leq_x \rangle$ is a finite forest if it is a finite partially ordered set such that $\langle |x|, \rightarrowtriangle_x \rangle$ is a forest, with $\rightarrowtriangle_x$ the immediate causality relation defined by: $\forall a, b \in |x|$, $a \rightarrowtriangle_x b$ iff:

1. $a <_x b$,
2. $\forall c \in |x|$, if $a \leq_x c \leq_x b$, then $a = c$ or $b = c$.

We call **events** the elements of $|x|$. A (simple) configuration y is presented in Figure 3.2: its events are $|y| = \{a, b, c\}$, ordered with $a <_y b$, and the display map ∂_y is given alongside the forest (drawn from top to bottom).

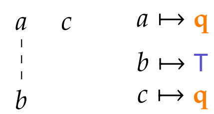


Figure 3.2: $y \in \mathsf{Conf}(\mathsf{bool})$

A configuration x on an arena A is **pointed**, noted $x \in \mathsf{Conf}_\bullet(\mathsf{A})$, if it has exactly one minimal event for $\leq_x$ (written $\mathsf{init}(x)$). The example configuration y from Figure 3.2 is not pointed (because a and c are both minimal for $\leq_y$), but z from Figure 3.3 is.

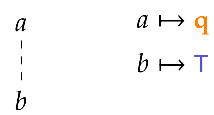


Figure 3.3: $z \in \mathsf{Conf}_\bullet(\mathsf{bool})$

Thanks to the display map, a polarity function on x can be deduced:

$$\mathsf{pol}_x(a) = \mathsf{pol}_{\mathsf{A}}(\partial_x(a))\,.$$

As with HO games, we write a^- (resp. a^+) for a such that $\mathsf{pol}_x(a) = -$ (resp. $\mathsf{pol}_x(a) = +$).

Any play induces a configuration *via* its desequentialization.

Definition 3.2 – Desequentialization

Consider an arena A and a play $s = s_1 \dots s_n \in \mathsf{Plays}(A)$. The **desequentialization** of s is $(\!|s|\!) = \langle |(\!|s|\!)|, \leq_{(\!|s|\!)}, \partial_{(\!|s|\!)} \rangle$ such that:

$$\begin{array}{rcl} |(\!|s|\!)| & = & \{1, \dots, n\}, \\ i \leq_{(\!|s|\!)} j & \Leftrightarrow & \text{there is a chain of pointers from } s_j \text{ to } s_i \text{ in } s, \\ \partial_{(\!|s|\!)}(i) & = & s_i\,. \end{array}$$

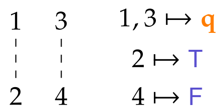


Figure 3.4: $(\!|s|\!)$ with $s = \mathsf{q} - \mathsf{T}\ \ \mathsf{q} - \mathsf{F}$

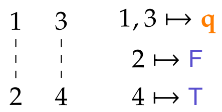


Figure 3.5: $(\!|t|\!)$ with $t = \mathsf{q} - \mathsf{F}\ \ \mathsf{q} - \mathsf{T}$

Figure 3.4 and Figure 3.5 present the desequentializations of the plays introduced at the beginning of this chapter.

If $s \in \mathsf{Plays}(\mathsf{A})$, then the definition of $(\!|-|\!)$ ensures that $(\!|s|\!) \in \mathsf{Conf}(\mathsf{A})$. Moreover, since $\leq_{(\!|s|\!)}$ follows the chains of pointers, it is clear that

$$(\!|s|\!) \in \mathsf{Conf}_\bullet(\mathsf{A}) \Leftrightarrow s \text{ is well-opened.}$$

3.1.2 Positions

As elements of $(\!|s|\!)$ are natural numbers reminiscent of the ordering, s can evidently be read back from $(\!|s|\!)$. However, in general, the exact name of events is not relevant: what we are really interested in is the display of those events in the arena, as well as their dependencies. Hence, we consider symmetries on configurations, which preserve the order relation and the display map; and we then quotient configurations by those symmetries.

Definition 3.3 – Symmetry

Consider $x, y \in \mathsf{Conf}(\mathsf{A})$. A **symmetry** $\varphi : x \cong_{\mathsf{A}} y$ is an isomorphism $\varphi : |x| \cong |y|$ preserving the order relation and the display map:

$$\begin{array}{rl} \textit{arena-preserving:} & \forall a \in |x|, \partial_y(\varphi(a)) = \partial_x(a), \\ \textit{causality-respecting:} & \forall a_1, a_2 \in |x|, a_1 \rightarrowtriangle_x a_2 \text{ iff } \varphi(a_1) \rightarrowtriangle_y \varphi(a_2). \end{array}$$

Consider again $(\!|s|\!)$ in Figure 3.4 and $(\!|t|\!)$ in Figure 3.5; then

$$\varphi : (\!|s|\!) \cong_{\mathsf{bool}} (\!|t|\!) \qquad \text{with } \varphi = \{1 \mapsto 3, 2 \mapsto 4, 3 \mapsto 1, 4 \mapsto 2\}\,.$$

Definition 3.4 – Position

A **position** of A, written $\mathsf{x} \in \mathsf{Pos}(\mathsf{A})$, is an isomorphism class of configurations on A.

A position x is **pointed**, written $x \in \mathsf{Pos}_\bullet(A)$, if any of its representatives is. If $x \in \mathsf{Conf}(A)$, we write $\overline{x} \in \mathsf{Pos}(A)$ for the corresponding position. Reciprocally, if $x \in \mathsf{Pos}(A)$, we fix $\underline{x} \in \mathsf{Conf}(A)$ a representative.

For any play $s \in \mathsf{Plays}_\bullet(A)$, its **position** $(\!|s|\!) \in \mathsf{Pos}(A)$ is the isomorphism class of $(\!|s|\!)$. The position of a play captures exactly the moves that have been played, along with their justification pointers; it is a snapshot of every interaction that occured between Opponent and Player at a given point, but without the order in which those interactions occurred. To represent positions graphically, one can draw forests of moves, where the nodes are the arena image of events (instead of their names as with configurations). For example, Figure 3.6 shows the position reached by both $s = \mathbf{q} - \mathsf{T}\ \mathbf{q} - \mathsf{F}$ and $t = \mathbf{q} - \mathsf{F}\ \mathbf{q} - \mathsf{T}$ (notice that the minimal nodes are incomparable).

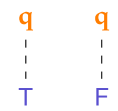


Figure 3.6: $(\!|s|\!) = (\!|t|\!)$

For any strategy $\sigma : A$, we define its **positions** $(\!|\sigma|\!)$ as the set of positions reached by well-opened plays, *i.e.*

$$(\!|\sigma|\!) = \{ (\!|s|\!) \mid s \in \sigma_\bullet \} \subseteq \mathsf{Pos}(A)\,.$$

Reminder: $\sigma_\bullet = \sigma \cap \mathsf{Plays}_\bullet(A)$, with $\mathsf{Plays}_\bullet(A)$ the well-opened plays on A.

Remark: We focus on *well-opened* plays, *i.e.* plays with only one initial moves, because those plays are the ones corresponding to points in Rel.

3.1.3 Relational Model

Positions of plays correspond to their collapse in the *relational model* [19], a **static** semantics where types are sets and programs are relations. More precisely, the relational model of the simply-typed λ-calculus is a cartesian closed category $\mathsf{Rel}_!$, with

[19]: Ehrhard (2012), 'The Scott model of linear logic is the extensional collapse of its relational model'

objects:	*sets*,
morphisms from E to F:	*relations* $R \subseteq \mathcal{M}_f(E) \times F$.

Reminder: $\mathcal{M}_f(E)$ is the set of finite multisets on E.

Consider simple types generated from the base type α and the arrow $\to$. We interpret them as:

$$\begin{aligned} [\![\alpha]\!]_{\mathsf{Rel}_!} &= \{\star\}\,, \\ [\![A \to B]\!]_{\mathsf{Rel}_!} &= \mathcal{M}_f([\![A]\!]_{\mathsf{Rel}_!}) \times [\![B]\!]_{\mathsf{Rel}_!}\,, \end{aligned}$$

where $\{\star\}$ is a singleton set. For example, the following relation

$$R = \{([\star], \star), ([\star, \star], \star)\}$$

is a subset of $[\![\alpha \to \alpha]\!]_{\mathsf{Rel}_!}$ (where multisets are noted with [ ] brackets).

Where does this relate to positions? First, we need to define thick subtrees (and subforests), a notion introduced by Boudes in [9]. Thick subtrees are rooted subtrees of a tree where some branches can be duplicated.

[9]: Boudes (2009), 'Thick Subtrees, Games and Experiments'

Definition 3.5 – Tree morphism

Consider two trees T and T'. A **tree morphism** $f : T \to T'$ is a function from the nodes of T to the nodes of T' which preserves the root of the tree and such that if $a \to_T b$, then $f(a) \to_{T'} f(b)$.

This definition can be extended to forests: forest morphisms preserve the roots of the forest as well as the immediate order, as in Figure 3.7.

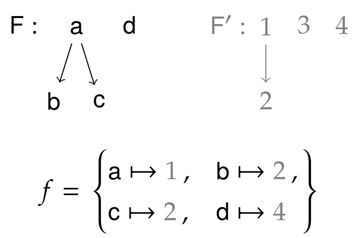


Figure 3.7: A forest morphism $f : \mathsf{F} \to \mathsf{F}'$

Definition 3.6 – Thick subtree

Consider a tree T. A **thick subtree** of T is $\langle T', f\rangle$ with T' a tree and $f : T' \to T$ a tree morphism.

This definition allows duplications of branches of the original tree, but ensures that no move can be copied without its predecessors.

Again, the definition can be generalized to forests – *e.g.* in Figure 3.7, $\langle \mathsf{F}', f\rangle$ is a thick subforest of F. For the sake of simplicity, we shall use "thick subtrees" for both thick subtrees and thick subforests.

[9]: Boudes (2009), 'Thick Subtrees, Games and Experiments'

Boudes [9, Proposition 2] showed that points of the web in relational semantics match thick subtrees (up to isomorphism) of arenas.

Consider a well-opened arena A, then $\langle |\mathsf{A}|, \rightharpoonup_{\mathsf{A}}\rangle$ is a tree. It is clear that configurations and positions of A match Boudes' thick subtrees: they represent partial explorations of A, where moves can be duplicated and must be *justified* by their ancestors. Moreover, consider two arenas A and B with B well-opened. There is a bijection:

$$\mathsf{Pos}(\mathsf{A} \Rightarrow \mathsf{B}) \cong \mathscr{M}_f(\mathsf{Pos}(\mathsf{A})) \times \mathsf{Pos}(\mathsf{B})$$

which matches exactly the definition of morphisms in $\mathsf{Rel}_!$.

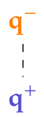


Figure 3.8: Arena $[\![\alpha \to \alpha]\!]_{\mathsf{Inn}}$

For any simple type A, considering its interpretation as an arena $[\![A]\!]_{\mathsf{Inn}}$, there is a bijection

$$\mathsf{R}_A : \mathsf{Pos}([\![A]\!]_{\mathsf{Inn}}) \cong [\![A]\!]_{\mathsf{Rel}_!} .$$

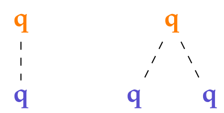


Figure 3.9: R as a thick subtree

Recall for example the relation:

$$R = \{([\star], \star), ([\star, \star], \star)\} \subseteq [\![\alpha \to \alpha]\!]_{\mathsf{Rel}_!} .$$

It is easy to see it as a thick subtree of $[\![\alpha \to \alpha]\!]_{\mathsf{Inn}}$ as in Figure 3.9, where the elements in a multiset in the left-hand side of a pair are Player moves, and the elements in the right-hand side of a pair are Opponent moves.

Actually, this extends to a functor $\mathsf{R}_A(\langle\!| - |\!\rangle) : \mathsf{Inn} \to \mathsf{Rel}_!$ which preserves the interpretation: for any simply-typed λ-term $M : A$,

$$\mathsf{R}_A(\langle\!| [\![M]\!]_{\mathsf{Inn}} |\!\rangle) = [\![M]\!]_{\mathsf{Rel}_!} .$$

[9]: Boudes (2009), 'Thick Subtrees, Games and Experiments'

[33]: Melliès (2006), 'Asynchronous games 2: The true concurrency of innocence'

[13]: Castellan, Clairambault, Paquet, and Winskel (2018), 'The concurrent game semantics of Probabilistic PCF'

[16]: Clairambault and Visme (2020), 'Full abstraction for the quantum lambda-calculus'

This *relational collapse* of innocent strategies is well-known. The inclusion $\subseteq$ is easy; the difficulty in proving $\supseteq$ is that game-semantic interaction is temporal: positions arising relationally might, in principle, fail to appear game-semantically because reproducing them yields a deadlock. For innocent strategies this does not happen: this was proved for HO polarized games [9, Theorem 7], asynchronous games [33, Proposition 4], probabilistic thin concurrent games [13, Lemma 3.12] or even quantum games [16, Theorem 5.7].

3.2 Positional Injectivity

3.2.1 Positionality

Before focusing on *positional injectivity*, we take a look at the stronger property of *positionality*. A strategy is **positional** if its behavior only depends of the current *position*, not the current *play*.

Definition 3.7 – Positionality

Consider $\sigma : \mathsf{A}$ a strategy on A. We set the condition:

positional: $\forall s\mathsf{ab}, t \in \sigma, t\mathsf{a}' \in \mathsf{Plays}(\mathsf{A}),$
$(\!| s\mathsf{a} |\!) = (\!| t\mathsf{a}' |\!) \Rightarrow \exists t\mathsf{a}'\mathsf{b} \in \sigma, (\!| s\mathsf{ab} |\!) = (\!| t\mathsf{a}'\mathsf{b} |\!)\,.$

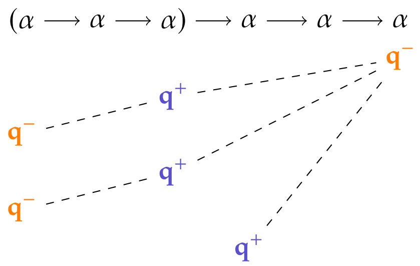
This is a rather strong requirement, which has also already been studied in the litterature. In Melliès' asynchronous games [33] for example, events carry explicit copy indices that help distinguish duplications of the same moves, so innocent strategies are positional [33, Theorem 2].

[33]: Melliès (2006), 'Asynchronous games 2: The true concurrency of innocence'

But what about innocent strategies for HO games? It is quite immediate to find counter-examples to positionality. Consider for example the term:

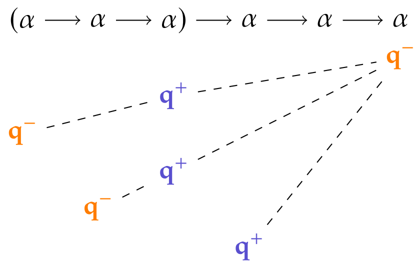
$$M = \lambda\, f^{\alpha\to\alpha\to\alpha}.\,\lambda\, x^{\alpha}.\,\lambda\, y^{\alpha}.\, f\,(f\,x\,x)\,(f\,y\,y)$$

whose interpretation $[\![M]\!]_{\mathsf{Inn}}$ is the innocent strategy with four maximal P-views given in Figure 3.10.



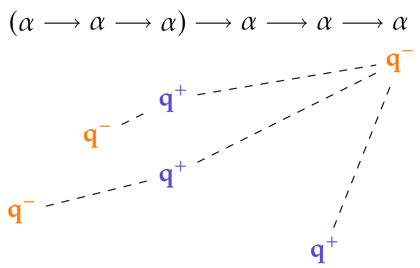
(a) The play $s\mathsf{ab}$ corresponding to the evaluation:

$\lambda f.\,\lambda x.\,\lambda y.\underline{f}\;(\underline{f}\;\underline{x}\;x)\,(f\;y\;y)\,.$



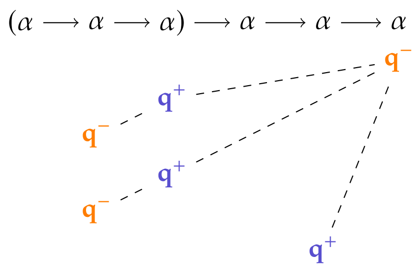
(b) The play $s\mathsf{a}'\mathsf{b}$ corresponding to the evaluation:

$\lambda f.\,\lambda x.\,\lambda y.\underline{f}\;(\underline{f}\;x\;\underline{x})\,(f\;y\;y)\,.$



(c) The play $t\mathsf{ac}$ corresponding to the evaluation:

$\lambda f.\,\lambda x.\,\lambda y.\underline{f}\;(f\;x\;x)\,(\underline{f}\;\underline{y}\;y)\,.$



(d) The play $t\mathsf{a}'\mathsf{c}$ corresponding to the evaluation:

$\lambda f.\,\lambda x.\,\lambda y.\underline{f}\;(f\;x\;x)\,(\underline{f}\;y\;\underline{y})\,.$

Figure 3.10: The four P-views of the meagre interpretation of M.

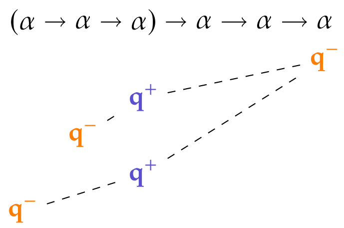


Figure 3.11: The *position* $(\!| sa' |\!) = (\!| ta |\!)$.

Reminder: $(\!|\sigma|\!)$ is the set of positions reached by well-opened plays of σ.

Then in particular both prefixes $s\mathsf{a}'$ (Subfigure 3.10b) and $t\mathsf{a}$ (Subfigure 3.10c) reach the same position (Figure 3.11), but by determinism $t\mathsf{a}$ cannot be extended with b into a play of $[\![M]\!]_{\mathsf{Inn}}$.

Hence, positionality fails in general for innocent strategies.

3.2.2 Positional Injectivity

We now turn ourselves to the weaker condition of *positional injectivity*: can an innocent strategy be uniquely identified by its positions? In other words, is the relational collapse $(\!| - |\!)$ injective?

Definition 3.8 – Positional Injectivity

A set of strategies $\mathcal{S}$ is **positionally injective** if for any $\sigma, \tau \in \mathcal{S}$,

$$(\!|\sigma|\!) = (\!|\tau|\!) \Rightarrow \sigma = \tau\,.$$

So, our main question is:

Question 4: are innocent strategies positionally injective?

[40]: Tsukada and Ong (2016), 'Plays as Resource Terms via Non-idempotent Intersection Types'

Tsukada and Ong [40] already studied the relational collapse of innocent strategies, but their interpretation in $\mathsf{Rel}_!$ is parametrized by a set X for the base type α. In [40], X is required to be countably infinite: this way one allocates one tag for each pair of chronologically contiguous O/P moves, encoding the *causal / axiom links*. In contrast, here we wish to interpret α with a singleton set $\{\mathbf{q}\}$, lest we lose the correspondence between points of the web and positions.

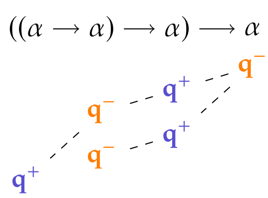
Unlike in [40], we cannot reconstruct an innocent strategy from the positions of its P-views only. Consider the infamous "Kierstead terms"

[30]: Kierstead (1980), 'A Semantics for Kleene's j-expressions'

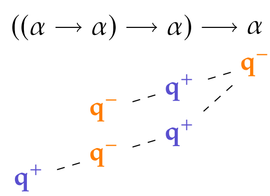
$$K_x = \lambda f^{(\alpha\to\alpha)\to\alpha}.\, f\ (\lambda x^\alpha.\, f\ (\lambda y^\alpha.x))$$

$$K_y = \lambda f^{(\alpha\to\alpha)\to\alpha}.\, f\ (\lambda x^\alpha.\, f\ (\lambda y^\alpha.y))$$

(which seem to first appear in [30, Example 3.6]). They only differ by the very last variable. The strategies $[\![K_x]\!]_{\mathsf{Inn}}$ and $[\![K_y]\!]_{\mathsf{Inn}}$ are innocent, and characterized by the following maximal P-views:



$s_x \in [\![K_x]\!]_{\mathsf{Inn}}$



$s_y \in [\![K_y]\!]_{\mathsf{Inn}}$.

Notice they only differ by the last move. However, this difference disappears once we forget the temporal order: both plays clearly reach the same position (*i.e.* a tree with two un-ordered branches).

Hence, P-views are not enough to positionally distinguish $[\![K_x]\!]_{\mathsf{Inn}}$ and $[\![K_y]\!]_{\mathsf{Inn}}$. Does this mean both strategies have the same positions?

In each play, let us duplicate the Opponent move which the deepest $\mathbf{q}^+$ points to – so, the third move of s_x and the fifth move of s_y. Since both strategies are innocent, they react by duplicating the following Player move: the fourth move for s_x, and the last one for s_y. We obtain the plays s'_x and s'_y presented in Figure 3.12 and Figure 3.13. It is clear that those two plays *do not* reach the same position – for a start, the root of $(\!|s'_x|\!)$ has degree 3 while the root of $(\!|s'_y|\!)$ has degree 2. But more importantly, $(\!|s'_x|\!)$ will never be reached by a play of $[\![K_y]\!]_{\mathsf{Inn}}$ – and conversely, $(\!|s'_y|\!) \notin (\!|[\![K_x]\!]_{\mathsf{Inn}}|\!)$.

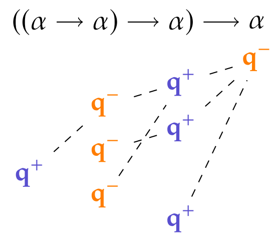


Figure 3.12: $s'_x \in [\![K_x]\!]_{\mathsf{Inn}}$.

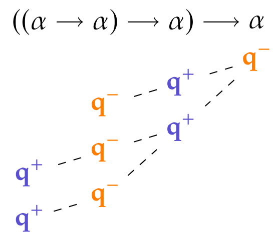


Figure 3.13: $s'_y \in [\![K_y]\!]_{\mathsf{Inn}}$.

By replicating Opponent moves, we are able to exhibit positions distinguishing the two strategies: the static behaviour of an innocent strategy under replication somehow informs us on temporality.

Most of Chapter 4 will be devoted to turning this idea into a proof. However, we have only been able to prove the result for *total finite* innocent strategies; actually, we know that positional injectivity fails in the case of infinite partial innocent strategies.

Before moving on to the proof, we introduce the main protagonists of pointer concurrent games: **augmentations**.

3.3 Augmentations

In order to identify strategies from their positions, we need to look at plays where Opponent *duplicates* moves. But such plays also contains the order in which Opponent performs the duplications, which is actually not relevant for our purposes since innocent strategies react in the same way to each duplication, no matter the order. Instead, we only want to look at the causal behavior of Player: we are interested in Player's point of view, and they don't know the number or order of duplications. Thus, we introduce *augmentations*, a *causal* version of plays and strategies inspired from concurrent games.

Intuitively, augmentations are trees of P-views; this connection with HO games is detailed in Section 3.4.

Reminder: P-views are plays where Opponent moves always point to the previous move (except for the initial move) – see Definition 2.9 in Chapter 2.

3.3.1 Definitions

Augmentations are configurations *augmented* with the causal order of events from Player.

Definition 3.9 – Augmentation

An **augmentation** on a negative arena A is $q = \langle |q|, \leq_{(\!|q|\!)}, \leq_q, \partial_q \rangle$ such that $(\!|q|\!) = \langle |q|, \leq_{(\!|q|\!)}, \partial_q \rangle \in \mathsf{Conf}(\mathsf{A})$ and $\langle |q|, \leq_q \rangle$ is a forest satisfying:

rule-abiding: if $a \leq_{(\!|q|\!)} b$, then $a \leq_q b$,
courteous: if $a \rightarrow_q b$ and $\mathsf{pol}(a) = +$ or $\mathsf{pol}(b) = -$, then $a \rightarrow_{(\!|q|\!)} b$,
deterministic: if $a \rightarrow_q b^+$ and $a \rightarrow_q c^+$, then $b = c$,
negative: if a is minimal for $\leq_q$, then $\mathsf{pol}(a) = -$,
+-covered: if a is maximal for $\leq_q$, then $\mathsf{pol}(a) = +$,

where pol is the polarity function deduced through the display map ∂_q. We write $q \in \mathsf{Aug}(\mathsf{A})$, and we say that $(\!|q|\!) \in \mathsf{Conf}(\mathsf{A})$ is the **desequentialization** of q or its **underlying configuration**.

Remark: We could relax some of these conditions and study for example non negative augmentations, or non +-covered ones, or even ∞-augmentations with infinitely many events. All these extensions are interesting and some of them will be discussed later, but for now on we focus on *negative +-covered finite* augmentations

Remark that by courtesy and rule-abiding, $a \in |q|$ minimal for $\leq_q$ implies a minimal for $\leq_{(\!|q|\!)}$. Since ∂_q preserves minimality, this implies $\partial_q(a)$ minimal in A.

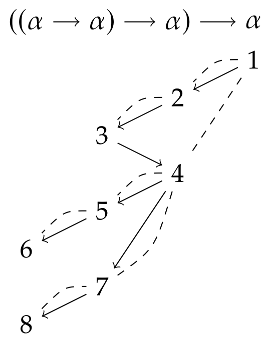


Figure 3.14: An augmentation q.

Consider Figure 3.14. It shows an augmentation q whose underlying configuration is $(\!|s'_y|\!)$, where s'_y is the play presented in Figure 3.13. The causal order $\rightarrow_q$ is noted with arrows, the static order $\rightarrow_{(\!|q|\!)}$ with dashed lines (read from top to bottom), and the arena image is given by the position of each event under its corresponding type component. Unlike plays, augmentations are not sequential: the vertical order here does not inform us on $\rightarrow_q$, and some events may be placed above or under others only for readability's sake. Following the condition *courteous*, the last two opponent moves of s'_y (namely 5 and 7) are incomparable in q: both are immediate successors of 4 for $\rightarrow_q$ (and $\rightarrow_{(\!|q|\!)}$).

By forestiality of $(\!|q|\!)$, for any $a \in |q|$ non minimal for $\leq_{(\!|q|\!)}$, there is a unique $b \in |q|$ such that $b \rightarrow_{(\!|q|\!)} a$. We say b is the **justifier** of a, written $b = \mathsf{just}(a)$. Likewise, if a is non minimal for $\leq_q$, then by rule-abiding there is a unique $c \rightarrow_q a$. We say c is the **predecessor** of a, written $c = \mathsf{pred}(a)$. Since arenas are alternating, we have

$$\mathsf{pol}(a) \neq \mathsf{pol}(b) \quad \text{and} \quad \mathsf{pol}(a) \neq \mathsf{pol}(c)\,.$$

The predecessor and the justifier of a can be different (*e.g.* $\mathsf{pred}(4) = 3$ and $\mathsf{just}(4) = 1$ in Figure 3.14), but they must coincide when $\mathsf{pol}(a) = -$ by courtesy. This corresponds to the construction of the P-view of a play, where we jump directly from an Opponent move to the Player move which justifies it – which is why augmentations are really trees of P-views, as we shall see in the next section.

3.3.2 Isogmentations

As with configurations, we care about the arena image and the order relations, but not so much about the identity of events. Hence, we define (iso)morphisms of augmentations.

Definition 3.10 – Augmentation (iso)morphism

Consider $q, p \in \mathsf{Aug}(\mathsf{A})$. An **augmentation morphism** $\varphi : q \rightarrow p$ is a morphism $\varphi\colon |q| \longrightarrow |p|$ with the properties:

$$\begin{array}{rl} \textit{arena-preserving:} & \partial_p \circ \varphi = \partial_q\,, \\ \textit{causality-preserving:} & \text{if } a \rightarrow_q b, \text{ then } \varphi(a) \rightarrow_p \varphi(b), \\ \textit{configuration-preserving:} & \text{if } a \rightarrow_{(\!|q|\!)} b, \text{ then } \varphi(a) \rightarrow_{(\!|p|\!)} \varphi(b). \end{array}$$

An **augmentation isomorphism**, noted $\varphi\colon q \cong p$, is an invertible morphism.

Remark that this definition ensures that the roots of an augmentation are preserved by morphisms.

Definition 3.11 – Isogmentation

An **isogmentation** of A, written $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$, is an isomorphism class of augmentations on A.

We write $\overline{q} \in \mathsf{Isog}(\mathsf{A})$ for the isomorphism class of $q \in \mathsf{Aug}(\mathsf{A})$, and we fix $\underline{\mathsf{q}} \in \mathsf{Aug}(\mathsf{A})$ a representative of $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$ (note the change of fonts). Remark that in particular, an augmentation isomorphism is a configuration isomorphism – hence isogmentations are compatible with positions. Isogmentations will be represented as in Figure 3.15, where we write directly the arena image of events instead of their identity.

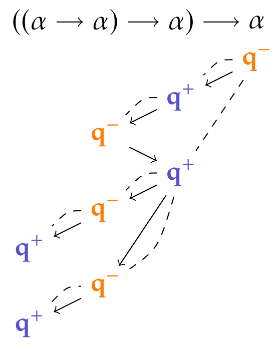


Figure 3.15: The isogmentation $\mathsf{q} = \overline{q}$.

Lemma 3.12 – Representatives and isomorphism classes

Consider $q \in \mathsf{Aug}(\mathsf{A})$ and $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$. Then,

$$\underline{(\overline{q})} \cong q \qquad \text{and} \qquad \overline{(\underline{\mathsf{q}})} = \mathsf{q}\,.$$

Proof. By definition. □

3.3.3 Additional Conditions on Augmentations

Before linking augmentations with P-views and plays, we define several additional conditions.

Definition 3.13 – Pointed, −-linear and total augmentations

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$. We set the conditions:

pointed: $\leq_q$ has only one minimal event,
−-linear: for any $a^-, b^- \in |q|$,
if $a, b \in \min_{\leq_q}(q)$ or $\mathsf{pred}(a) = \mathsf{pred}(b)$,
then $a = b$ or $\partial_q(a) \neq \partial_q(b)$.
total: for any $a^+ \in |q|$, if $\partial_q(a) \rightarrow_{\mathsf{A}} b'$,
there exists $b \in |q|$ s.t. $\partial_q(b) = b'$ and $a \rightarrow_q b$.

If q is pointed, we write $\mathsf{init}(q)$ for its unique minimal event, and we write $\mathsf{Aug}_\bullet(\mathsf{A})$ the set of pointed augmentations on A.

Reminder: for any $a \in |q|$ non-minimal for $\leq_q$, $\mathsf{pred}(a)$ is the predecessor of a, *i.e.* the only $a' \in |q|$ such that $a' \rightarrow_q a$.

The *totality* condition only seems to constrain Opponent – whenever a move is available to Opponent, they must play it. However, Player must react to any Opponent move since augmentations are +-covered. Hence, the totality condition ensures that both Opponent *and* Player keep playing until they reach maximal events.

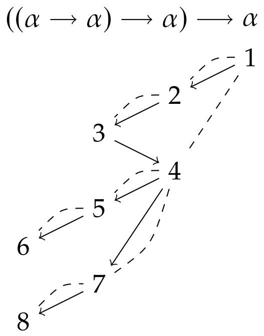


Figure 3.14: An augmentation q.

For example, the augmentation from Figure 3.14 is:

- pointed (the only initial event is 1),
- total (2 and 4 both have successors with the right arena image),
- not $-$-linear (because of 5 and 7).

These three conditions are stable by isomorphism, so it makes sense to define **pointed**, **$-$-linear** and **total isogmentations** as isogmentations for which any representative is respectively pointed, $-$-linear or total.

Moreover, they only constrain minimal events and links from positive events to negative ones; so $-$-linearity and totality can actually be defined as properties of the underlying configuration.

Definition 3.14 – $-$-linear and total configurations

Consider a configuration $x \in \mathsf{Conf}(\mathsf{A})$. We set the conditions:

$-$-linear:	for any $a^-, b^- \in |x|$, if $a, b \in \min(x)$ or $\mathsf{just}(a) = \mathsf{just}(b)$, then $a = b$ or $\partial_x(a) \neq \partial_x(b)$.
total:	for any $a^+ \in |x|$, if $\partial_x(a) \rightarrowtriangle_{\mathsf{A}} \mathsf{b}'$, there exists $b \in |x|$ s.t. $\partial_x(b) = \mathsf{b}'$ and $a \rightarrowtriangle_x b$.

Lemma 3.15

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$. Then:

- $q \in \mathsf{Aug}_\bullet(\mathsf{A})$ if $\llparenthesis q \rrparenthesis \in \mathsf{Conf}_\bullet(\mathsf{A})$,
- *if* A *is negative*, $q \in \mathsf{Aug}_\bullet(\mathsf{A})$ if and only if $\llparenthesis q \rrparenthesis \in \mathsf{Conf}_\bullet(\mathsf{A})$,
- q is $-$-linear if and only if $\llparenthesis q \rrparenthesis$ is $-$-linear,
- q is total if and only if $\llparenthesis q \rrparenthesis$ is total.

Proof. Immediate by courtesy and definitions:

Pointedness. We have $\min_{\leq_q}(q) \subseteq \min_{\leq_{\llparenthesis q \rrparenthesis}}(q)$. It A is negative, the inclusion is actually an equality by courtesy.

$-$-linearity. For any $a^- \in |q|$, $\mathsf{pred}(a) = \mathsf{just}(a)$ by courtesy.

Totality. For any $a^+, b \in |q|$, we know that $a^+ \rightarrowtriangle_q b$ if and only if $a^+ \rightarrowtriangle_{\llparenthesis q \rrparenthesis} b$ by courtesy. □

Since these properties are stable by *configuration isomorphisms*, we can even consider **$-$-linear or total positions** – *i.e.* positions for which any representative is respectively $-$-linear or total.

3.4 Augmentations in PCG *v.* Plays in HO

We introduced augmentations as "trees of P-views". Indeed, recall that P-views are plays in which negative moves always point to their predecessor (save from the initial move). This is similar to the *courtesy* condition of augmentations, which implies that in an augmentation q, for any negative event $a^- \in |q|$, $\mathsf{just}(a) = \mathsf{pred}(a)$. Hence, a P-view already *is* an augmentation, where the causal order is given by the sequential order and the static order by pointers.

But what about plays that may not be P-views? Augmentations intuitively represent "the tree – or forest – of all the P-views of the prefixes of a visible play". In other words, augmentations are visible plays, but quotiented by the order in which Opponent chooses to move from one program thread to another. This quotient is formalized with Melliès' homotopy equivalence [33].

[33]: Melliès (2006), 'Asynchronous games 2: The true concurrency of innocence'

3.4.1 Homotopy relation

Definition 3.16 – Melliès' homotopy relation

Consider two visible plays s and s' on an arena A. Then $s \sim_H s'$ iff

$$s = t\ \mathsf{a}_1^-\ \mathsf{b}_1^+\ \mathsf{a}_2^-\ \mathsf{b}_2^+\ t' \qquad \text{and} \qquad s' = t\ \mathsf{a}_2^-\ \mathsf{b}_2^+\ \mathsf{a}_1^-\ \mathsf{b}_1^+\ t'$$

with the same pointers.

Reminder: The *legality* condition of plays requires that each non initial move is *justified* by a pointer to an earlier move.

Remark that since s and s' are both *legal*, a_2 does not point to b_1 (and conversely a_1 does not point to b_2).

The example plays in $\mathsf{Plays}(\mathsf{bool})$ from the beginning of this chapter,

$$s = \mathsf{q} - \mathsf{T}\ \ \mathsf{q} - \mathsf{F} \qquad \text{and} \qquad t = \mathsf{q} - \mathsf{F}\ \ \mathsf{q} - \mathsf{T},$$

are such that $s \sim_H t$.

Definition 3.17 – Melliès' homotopy equivalence

We define $\sim_E$ the reflexive transitive closure of $\sim_H$.

Remark: $\sim_H$ is symmetric by definition.

Since plays are defined on negative arenas, let us consider a fixed negative arena A. We want to prove that augmentations – or rather, isogmentations – on A are isomorphic to P-visible positive plays quotiented by $\sim_E$:

Notation: We write $\mathsf{VisPlays}^+(\mathsf{A})$ for the set of P-visible positive plays on A.

Claim 1: There is a bijection $\chi : \mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E} \cong \mathsf{Isog}(\mathsf{A})$.

In order to prove this claim, we first define each side of the bijection, and then show they are inverses.

3.4.2 From plays to isogmentations

We start by defining the construction from $\mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E}$ to $\mathsf{Isog}(\mathsf{A})$.

Given a play, we construct an augmentation with the causal order following the construction of a P-view.

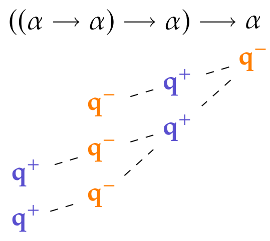


Figure 3.13: $s'_y \in [\![K_y]\!]_{\mathsf{Inn}}$.

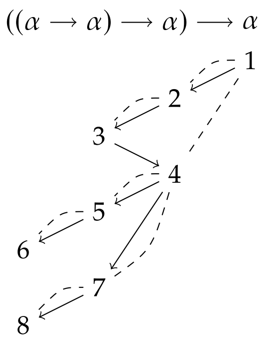


Figure 3.14: An augmentation q.

Reminder: see Definition 3.1.
Minimality-respecting:

$$a \in \min_{\leq_x}(|x|) \Leftrightarrow \partial_x(a) \in \min_{\leq_{\mathsf{A}}}(|\mathsf{A}|)\,.$$

Causality-preserving:

$$\text{if } a \rightarrowtriangle_x b \text{ then } \partial_x(a) \rightarrowtriangle_{\mathsf{A}} \partial_x(b)\,.$$

Definition 3.18 – Augmentation from play

Consider $s = s_1 \dots s_n \in \mathsf{VisPlays}^+(\mathsf{A})$. We construct $q = \mathsf{aug}(s)$ as:

$$\begin{array}{rcl} |q| & = & \{1, \dots, n\}\,, \\ \partial_q(i) & = & s_i\,, \\ i \rightarrowtriangle_{(\!|q|\!)} j & \text{iff} & s_j \text{ points to } s_i\,, \\ i^- \rightarrowtriangle_q j^+ & \text{iff} & j = i+1\,, \\ i^+ \rightarrowtriangle_q j^- & \text{iff} & s_j \text{ points to } s_i\,. \end{array}$$

Recall the play s'_y from Figure 3.13; then $\mathsf{aug}(s'_y)$ is the augmentation q from Figure 3.14.

It is clear – checking the conditions one by one – that $\mathsf{aug}(-)$ always constructs an augmentation.

Lemma 3.19 – $\mathsf{aug}(s)$ is an augmentation

Consider $s = s_1 \dots s_n \in \mathsf{VisPlays}^+(\mathsf{A})$. Then $\mathsf{aug}(s) \in \mathsf{Aug}(\mathsf{A})$.

Proof. We write $q = \mathsf{aug}(s)$.

First, we check that $(\!|q|\!) = \langle |q|, \leq_{(\!|q|\!)}, \partial_q \rangle$ is a configuration.

Finite forest. Since s is finite, so is $|q|$. It is clear from definition that $\langle |q|, \leq_{(\!|q|\!)} \rangle$ is a finite forest.

Minimality-respect. Clear from definition and legality of s.

Causality-preservation. Consider $i, j \in |q|$ such that $i \rightarrowtriangle_{(\!|q|\!)} j$. Then by definition s_j points to s_i in s. By rigidity of plays, $\partial_q(i) \rightarrowtriangle_{\mathsf{A}} \partial_q(j)$.

Hence, $(\!|q|\!) \in \mathsf{Conf}(\mathsf{A})$. We now check q is an augmentation.

Forestiality. By definition of the prefix order, it is clear that $\langle |q|, \leq_q \rangle$ is a forest.

Rule-abidingness. Consider $i, j \in |q|$ such that $i \leq_{(\!|q|\!)} j$. By definition, s_j points to s_i. If $\mathsf{pol}(i) = -$, then $j = i+1$ since s is visible. In both cases, $i \rightarrowtriangle_q j$.

Courtesy. Consider $i \rightarrowtriangle_q j$ such that $\mathsf{pol}(i) = +$ or $\mathsf{pol}(j) = -$. By definition of $\rightarrowtriangle_q$, we have s_j) points to s_j, so $i \rightarrowtriangle_{(\!|q|\!)} j$.

Determinism. Consider $i^- \rightarrowtriangle_q j$ and $i^- \rightarrowtriangle_q j'$, then $j = j' = i+1$.

Negativity. By negativity of A.

+-coveredness. Because $s \in \mathsf{Plays}^+(\mathsf{A})$.

Hence, $q \in \mathsf{Aug}(\mathsf{A})$. □

Moreover, this construction preserves homotopy, in the sense that homotopic plays become isomorphic augmentations.

Lemma 3.20 – Homotopic plays imply isomorphic aug.

Consider $s, t \in \mathsf{VisPlays}^+(\mathsf{A})$ such that $s \sim_E t$. Then

$$\mathsf{aug}(s) \cong \mathsf{aug}(t)\,.$$

Proof. By induction on $s \sim_E t$. For the base case, if

$$s = u\ \mathsf{a}_1\ \mathsf{b}_1\ \mathsf{a}_2\ \mathsf{b}_2\ v \qquad \text{and} \qquad t = u\ \mathsf{a}_2\ \mathsf{b}_2\ \mathsf{a}_1\ \mathsf{b}_1\ v$$

with $|u| = k$, we construct the isomorphism

$$\begin{array}{llll} \varphi: & \mathsf{aug}(s) & \cong & \mathsf{aug}(t) \\ & k+1 & \mapsto & k+3 \\ & k+2 & \mapsto & k+4 \\ & k+3 & \mapsto & k+1 \\ & k+4 & \mapsto & k+2 \\ & i & \mapsto & i \qquad \text{otherwise} \end{array}$$

and one can check it is indeed an augmentation isomorphism. □

Consider an equivalence class $\mathbf{s} \in \mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E}$. Thanks to the above lemma, we define $\mathbf{isog}(\mathbf{s}) \in \mathsf{Isog}(\mathsf{A})$ as

$$\mathbf{isog}(\mathbf{s}) = \overline{\mathsf{aug}(s)} \text{ for any } s \in \mathbf{s}\,.$$

We have now described one side of the isomorphism from **Claim 1**:

$$\chi: \mathbf{s} \in \mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E} \quad \mapsto \quad \mathbf{isog}(\mathbf{s}) \in \mathsf{Isog}(\mathsf{A})\,.$$

Before proving it is indeed a bijection, we focus on the reverse operation.

3.4.3 From isogmentations to plays

Plays are obtained from alternating linearisations of augmentations.

Definition 3.21 – Alternating linearisation

Consider $q \in \mathsf{Aug}(\mathsf{A})$. An **alternating linearisation** of q is a total order on the events of q, noted $\mathsf{t} = \mathsf{t}_1 \dots \mathsf{t}_n$ with $\{\mathsf{t}_i \mid 1 \le i \le n\} = |q|$, such that:

$$\begin{array}{ll} \textit{polarity-alternating:} & \forall i < n,\ \mathsf{pol}(\mathsf{t}_i) \neq \mathsf{pol}(\mathsf{t}_{i+1})\,. \\ \textit{causality-respecting:} & \forall i < n,\ \mathsf{t}_i \le_q \mathsf{t}_{i+1}\,. \end{array}$$

We write $\mathsf{Alt}(q)$ for the set of **alternating linearisations** of q.

For instance, the augmentation q from Figure 3.14 admits two alternating linearisations:

$$\mathsf{Alt}(q) = \{1\,2\,3\,4\,5\,6\,7\,8\,,\,1\,2\,3\,4\,7\,8\,5\,6\}\,.$$

By determinism of plays, alternating linearisations preserve the immediate order between negative and positive moves.

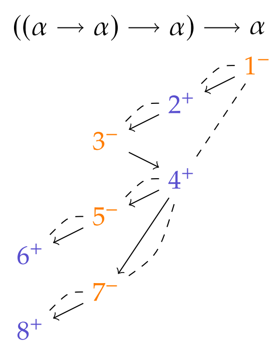


Figure 3.14: An augmentation q.

Lemma 3.22 – Alternating linearisations preserve O–P pairs

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$.

For any $\mathsf{t} \in \mathsf{Alt}(q)$, if $a^- \rightarrowtriangle_{\mathsf{t}} b^+$ then $a^- \rightarrowtriangle_q b^+$.

Proof. Since A is negative, t starts with a negative move. Consider the prefixes $\mathsf{t}^i = \mathsf{t}_1 \dots \mathsf{t}_i \sqsubseteq \mathsf{t}$. By induction on i, we prove that:

- if $i = 2k$ (*i.e.* t^i ends with a positive event), all the maximal events of t^i are positive;
- if $i = 2k+1$ (*i.e.* t^i ends with a negative event), all the maximal events of t^i are positive *except exactly one negative event*.

Since t starts with a negative move, the invariant H_i is true for $i = 1$.

Since t is alternating, H_{2k} directly implies H_{2k+1}.

Finally, if $i = 2k+1$, then by H_i t^i has exactly one maximal negative event. By determinism there is exactly one "available" positive move next, *i.e.* a positive move that has not been used yet in t^i and is an immediate successor of an event of t^i. Hence $\mathsf{t}_{i+1} = \mathsf{succ}(\mathsf{t}_i)$, and every maximal event of t^{i+1} is positive. □

These alternating linearisations can be translated to plays thanks to the display map.

Definition 3.23 – Display map of a linearisation

Consider $q \in \mathsf{Aug}(\mathsf{A})$ and $\mathsf{t} \in \mathsf{Alt}(q)$, noted $\mathsf{t} = \mathsf{t}_1 \dots \mathsf{t}_n$. We define:

$$\partial_q(\mathsf{t}) = \partial_q(\mathsf{t}_1) \dots \partial_q(\mathsf{t}_n) \ \in \ \mathsf{VisPlays}^+(A)$$

where $\partial_q(\mathsf{t}_j)$ points to $\partial_q(\mathsf{t}_i)$ if and only if $\mathsf{t}_i \rightarrowtriangle_{(\!|q|\!)} \mathsf{t}_j$.

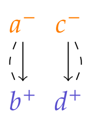
Proof. Since $\leq_{(\!|q|\!)}$ is a forestial order, $\partial_q(\mathsf{t})$ is a pointing string. It is alternating by definition, and the pointers follows $\rightarrowtriangle_{(\!|q|\!)}$, which follows $\rightarrowtriangle_{\mathsf{A}}$. Legality is ensured by minimality-preservation of $(\!|q|\!)$, and positivity by +-coveredness of q. Finally, visibility is a consequence of Lemma 3.22: considering the prefixes $\mathsf{t}^i = \mathsf{t}_1 \dots \mathsf{t}_i \sqsubseteq \mathsf{t}$, we can inductively prove that for all $i \leq n$, $\ulcorner \partial_q(\mathsf{t}^i) \urcorner = \partial_q([\mathsf{t}_i]_q)$. □

This allows us to consider the set of plays described by an augmentation.

Definition 3.24 – Plays of an augmentation

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$. Then we define

$$\mathsf{Plays}(q) = \{\partial_q(\mathsf{t}) \mid \mathsf{t} \in \mathsf{Alt}(q)\}\,.$$



$\partial_q(a) = \partial_q(c) = \mathbf{q}$
$\partial_q(b) = \mathsf{T}$
$\partial_q(d) = \mathsf{F}$

Figure 3.16: $q \in \mathsf{Aug}(\mathsf{bool})$.

Consider the augmentation $q \in \mathsf{Aug}(\mathsf{bool})$ in Figure 3.16. Then

$$\mathsf{Alt}(q) = \{\, a\ b\ c\ d\ ,\ c\ d\ a\ b \,\}\,,$$

and we obtain the plays

$$\mathsf{Plays}(q) = \{\, \mathbf{q} - \mathsf{T} \;\; \mathbf{q} - \mathsf{F}\,, \quad \mathbf{q} - \mathsf{F} \;\; \mathbf{q} - \mathsf{T} \,\}\,.$$

This operation is stable under augmentation isomorphism.

Lemma 3.25 – Isomorphic augmentations have the same plays

Consider $q, p \in \mathsf{Aug}(\mathsf{A})$. Then

$$q \cong p \Rightarrow \mathsf{Plays}(q) = \mathsf{Plays}(p)\,.$$

Proof. Consider $\mathsf{t} = \mathsf{t}_1 \dots \mathsf{t}_n \in \mathsf{Alt}(q)$ and the isomorphism $\varphi\colon q \cong p$. Then it is clear that $\varphi(\mathsf{t}_1)\dots\varphi(\mathsf{t}_n)$ is an alternating linearisation of p, and that

$$\partial_p(\varphi(\mathsf{t}_1)\dots\varphi(\mathsf{t}_n)) = \partial_q(\mathsf{t})\,,$$

so $\partial_q(\mathsf{t}) \in \mathsf{Plays}(p)$, and we get $\mathsf{Plays}(q) \subseteq \mathsf{Plays}(p)$. Since φ is an isomorphism, we also have $\mathsf{Plays}(p) \subseteq \mathsf{Plays}(q)$. □

This allows us to consider, for any isogmentation $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$, its plays $\mathsf{Plays}(\mathsf{q})$, which are $\mathsf{Plays}(q)$ for any $q \in \mathsf{q}$.

Now, we want those plays to be equivalent up to Mellies' homotopy relation.

Lemma 3.26 – Plays of an augmentation are homotopic

Consider $q \in \mathsf{Aug}(\mathsf{A})$ and $s, s' \in \mathsf{Plays}(q)$.

Then $s \sim_E s'$.

Proof. Consider $\mathsf{t}, \mathsf{t}' \in \mathsf{Alt}(q)$ such that $s = \partial_q(\mathsf{t})$ and $s' = \partial_q(\mathsf{t}')$. Writing $\mathsf{t} = \mathsf{t}_1 \dots \mathsf{t}_n$ and $\mathsf{t}' = \mathsf{t}'_1 \dots \mathsf{t}'_n$, let k be the first index such that $\mathsf{t}_k \neq \mathsf{t}'_k$ (assuming t and t' are different, otherwise the result is trivial). We show the equivalence inductively.

First, remark that t_k (and t'_k) must be negative by Lemma 3.22 and determinism of q. Actually, since alternating linearisations preserve immediate causality from negative to positive events, we must have $\mathsf{t} = \mathsf{u}\; a_1^-\; b_1^+\; \mathsf{v}\; a_2^-\; b_2^+\; \mathsf{w}$ and $\mathsf{t}' = \mathsf{u}\; a_2^-\; b_2^+\; \mathsf{v}'\; a_1^-\; b_1^+\; \mathsf{w}'$. But then,

$$s \;\sim_E\; \partial_q\left(\mathsf{u}\; a_2^-\; b_2^+\; a_1^-\; b_1^+\; \mathsf{v}\; \mathsf{w}\right) \;\sim_E\; s'$$

by definition for the first equivalence, and induction hypothesis for the second one. □

For any isogmentation $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$, we define $\mathbf{Plays}(\mathsf{q}) \in \mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E}$ as $\mathsf{Plays}(\mathsf{q})_{/\sim_E}$. Now we have the other side of the bijection from **Claim 1**:

$$\chi^{-1}\colon \mathsf{q} \in \mathsf{Isog}(\mathsf{A}) \quad\mapsto\quad \mathbf{Plays}(\mathsf{q}) \in \mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E}\,.$$

It remains to show that χ is indeed a bijection, whose inverse is as described above.

3.4.4 χ is a bijection

Theorem 3.27 – Isogmentations are plays up to $\sim_E$

There exists a bijection $\chi : \mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E} \cong \mathsf{Isog}(\mathsf{A})$.

Proof. We state that χ and its inverse are:

$$\begin{array}{rcl} \mathbf{s} & \mapsto & \mathbf{isog}(\mathbf{s}) \\ \mathbf{Plays}(\mathsf{q}) & \leftarrow\!\shortmid & \mathsf{q}\,. \end{array}$$

First, consider $\mathbf{s} \in \mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E}$. We want to prove that

$$\mathbf{Plays}(\mathbf{isog}(\mathbf{s})) = \mathbf{s}\,. \tag{3.1}$$

Consider $s \in \mathbf{s}$. Writing $s = s_1 \dots s_n$, it is clear that $1 \dots n$ is an alternating linearisation of $q = \mathsf{aug}(s)$, and $s = \partial_q(1 \dots n)$ by definition. Hence, $s \in \mathsf{Plays}(q)$. Since isomorphic augmentations have the same set of plays (Lemma 3.25), we have $s \in \mathsf{Plays}(\mathbf{isog}(\mathbf{s}))$. So we have $\mathbf{s} \subseteq \mathbf{Plays}(\mathbf{isog}(\mathbf{s}))$. Moreover, plays constructed from an augmentation are homotopic (Lemma 3.26), so we also have $\mathbf{Plays}(\mathbf{isog}(\mathbf{s})) \subseteq \mathbf{s}$, which gives us Equation (3.1).

Now, for the other side, consider $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$. We want:

$$\mathbf{isog}(\mathbf{Plays}(\mathsf{q})) = \mathsf{q}\,. \tag{3.2}$$

Consider $q \in \mathsf{q}$ and $s \in \mathsf{Plays}(q)$, with $\mathsf{t} \in \mathsf{Alt}(q)$ such that $\partial_q(\mathsf{t}) = s$. Then

$$|q| = \{\mathsf{t}_1, \dots, \mathsf{t}_n\} \quad \text{and} \quad |\mathsf{aug}(s)| = \{1, \dots, n\}\,,$$

and by definitions and Lemma 3.22, we have an isomorphism $\varphi : \mathsf{t}_i \mapsto i$. Hence,

$$q \cong \mathsf{aug}(s)\,. \tag{3.3}$$

So we have

$$\begin{array}{rll} \mathbf{isog}(\mathbf{Plays}(\mathsf{q})) & = \mathbf{isog}(\mathsf{Plays}(\mathsf{q})_{/\sim_E}) & \text{(by Lemma 3.26)} \\ & = \mathbf{isog}(\mathsf{Plays}(q)_{/\sim_E}) & \text{(by Lemma 3.25)} \\ & = \overline{\mathsf{aug}(s)} & \text{(by Lemma 3.20)} \\ & = \mathsf{q} & \text{(by (3.3))} \end{array}$$

which proves (3.2). □

$$\mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E} \;\underset{\mathbf{Plays}(-)}{\overset{\mathbf{isog}(-)}{\rightleftarrows}}\; \mathsf{Isog}(\mathsf{A})$$

Figure 3.17: Correspondence between HO and PCG, part 1.

Hence, we have an isomorphism between isogmentations and visible plays quotiented by Mellies' homotopy equivalence.

This allows us to interpret innocent strategies in HO games as sets of isogmentations in PCG (applying χ to the set of visible plays of the strategy – innocence ensures this set is stable by homotopy). However, sets of isogmentations in general do not translate to innocent strategies in HO: we obtain a set of plays, but innocent strategies need additional conditions such as prefix-closure, determinism, and so on. We now focus on characterizing those sets of isogmentations which do translate to innocent strategies in HO.

3.5 Meagre Innocent Strategies in PCG

In traditional HO games, innocent strategies can be characterized both by their P-views (the "meagre" version of the strategy) or their plays (the "fat" innocent strategy). Likewise, in PCG, innocent strategies have a meagre representation (a single isogmentation informing us on all the P-views) and a fat representation (a set of isogmentations corresponding to all possible plays). Before characterizing sets of isogmentations corresponding to innocent strategies, we focus on the meagre representation.

3.5.1 Meagre Innocent Augmentations and Isogmentations

Innocent strategies in HO games are characterized by the fact that Player does not change their behavior according to the number of duplications of Opponent moves: they always react in the same way. Therefore, all the information about the strategy is contained in its P-views, in which there is no duplication of Opponent moves. So an innocent strategy can be characterized by a unique augmentation without duplication of negative events, corresponding to the tree of its maximal P-views.

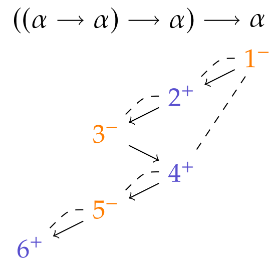


Figure 3.18: A mia q.

Reminder: For an innocent strategy σ, the set $\ulcorner\sigma\urcorner$ is the set of its P-views:

$$\ulcorner\sigma\urcorner = \{\ulcorner s \urcorner \mid s \in \sigma\}.$$

Definition 3.28 – Meagre Innocent Augmentation (mia)

A **meagre innocent augmentation** $q \in$ **MIA**(A) is a $-$-linear augmentation $q \in$ Aug(A).

Figure 3.18 features an example of a mia. Since $-$-linearity is stable by isomorphism, we can also define meagre innocent *isogmentations*:

Definition 3.29 – Meagre Innocent Isogmentation (mii)

A **meagre innocent isogmentation** q $\in$ **MII**(A) is a $-$-linear isogmentation q $\in$ Isog(A).

Thanks to the previous isomorphism, any mii can be translated into a play (up to $\sim_E$). We show that mii's correspond exactly to "trees of the maximal P-views of innocent (finite) strategies".

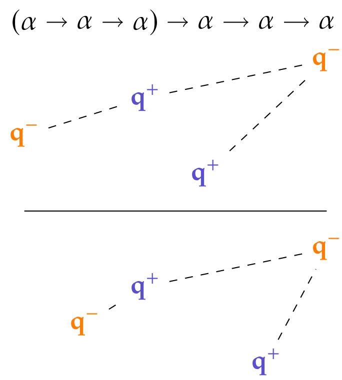


Figure 3.19: Maximal P-views of $[\![M]\!]_{\mathsf{HO}}$

Claim 2: There is a bijection **MII**(A) $\cong$ $\mathsf{HO}_f^{\mathsf{Inn}}$(A).

3.5.2 From innocent strategies to mii's

Definition 3.30 – MIA of a strategy

Consider a finite innocent strategy σ: A in HO. We construct the augmentation MIA(σ) with:

$$\begin{array}{rcl} |\mathsf{MIA}(\sigma)| & = & \{t \mid t \sqsubseteq s \wedge s \in \ulcorner\sigma\urcorner \wedge t \neq \epsilon\}, \\ s\,\mathsf{a} \leq_{(\mathsf{MIA}(\sigma))} s\,\mathsf{a}\,t\,\mathsf{b} & \text{iff} & \text{there is a chain of justifiers from b to a}, \\ s \leq_{\mathsf{MIA}(\sigma)} t & \text{iff} & s \sqsubseteq t, \\ \partial_{\mathsf{MIA}(\sigma)}(s\,\mathsf{a}) & = & \mathsf{a} \end{array}$$

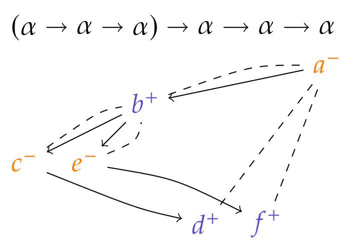


Figure 3.20: MIA($[\![M]\!]_{\mathsf{HO}}$)

Consider for instance the term

$$M = \lambda f^{\alpha\to\alpha\to\alpha}.\, \lambda x^{\alpha}.\, \lambda y^{\alpha}.\, f\ x\ y$$

and its interpretation $[\![M]\!]_{\mathsf{HO}}$ the innocent strategy defined by the two maximal P-views presented in Figure 3.19. Then $\mathsf{MIA}([\![M]\!]_{\mathsf{HO}})$ is the augmentation in Figure 3.20, with the following events:

$$\begin{aligned} a &= \mathbf{q}_6, & b &= \mathbf{q}_6\ \mathbf{q}_3, \\ c &= \mathbf{q}_6\ \mathbf{q}_3\ \mathbf{q}_1, & d &= \mathbf{q}_6\ \mathbf{q}_3\ \mathbf{q}_1\ \mathbf{q}_4, \\ e &= \mathbf{q}_6\ \mathbf{q}_3\ \mathbf{q}_2, & f &= \mathbf{q}_6\ \mathbf{q}_3\ \mathbf{q}_2\ \mathbf{q}_5, \end{aligned}$$

indexing the arena as $(\mathbf{q}_1 \Rightarrow \mathbf{q}_2 \Rightarrow \mathbf{q}_3) \Rightarrow \mathbf{q}_4 \Rightarrow \mathbf{q}_5 \Rightarrow \mathbf{q}_6$ for clarity. Also for the sake of clarity, the pointers are not represented in the list above, but d and f are the plays from Figure 3.19, with

$$a \sqsubseteq b \sqsubseteq c \sqsubseteq d \qquad \text{and} \qquad a \sqsubseteq b \sqsubseteq e \sqsubseteq f\,.$$

We check that this construction always define a mia.

Proposition 3.31 – MIA of a strategy

Consider a finite innocent strategy $\sigma : \mathsf{A}$. Then

$$\mathsf{MIA}(\sigma) = \langle |\mathsf{MIA}(\sigma)|, \leq_{(\!|\mathsf{MIA}(\sigma)|\!)}, \leq_{\mathsf{MIA}(\sigma)}, \partial_{\mathsf{MIA}(\sigma)} \rangle$$

is an augmentation $\mathsf{MIA}(\sigma) \in \mathbf{MIA}(\mathsf{A})$.

Moreover, $\mathsf{MIA}(\sigma)$ is total if and only if σ is total.

Proof. We write $q = \mathsf{MIA}(\sigma)$.

First, we check that $(\!|q|\!) = \langle |q|, \leq_{(\!|q|\!)}, \partial_q \rangle$ is a configuration.

Finite forest. Since σ is finite, so is $|q|$. It is clear from definition that $\langle |q|, \leq_{(\!|q|\!)} \rangle$ is a finite forest.

Minimality-respect. Clear from definition and legality of plays.

Causality-preservation. Consider $s, t \in |q|$ such that $s \rightarrowtriangle_{(\!|q|\!)} t$. Then by definition $s = s'\ \mathsf{a}$ and $t = t'\ \mathsf{a}\ t''\ \mathsf{b}$ with b pointing to a in t. By rigidity of plays, $\partial_q(s) \rightarrowtriangle_{\mathsf{A}} \partial_q(t)$.

Hence, $(\!|q|\!) \in \mathsf{Conf}(\mathsf{A})$. We now check q is an augmentation.

Forestiality. By definition of the prefix order, it is clear that $\langle |q|, \leq_q \rangle$ is a forest.

Rule-abidingness. Consider $s, t \in |q|$ such that $s \leq_{(\!|q|\!)} t$. By definition, s is a prefix of t, so $s \leq_q t$.

Courtesy. Consider $s \rightarrowtriangle_q t$ such that $\mathsf{pol}(s) = +$ or $\mathsf{pol}(t) = -$. By definition of ∂_q and $\rightarrowtriangle_q$, we have $s = s'\ \mathsf{a}^+$ and $t = s\ \mathsf{b}^-$ (plays are alternating). But t is a P-view, so b^- points to a^+ and $s \rightarrowtriangle_{(\!|q|\!)} t$.

Determinism. Consider $s^- \rightarrowtriangle_q t$ and $s^- \rightarrowtriangle_q t'$. We must have $t = s\ \mathsf{a}^+$ and $t' = s\ \mathsf{b}^+$, and by determinism of σ, $t = t'$.

Negativity. By negativity of A.

***+*-coveredness.** Strategies are sets of positive plays, so all maximal events of $|q|$ are positive.

Hence, $q \in \mathsf{Aug}(\mathsf{A})$. Finally, we check the additional conditions.

***−*-linearity.** Immediate for minimal events. If $s^+ \rightarrow_q s\ \mathsf{a}^-$ and $s^+ \rightarrow_q s\ \mathsf{b}^-$ with $\partial_q(s\ \mathsf{a}) = \partial_q(s\ \mathsf{b})$, then $\mathsf{a} = \mathsf{b}$ and both points to the last move of s by courtesy, so $s\ \mathsf{a} = s\ \mathsf{b}$.

Hence, $q \in \mathbf{MIA}(\mathsf{A})$.

Totality. Immediate by definition: the condition of totality for σ matches exactly the condition of totality for q.

Hence, q is total iff σ is total. □

This give us our first representation of strategies as isogmentations.

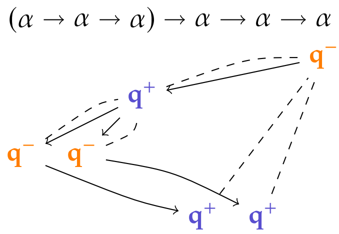


Figure 3.21: $\mathsf{MII}([\![M]\!]_{\mathsf{HO}})$

Definition 3.32 – MII of a strategy

Consider a finite innocent strategy $\sigma : \mathsf{A}$. We define

$$\mathsf{MII}(\sigma) = \overline{\mathsf{MIA}(\sigma)}$$

the isomorphism class of $\mathsf{MIA}(\sigma)$. Then $\mathsf{MII}(\sigma) \in \mathbf{MII}(\mathsf{A})$.

Figure 3.21 features the isogmentation corresponding to our earlier example of a mia (Figure 3.20).

We now have the first half of the isomorphism from **Claim 2**:

$$\mathsf{MII} : \sigma \in \mathsf{HO}_f^{\mathsf{Inn}}(\mathsf{A}) \mapsto \mathsf{MII}(\sigma) \in \mathbf{MII}(\mathsf{A})\,.$$

3.5.3 From mii's to innocent strategies

Likewise, a mii corresponds to an innocent strategy in HO games, whose P-views are constructed from the *branches* of the isogmentation.

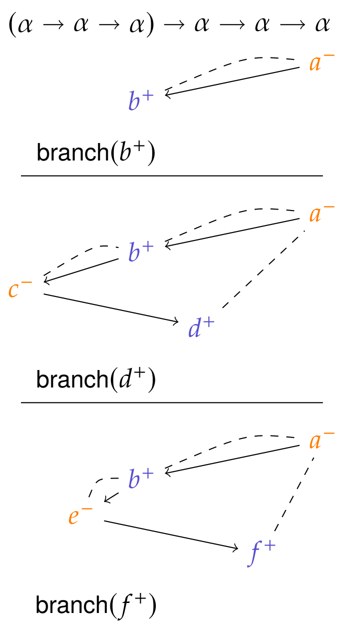


Figure 3.22: $\mathsf{Branches}(\mathsf{MIA}([\![M]\!]_{\mathsf{HO}}))$

Definition 3.33 – Branches

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$, and an event $a^+ \in |q|$.

Then we define $\mathsf{branch}(a)$ as the augmentation $\mathscr{b}$ with:

$$\begin{array}{rcl} |\mathscr{b}| & = & [a]_{\leq_q} \text{ the predecessors of } a \text{ for } \leq_q, \\ c \rightarrow_{(\!|\mathscr{b}|\!)} d & \text{iff} & c \rightarrow_{(\!|q|\!)} d\,, \\ c \rightarrow_{\mathscr{b}} d & \text{iff} & c \rightarrow_q d\,, \\ \partial_{\mathscr{b}}(c) & = & \partial_q(c)\,. \end{array}$$

The set of **branches** of q is $\mathsf{Branches}(q)$.

Looking back at the mia from Figure 3.20, the positive events of q define three branches, shown in Figure 3.22.

It is easy to check that branches are augmentations (in particular, the static order still has the correct properties thanks to rule-abidingness).

Lemma 3.34 – Branches are augmentations

Consider an augmentation $q \in \mathsf{Aug}(A)$. Any branch $\mathscr{b} \in \mathsf{Branches}(q)$ is an augmentation $\mathscr{b} \in \mathsf{Aug}(A)$.

Proof. By rule-abiding of q, we have:

$$\forall a^+ \in |q|, \quad [a]_{\leq_{(q)}} \subseteq [a]_{\leq_q} ,$$

which ensures $\mathsf{branch}(a)$ verifies all necessary conditions. □

We say that an augmentation $q \in \mathsf{Aug}(A)$ **is a branch** if $q \in \mathsf{Branches}(q)$.

Since branches are just restrictions of augmentations, they are preserved by isomorphism.

Lemma 3.35 – Branches and isomorphisms

Consider two augmentations $q, p \in \mathsf{Aug}(A)$ with $\varphi : q \cong p$.

For any $\mathscr{b} \in \mathsf{Branches}(q)$, we have $\varphi(\mathscr{b}) \in \mathsf{Branches}(p)$.

Now, since events in a branch $\mathscr{b}$ are totally ordered by $\leq_{\mathscr{b}}$, there exists only one alternating linearisation of $\mathscr{b}$.

Lemma 3.36 – A branch defines a unique play

Consider an augmentation $q \in \mathsf{Aug}(A)$ and a branch $\mathscr{b} \in \mathsf{Branches}(q)$.

Then $\mathsf{Plays}(\mathscr{b})$ is a singleton, and we write

$$\mathsf{Plays}(\mathscr{b}) = \{\mathsf{play}(\mathscr{b})\} .$$

By courtesy, such a play is always a P-view.

Lemma 3.37 – Branches define P-views

Consider an augmentation $q \in \mathsf{Aug}(A)$ and a branch $\mathscr{b} \in \mathsf{Branches}(q)$.

Then $\mathsf{play}(\mathscr{b})$ is a P-view.

Proof. Every negative move (apart from the initial one) points to its predecessor thanks to courtesy of $\mathscr{b}$. □

The following lemma formalizes the intuition that branches of an augmentation correspond to P-views of a play.

Lemma 3.38 – P-view from a branch

Consider $q \in \mathsf{Aug}(\mathsf{A})$, and $s = s_1 \dots s_n \in \mathsf{Plays}(q)$ corresponding to the alternating linearisation $e_1 \dots e_n$.

Then for any $s_1 \dots s_i \sqsubseteq^+ s$ with $i = 1, \dots, n$,

$$\ulcorner s_1 \dots s_i \urcorner \text{ is defined and } \ulcorner s_1 \dots s_i \urcorner = \mathsf{play}(\mathsf{branch}(e_i))\,.$$

Proof. We prove the equality by induction on i (which also proves the existence).

First, remark that $e_{i-1}^- \rightarrowtriangle_q e_i^+$ by Lemma 3.22.

If s_{i-1} is initial, so is e_{i-1}, and:

$$\ulcorner s_1 \dots s_i \urcorner = s_{i-1}\, s_i \quad \text{and} \quad |\mathsf{branch}(e_i)| = \{e_{i-1}, e_i\}$$

and the equality is clear.

Otherwise, s_{i-1} points to some s_j by legality of s, and (if it's defined):

$$\ulcorner s_1 \dots s_i \urcorner = \ulcorner s_1 \dots s_j \urcorner\, s_{i-1}\, s_i\,.$$

By induction hypothesis, $\ulcorner s_1 \dots s_j \urcorner$ is defined and:

$$\ulcorner s_1 \dots s_j \urcorner = \mathsf{play}(\mathsf{branch}(e_j))\,.$$

But since s_{i-1}^- points to s_j^+, we have $e_j^+ \rightarrowtriangle_{(\!|q|\!)} e_{i-1}^-$, and by courtesy $e_j^+ \rightarrowtriangle_q e_{i-1}^-$. So, we have:

$$e_j \rightarrowtriangle_q e_{i-1} \rightarrowtriangle_q e_i\,,$$

and by rigidity $\mathsf{just}(e_i) \in \mathsf{branch}(e_j)$. It is clear that $\mathsf{branch}(e_i)$ is $\mathsf{branch}(e_j)$ "extended" with e_{i-1} and e_i, and:

$$\mathsf{play}(\mathsf{branch}(e_i)) = \mathsf{play}(\mathsf{branch}(e_j))\, s_{i-1}\, s_i\,,$$

which gives us the desired equality. □

Now we can construct the reverse part of the isomorphism from **Claim 2**.

Proposition 3.39

Consider $\mathsf{q} \in \mathbf{MII}(\mathsf{A})$. Then we construct an innocent strategy $\mathsf{HOstrat}(\mathsf{q})$ whose P-views are:

$$\{\mathsf{play}(\mathsf{branch}(e)) \mid e \in |\underline{\mathsf{q}}|^+\} \cup \{\varepsilon\}\,.$$

Moreover, $\mathsf{HOstrat}(\mathsf{q})$ is total if and only if q is total.

Proof. All those plays are P-views by courtesy of q (Lemma 3.37). Since $\varepsilon \in \ulcorner\!\ulcorner \mathsf{HOstrat}(\mathsf{q}) \urcorner\!\urcorner$, the strategy is non-empty. It is prefix-closed by Lemma 3.38. Finally, all of these P-views are compatible by −-linearity and determinism of q. Hence, $\mathsf{HOstrat}(\mathsf{q})$ is an innocent strategy. Moreover, both definitions of totality coincide. □

Remark that HOstrat(q) does not depend on the choice of representative.

3.5.4 The isomorphism

Finally we check MII and HOstrat are inverses.

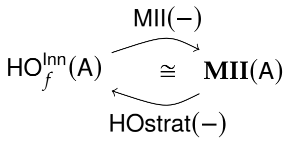


Figure 3.23: Correspondence between HO and PCG, part 2.

Theorem 3.40

Consider a negative arena A, then there exists a bijection

$$\mathsf{MII}\colon \mathsf{HO}_f^{\mathsf{Inn}}(A) \cong \mathbf{MII}(A)\,.$$

Moreover, MII preserves totality.

Proof. Consider $\sigma : A$ innocent and $\mathsf{q} \in \mathbf{MII}(A)$. We show:

$$\mathsf{HOstrat}(\mathsf{MII}(\sigma)) = \sigma \qquad \text{and} \qquad \mathsf{MII}(\mathsf{HOstrat}(\mathsf{q})) = \mathsf{q}\,.$$

For the first equality, consider a non-empty play $s \in \ulcorner\!\ulcorner \sigma \urcorner\!\urcorner$. Then $s \in |\mathsf{MII}(\sigma)|$, and by construction it is clear that $\mathsf{play}(\mathsf{branch}(s)) = s$, so $s \in \ulcorner\!\ulcorner \mathsf{HOstrat}(\mathsf{MII}(\sigma)) \urcorner\!\urcorner$.

Conversely, consider $s \in \ulcorner\!\ulcorner \mathsf{HOstrat}(\mathsf{MII}(\sigma)) \urcorner\!\urcorner$ with $s \neq \varepsilon$, then $s = \mathsf{play}(\mathscr{b})$ for some $\mathscr{b} \in \mathsf{Branches}(\mathsf{MIA}(\sigma))$. By construction, $s \in \ulcorner\!\ulcorner \sigma \urcorner\!\urcorner$. Hence $\ulcorner\!\ulcorner \sigma \urcorner\!\urcorner = \ulcorner\!\ulcorner \mathsf{HOstrat}(\mathsf{MII}(\sigma)) \urcorner\!\urcorner$.

Likewise, we can show that MII(HOstrat(q) = q by constructing:

$$\underline{q} \cong \mathsf{MIA}(\mathsf{HOstrat}(q))\,.$$

Finally, both constructions preserve totality. □

We now have the second part of our correspondence between HO and PCG (Figure 3.23); there is an isomorphism between isogmentations in PCG and (homotopy equivalence classes of positive visible) plays in HO, and there is another isomorphism between meagre innocent isogmentations in PCG and finite innocent strategies in HO. Can these two isomorphisms be shown to agree, and is there a way to deduce, for example, the isogmentations obtained from an innocent strategy σ *via* **isog**(−) if we only know MII(σ)? We show in the next subsection how these isogmentations can be constructed with the notion of *expansions*.

3.6 Fat Innocent Strategies in PCG

3.6.1 Expansions

Besides including meagre representations of innocent strategies, augmentations can also represent their *expansions*, *i.e.* arbitrary plays (with Opponent's scheduling factored out).

Definition 3.41 – Expansion

Consider an arena A and $p \in \mathbf{MIA}(\mathsf{A})$. An **expansion** of p is an augmentation $q \in \mathsf{Aug}(\mathsf{A})$ such that:

simulation: there is a (necessarily unique) morphism $\varphi : q \to p$.

We write $\exp(p)$ the set of expansions of p.

The relationship between a mia p and one of its expansions $q \in \exp(p)$ is analogous to that between an arena A and a configuration $x \in \mathsf{Conf}(\mathsf{A})$: q explores a prefix of p, possibly visiting the same branch many times. However, *determinism* ensures that only Opponent may cause duplications, and +-*coveredness* ensures that only Opponent may refuse to explore certain branches – if a Player move is available in p, then it must appear in all corresponding branches of q.

Recall the mia p from Figure 3.20. Then $\exp(p)$ includes for instance the augmentation q from Figure 3.24, where Opponent chooses to duplicate the event c^- and refuses to explore e^-.

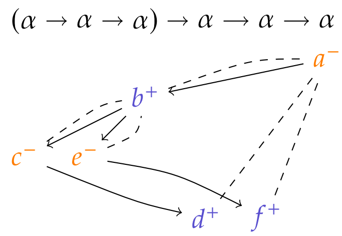


Figure 3.20: $p \in \mathbf{MIA}(\mathsf{A})$.

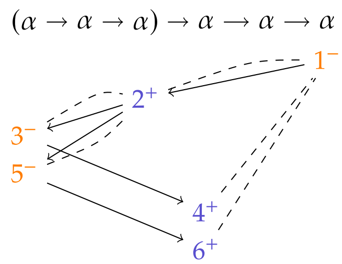


Figure 3.24: $q \in \exp(p)$, with:

$$\varphi : \begin{cases} 1 \mapsto a, & 4 \mapsto d, \\ 2 \mapsto b, & 5 \mapsto c, \\ 3 \mapsto c, & 6 \mapsto d. \end{cases}$$

Uniqueness of the morphism follows from −-linearity and determinism.

Lemma 3.42 – Unicity of morphism for expansions of mia's

Consider $p \in \mathbf{MIA}(\mathsf{A})$ and $q \in \exp(p)$.

Then there exists a unique morphism $\varphi : q \to p$.

Proof. The existence is given by the definition of $q \in \exp(p)$.

Assume there exist two morphisms $\varphi, \psi : q \to p$. Consider a minimal (for $\leq_q$) $a \in |q|$ such that $\varphi(a) \neq \psi(a)$.

If a is minimal in q, then $\partial_q(a)$ is minimal in A. By causality-preserving, we also have $\varphi(a), \psi(a)$ minimal for $\leq_p$, and by arena-preserving we have $\partial_p(\varphi(a)) = \partial_p(\psi(a))$, so by −-linearity of p, $\varphi(a) = \psi(a)$.

Therefore, a has an predecessor $b = \mathsf{pred}(a)$. By hypothesis, $\varphi(b) = \psi(b)$, hence by causality-preservation of morphisms, we have $\varphi(b) \rightarrowtriangle_p \varphi(a)$ and $\varphi(b) \rightarrowtriangle_p \psi(a)$.

If a is positive, then b must be negative, and by determinism $\varphi(a) = \psi(a)$, contradiction. If a is negative, then b must be positive. Moreover, by arena-preservation of morphisms,

$$\partial_p(\varphi(a)) = \partial_q(a) = \partial_p(\psi(a))\,.$$

By −-linearity of p, $\varphi(a) = \psi(a)$, contradiction. □

Moreover, remark that by definition, two isomorphic augmentations have the same expansions. This allows us to lift the definition of expansions to *isoexpansions*.

Reminder: For any $p \in \mathsf{Aug}(A)$, we write $\overline{p} \in \mathsf{Isog}(A)$ for the isomorphism class of p. For any $\mathsf{q} \in \mathsf{Isog}(A)$, we set $\underline{\mathsf{q}} \in \mathsf{Aug}(A)$ a representative of q.

Definition 3.43 – Isoexpansion

Consider $\mathsf{q} \in \mathbf{MII}(A)$. Then we define the **isoexpansions** of q as:

$$\mathsf{iexp}(\mathsf{q}) = \{\overline{p} \mid p \in \exp(\underline{\mathsf{q}})\} .$$

3.6.2 Fat Innocent (Iso)expansions

In HO games, plays of an innocent strategy are determined by the P-views of the corresponding meagre innocent strategy. Likewise, in PCG, isogmentations in an innocent strategy are *expansions* of a meagre innocent augmentation.

Definition 3.44 – Fat Innocent Expansion

Consider $q \in \mathbf{MIA}(A)$. Then we say $\exp(q)$ is the **fat innocent expansion** of q, noted $\exp(q) \in \mathbf{FIE}(A)$.

We can obviously lift this definition to isogmentations.

Definition 3.45 – Fat Innocent Isoexpansion

Consider $\mathsf{q} \in \mathbf{MII}(A)$. Then we say $\mathsf{iexp}(\mathsf{q})$ is the **fat innocent isoexpansion** of q, noted $\mathsf{iexp}(\mathsf{q}) \in \mathbf{FII}(A)$.

This alternative presentation of innocent strategies is equivalent to using only the meagre isogmentation: constructing the fii of a mii is an injective operation.

Proposition 3.46 – Injectivity of $\mathsf{iexp}(-)$

Consider two fat innocent isoexpansions $\mathbf{f}, \mathbf{g} \in \mathbf{FII}(A)$. Then:

$$\mathbf{f} = \mathbf{g} \quad \text{if and only if} \quad \text{there exists } \mathsf{q} \in \mathbf{MII}(A),\ \mathbf{f} = \mathbf{g} = \mathsf{iexp}(\mathsf{q}) .$$

Proof. ***If.*** Immediate.

Only if. Assume $\mathbf{f} = \mathbf{g}$. By definition of $\mathbf{FII}(A)$, there exist isogmentations $\mathsf{q}, \mathsf{p} \in \mathbf{MII}(A)$ such that $\mathbf{f} = \mathsf{iexp}(\mathsf{q})$ and $\mathbf{g} = \mathsf{iexp}(\mathsf{p})$. We write q and p for the respective representants of q and p. By hypothesis, $\exp(q) = \exp(p)$; which means in particular that $q \in \exp(p)$ and $p \in \exp(q)$. By unicity of morphisms for expansions of mia's (Lemma 3.42), we obtain $q \cong p$. □

In other words, we obtain an isomorphism $\mathsf{iexp}\colon \mathbf{MII}(A) \cong \mathbf{FII}(A)$, which composes with the previous isomorphism for meagre innocent isogmentations:

$$\mathsf{iexp} \circ \mathsf{MII}\colon \mathsf{HO}_f^{\mathsf{Inn}}(A) \cong \mathbf{FII}(A) .$$

3.6.3 The isomorphisms isog(−) and iexp ∘ MII(−) coincide

We now have **isog**(−) an isomorphism between (positive visible) plays (quotiented by homotopy) and isogmentations on the one side, and iexp ∘ MII(−) an isomorphism between finite innocent strategies and some sets of isogmentations on the other. Since innocent strategies are entirely defined thanks to their P-views, and mii's are "trees of P-views", these two notions coincide:

Notation: If $\sigma \in \mathsf{HO}_f^{\mathsf{Inn}}(\mathsf{A})$, we define:

$$\mathsf{Vis}(\sigma) = \sigma \cap \mathsf{VisPlays}(\mathsf{A})$$

the set of visible plays of σ, and:

$$\mathbf{isog}(\sigma) = \{\mathbf{isog}(s_{/\sim_E}) \mid s \in \mathsf{Vis}(\sigma)\}.$$

Proposition 3.47 – Compatibility of both isomorphisms

Consider $\sigma \in \mathsf{HO}_f^{\mathsf{Inn}}(\mathsf{A})$. Then:

$$\mathbf{isog}(\sigma) = \mathsf{iexp}(\mathsf{MII}(\sigma)).$$

Conversly, consider $p \in \mathbf{MII}(\mathsf{A})$. Then:

$$\mathsf{HOstrat}(p)_{/\sim_E} = \mathbf{Plays}(\mathsf{iexp}(p)).$$

In particular, this ensure that the positions of an innocent strategy in HO are the positions of its interpretation as a mia.

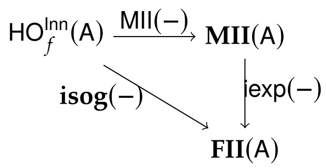


Figure 3.25: Correspondence between HO and PCG, part 3.

Proposition 3.48 – Positions in HO and PCG

Consider an innocent strategy $\sigma : \mathsf{A}$. Then $(\!|\sigma|\!) = (\!|\mathsf{MIA}(\sigma)|\!)$.

3.7 A few words on Infinite Strategies

Reminder: An innocent strategy is **infinite** if its set of P-views is infinite.

Until now we only considered finite objects, but *infinite* innocent strategies can also be represented in PCG. Obviously, innocent infinite strategies are still sets of finite plays, so our first traduction **isog**(−): HO ⟶ PCG is actually defined from innocent strategies to isogmentations. But what about the meagre representation? We still want a "tree of the P-views" but now we must represent infinite sets of P-views, so we extend our previous definitions of configurations and augmentations to infinite objects.

Definition 3.49 – ∞-configuration

An **∞-configuration** $x \in \mathsf{Conf}^\infty(\mathsf{A})$, is a tuple $x = \langle |x|, \leq_x, \partial_x \rangle$ such that $\langle |x|, \leq_x \rangle$ is a forest, and $\partial_x : |x| \longrightarrow |\mathsf{A}|$ is the display map with:

minimality-respecting: for any $a \in |x|$, a is $\leq_x$-minimal iff $\partial_x(a)$ is $\leq_\mathsf{A}$-minimal,

causality-preserving: $\forall a, b \in |x|$, if $a \rightarrow_x b$ then $\partial_x(a) \rightarrow_\mathsf{A} \partial_x(b)$.

If x has only one minimal event, we say that x is **well-opened**, noted $x \in \mathsf{Conf}_\bullet^\infty(\mathsf{A})$, and we note $\mathsf{init}(x)$ the minimal event.

As for (finite) configurations, an ∞-configuration can be seen as visiting a prefix of the arena, with possible reopenings. A polarity function for

x can be unambiguously deduced from the arena with, for any $a \in |x|$, $\mathsf{pol}_x(a) = \mathsf{pol}_{\mathsf{A}}(\partial_x(a))$.

(Iso)morphisms of ∞-configurations are defined just as in the finite case, as well as ∞-positions.

Definition 3.50 – ∞-augmentation

An ∞-**augmentation** q on an arena A, noted $q \in \mathsf{Aug}^\infty(\mathsf{A})$, is a tuple $q = \langle |q|, \leq_{(\!|q|\!)}, \leq_q, \partial_q \rangle$, where $(\!|q|\!) = \langle |q|, \leq_{(\!|q|\!)}, \partial_q \rangle \in \mathsf{Conf}^\infty(\mathsf{A})$, and $\langle |q|, \leq_q \rangle$ is an order satisfying:

finitary:	for all $a \in |q|$, $[a]_q = \{a' \in |q| \mid a' \leq_q a\}$ is finite,
forestial:	for all $a_1, a_2 \leq_q a$, then $a_1 \leq_q a_2$ or $a_2 \leq_q a_1$,
rule-abiding:	for all $a_1, a_2 \in |q|$, if $a_1 \leq_{(\!|q|\!)} a_2$, then $a_1 \leq_q a_2$,
courteous:	if $a \rightarrowtriangle_q b$ and $\mathsf{pol}(a) = +$ or $\mathsf{pol}(b) = -$, then $a \rightarrowtriangle_{(\!|q|\!)} b$,
deterministic:	for all $a^- \rightarrowtriangle_q b_1^+$ and $a^- \rightarrowtriangle_q b_2^+$, then $b_1 = b_2$,
negative:	for all $a \in |q|$ minimal for $\leq_q$, we have $\mathsf{pol}(a) = -$,
+-covered:	for all $a \in |q|$ maximal for $\leq_q$, we have $\mathsf{pol}(a) = +$.

We call $(\!|q|\!) \in \mathsf{Conf}^\infty(\mathsf{A})$ the **desequentialization** of q.

As before, we can extend the definition of (iso)morphisms and isogmentations. Isomorphism classes of ∞-augmentations of A will be called ∞-isogmentations, noted $\mathsf{Isog}^\infty(\mathsf{A})$.

Adapting the constructions from the previous section, we can extend the meagre representation to infinite innocent strategies. Given an innocent strategy σ, it can be translated either to an infinite isogmentation q, or to the set of all finite extensions of q.

3.8 Conclusion

We now have a model with:

- configurations / positions representing the “static” informations contained in a play,
- augmentations / isogmentations representing “trees of P-views”,
- a notion of expansion matching the construction of an innocent strategy from the set of its P-views,

along with isomorphisms between HO and PCG for innocent strategies, preserving finiteness and totality.

The next chapter present a first result obtained thanks to this model: innocent total finite strategies are positionaly injective.

The compositional aspect of the model will be studied in Chapter 6.

Positional Injectivity, for PCG and for HO

4

We now focus on positional injectivity, first for *finite total meagre innocent isogmentations* in PCG, then for *finite total innocent strategies* in HO.

Drawing inspiration from the proof of injectivity of the relational model for MELL proof nets [18], we construct **characteristic expansions** in Section 4.1 in such a way that we can track down duplications of negative events in the positions in order to recover a "sufficient" portion of the causal structure. We explain what we mean by sufficient in Section 4.2 with the introduction of several **bisimulation** relations, between events and between augmentations. Finally, we deduce positional injectivity for PCG in Section 4.3. Section 4.4 goes back to HO games and presents a counter-example for positional injectivity in the case of *infinite*, *partial* innocent strategies.

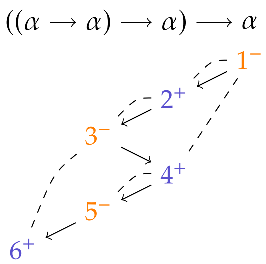
[18]: de Carvalho (2016), 'The Relational Model Is Injective for Multiplicative Exponential Linear Logic'

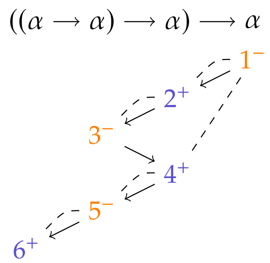
In all this chapter, A is a negative well-opened arena, unless stated otherwise.

4.1 Duplicating Opponent Moves

4.1.1 Proof idea

We already know that given a position $\mathsf{x} \in \mathsf{Pos}(\mathsf{A})$, we cannot in general uniquely reconstruct its causal explanation. Consider for instance q and p the mia's for K_x and K_y defined in Subsection 3.2.2:





$q = \mathsf{MIA}([\![K_x]\!]_{\mathsf{HO}})$, $p = \mathsf{MIA}([\![K_y]\!]_{\mathsf{HO}})$.

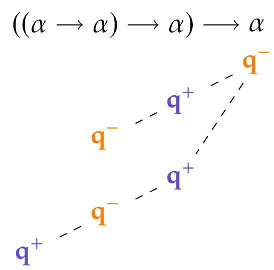
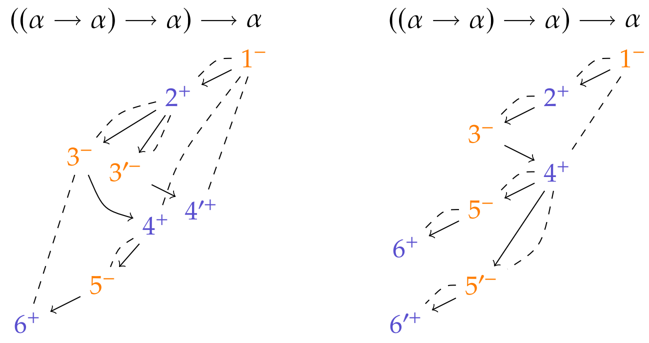


Figure 4.1: $(\!|q|\!) = (\!|p|\!)$.

Then both augmentations reach the same position (Figure 4.1): if we forget the causal order, we cannot distinguish between the two branches of the position anymore.

However, we already constructed two positions distinguishing between K_x and K_y in Subsection 3.2.2, by duplicating the Opponent move justifying the last Player move. This triggers two different reactions from the two augmentations: in q, 6^+ is justified by 3^-, so Player reacts to the duplication of 3^- with a copy of 4^+. However, in p, 6^+ is justified by 5^-, so Player reacts to the duplication of 5^- with a copy of 6^+.

We obtain the following expansions:



$q' \in \exp(q)$, $\qquad$ $p' \in \exp(p)$.

Reminder: for any $q \in \mathsf{Aug}(\mathsf{A})$, $(\!|q|\!)$ is the position $\overline{(\!|q|\!)}$ reached by the desequentialization of q. For any $p \in \mathbf{MIA}(\mathsf{A})$,

$$\begin{aligned} \exp_\bullet(p) &= \exp(p) \cap \mathsf{Aug}_\bullet(\mathsf{A}) \,, \\ (\!|p|\!) &= \{ (\!|q|\!) \mid q \in \exp_\bullet(p) \} \,. \end{aligned}$$

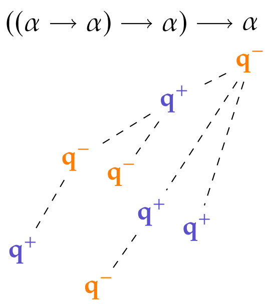


Figure 4.2: $\mathsf{x}' = (\!|q'|\!)$.

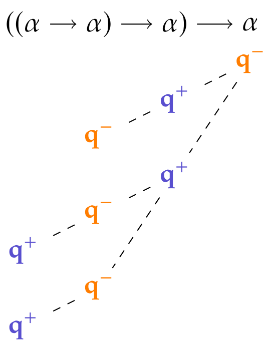


Figure 4.3: $\mathsf{y}' = (\!|p'|\!)$.

Then it is clear that $(\!|q'|\!) \neq (\!|p'|\!)$ (see Figures 4.2 and 4.3).

Furthermore, consider $\mathsf{y}' = (\!|p'|\!)$ the position represented in Figure 4.3. The only (up to iso) expansion of a mia yielding y' as a position is p': every other attempt to guess causal wiring fails, because of $-$-linearity and the cardinality of duplications. But p' is an expansion of the unique maximal branch of (the mia representing) K_y; which means that if we are given y' and the information that y' comes from a "maximal" expansion of a mia r (in the sense that it explores all branches at least once), then we know r can only be (the mia representing) K_y.

This suggests a proof idea: given $p_1 \in \mathbf{MIA}(\mathsf{A})$, we seek to construct an expansion $q_1 \in \exp(p_1)$ whose position would uniquely characterize p_1, in the sense that for any $p_2 \in \mathbf{MIA}(\mathsf{A})$ such that $(\!|p_1|\!) = (\!|p_2|\!)$, the fact that $(\!|q_1|\!) \in (\!|p_2|\!)$ implies that $p_1 \cong p_2$.

Such expansions will be called *characteristic expansions*; we give the definition in Subsection 4.1.2. Then in Section 4.2, we define bisimulations between augmentations, aiming to prove that

1. characteristic expansions reaching the same position are bisimilar (Section 4.3),
2. if two MIA have bisimilar characteristic expansions, they are actually equal (up to isomorphism) (Subsection 4.2.2).

4.1.2 Characteristic Expansions

Characteristic expansions are expansions with conditions on the cardinality of duplications of Opponent moves. Hence, we first need to define those sets of duplicated moves.

Definition 4.1 – Fork

Consider an augmentation $q \in \mathsf{Aug}(A)$. A **fork** in q is a maximal non-empty set $X \subseteq |q|$ such that:

negative:	for all $a \in X$, $\mathsf{pol}(a) = -$,
sibling:	either all $a \in X$ are minimal for $\leq_q$,
	or there exists $b \in |q|$ s.t. for all $a \in X$, $b \rightarrowtriangle_q a$,
identical:	for all $a, b \in X$, $\partial_q(a) = \partial_q(b)$.

We write $\mathsf{Fork}(q)$ for the set of forks in q.

For $p \in \mathbf{MIA}(A)$ and $q \in \exp(p)$, the forks of q are exactly the sets of duplicated Opponent moves. Recall the augmentation q from Figure 3.14 for instance; Figure 4.4 shows the three forks of q, where 5 and 7 belong in the same fork, as copies of the same Opponent move.

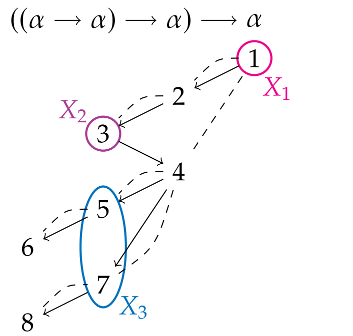


Figure 4.4: Forks of an augmentation q.

Moreover, the definition of forks only depends on causal links of the form $a^+ \rightarrowtriangle_q b^-$; and by courtesy, these are exactly the static causal links of the form $a^+ \rightarrowtriangle_{(\!|q|\!)} b^-$. Hence, forks are preserved by desequentialization and isomorphisms.

Lemma 4.2 – Forks of a configuration

Consider an augmentation $q \in \mathsf{Aug}(A)$. For any $p \in \mathsf{Aug}(A)$ such that $\varphi : (\!|q|\!) \cong (\!|p|\!)$, we have:

$$\forall X \subseteq |q|, X \in \mathsf{Fork}(q) \Leftrightarrow \varphi(X) \in \mathsf{Fork}(p).$$

Proof. By courtesy and the fact configuration morphisms preserves the static order and the arena image. □

Remark: In particular, Lemma 4.2 implies that for any two augmentations $q, p \in \mathsf{Aug}(A)$ such that $\varphi : q \cong p$, we have

$$\forall X \subseteq |q|, X \in \mathsf{Fork}(q) \Leftrightarrow \varphi(X) \in \mathsf{Fork}(p).$$

This allows us to consider $\mathsf{Fork}((\!|q|\!))$, where the fact that $X \in \mathsf{Fork}((\!|q|\!))$ can be deduced without knowing $\leq_q$. For instance, Figure 4.5 shows the forks of the configuration $(\!|q|\!)$, where q is the augmentation from Figure 4.4 – remark that both sets of forks coincide.

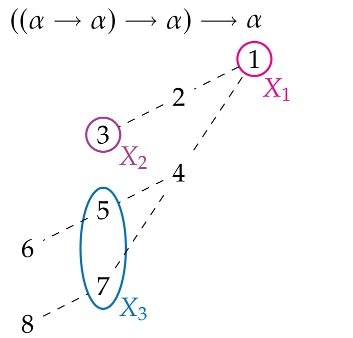


Figure 4.5: Forks of a configuration $(\!|q|\!)$.

Consider a fork X. Since augmentations are +-covered and by causality-preserving of morphisms and determinism of augmentations, all Player moves in q caused by Opponent moves in X are copies of the same Player move in p. So, if X has **cardinality** $\sharp X = n$ and we find exactly one set Y of "equivalent" Player moves of cardinality $\sharp Y = m \geq n$, we may deduce that the successors of the events of X are in Y. We will formalize what we mean by "equivalent" in Subsection 4.2.3; for now it suffices to think of those sets as sets of Player moves "behaving the same way" in the position (*e.g.* a minimal requirement would be that all moves of such a set have the same arena image). In Figure 4.4 for example, the fork X_3 has two elements 5 and 7. The set of their successors can only be $\{6, 8\}$: it cannot contain 4 by acyclicity of $\rightarrowtriangle_q$, and it cannot be $\{2, 6\}$ or $\{2, 8\}$ because 2 does not have the same arena image as the others. Hence, in this very simple case, we are able to deduce causal links thanks to the cardinality of forks.

In general though, distinct Opponent moves may trigger identical Player moves, so that the cardinality of a set Y of "similar" Player moves is the sum of the cardinalities of the predecessor forks. To allow us to identify these predecessor sets uniquely, the trick is to construct the characteristic expansion so that all forks have cardinality a distinct power of 2, making it so that the predecessor forks can be inferred from the (unique) binary decomposition of $\sharp Y$. This brings us to the following definition.

Definition 4.3 – Characteristic Expansion

Consider a MIA $p \in \mathsf{Aug}(\mathsf{A})$. A **characteristic expansion** of p is an augmentation $q \in \exp(p)$, with $\varphi : q \to p$, such that:

fork-injective:	for $X, Y \in \mathsf{Fork}(q)$, if $\sharp X = \sharp Y$ then $X = Y$,
well-powered:	for $X \in \mathsf{Fork}(q)$, there is $n \in \mathbb{N}$ such that $\sharp X = 2^n$,
−-obsessional:	for $a^+ \in \lvert q \rvert$, if $\varphi(a^+) \rightarrowtriangle_p b^-$, there is $a^+ \rightarrowtriangle_q a'$ such that $\varphi(a') = b^-$.

The condition −-obsessional means that q has at least one copy of every negative element of p; since augmentations are +-covered it also copies at least once every positive element of p. Hence, q is characteristic in the sense that it contains all the information given in p.

This definition is stable by isomorphism, allowing us to consider **characteristic iso-expansions**, as in Figure 4.6.

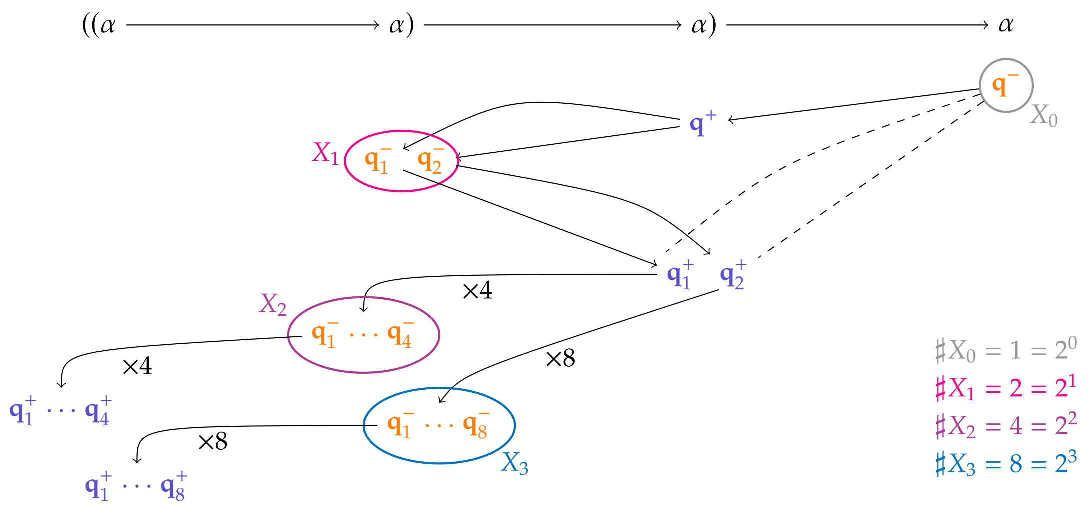


Figure 4.6: A characteristic (iso-)expansion q for the mia representing K_y, with four forks. We only write $\rightarrowtriangle_{(\!|q|\!)}$ when it differs from $\rightarrowtriangle_q$, and use indices to indicate the number of duplications.

Given a mia p and an augmentation $q \in \exp(p)$, is it possible to deduce from the position $(\!|q|\!)$ whether or not q is a characteristic expansion of p? The first two conditions, *fork-injective* and *well-powered*, only constrain the cardinality of forks, and $\mathsf{Fork}(q) = \mathsf{Fork}((\!|q|\!))$. However, we cannot say in general if q is *−-obsessional*: if an Opponent move (available in A) does not appear in q, is it because it never occurs in p, or because q forgot to copy it? Without knowing p it is impossible to conclude in general. However, −-obsessional expansions have the very interesting property of preserving totality: for any $p \in \mathbf{MIA}(\mathsf{A})$ and $q \in \exp(p)$ a characteristic expansion of p, q is total if and only if p is total.

Reminder: $q \in \mathsf{Aug}(\mathsf{A})$ is **total** if for any $a^+ \in \lvert q \rvert$, if there exists $b' \in \lvert \mathsf{A} \rvert$ such that $\partial_q(a) \rightarrowtriangle_{\mathsf{A}} b'$, then there exists $b \in \lvert q \rvert$ such that $\partial_q(b) = b'$ and $a \rightarrowtriangle_q b$.

Lemma 4.4 – Totality of characteristic expansions

Consider $p \in \mathbf{MIA}(\mathsf{A})$ and a characteristic expansion $q \in \exp(p)$.

Then q is total if and only if p is total.

Proof. Immediate by definition. □

Moreover, if we know that p is total, then $q \in \exp(p)$ is $-$-obsessional if and only if it is total – which means, by courtesy, that being $-$-obsessional *is* a property of $\llparenthesis q \rrparenthesis$ in that case. All in all, we get that for a *total* mia p, we *can* deduce if an expansion $q \in \exp(p)$ is characteristic only by looking at its position $\boldsymbol{(} q \boldsymbol{)}$.

Lemma 4.5 – Positions of characteristic expansions

Consider $p \in \mathbf{MIA}(\mathsf{A})$ a total augmentation, and $q \in \exp(p)$.

Then q is a characteristic expansion of p if and only if $\boldsymbol{(} q \boldsymbol{)}$ is fork-injective, well-powered and total.

Proof. We define fork-injectivity and well-poweredness for a position as fork-injectivity and well-poweredness for any of its representative. The result is trivial by courtesy and totality of p. □

Consider a well-opened[1] arena A and two total mia p_1 and p_2 such that $\boldsymbol{(} p_1 \boldsymbol{)} = \boldsymbol{(} p_2 \boldsymbol{)}$. Then for any q_1 a characteristic expansion of p_1, there exists $q_2 \in \exp(p_2)$ such that $\boldsymbol{(} q_1 \boldsymbol{)} = \boldsymbol{(} q_2 \boldsymbol{)}$, and by the above lemma q_2 is a characteristic expansion of p_2.

1: We need well-openedness because we only consider pointed positions in $\boldsymbol{(} - \boldsymbol{)}$; the results can be easily extended to a general negative arena A by decomposing A in a product of well-opened arenas $\mathsf{A}_1, \ldots, \mathsf{A}_n$. Any total $p \in \mathbf{MIA}(\mathsf{A})$ can be decomposed as a tuple of pointed mia $p_i \in \mathbf{MIA}(\mathsf{A}_i)$.

How different can be those two characteristic expansions $q_1 \in \exp(p_1)$ and $q_2 \in \exp(p_2)$? Since $\llparenthesis q_1 \rrparenthesis \cong \llparenthesis q_2 \rrparenthesis$, a first guess would be isomorphic; however that is not always true. Consider the following total mia p:

$$((\alpha \longrightarrow \alpha) \longrightarrow \alpha) \longrightarrow (\alpha \longrightarrow \alpha \longrightarrow \alpha) \longrightarrow \alpha$$

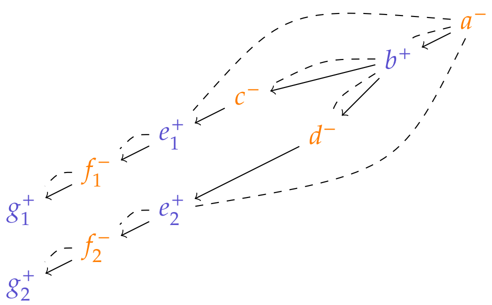


Figure 4.7: A mia p.

Since c^- and d^- trigger (copies of) the same events in p, we can construct several non-isomorphic characteristic expansions of p reaching the same position. For instance, if c^- and d^- are duplicated respectively 2 and 4 times, we obtain $2 + 4 = 6$ moves that are copies of e_1^+ or of e_2^+, with no way of recovering precisely if a copy corresponds to e_1^+ or e_2^+.

So, characteristic expansions have some degree of liberty in swapping forks around: they might have "the same branches" with different multiplicity. Hence, we need a weaker relation between q_1 and q_2. Thus we define *bisimulations* between augmentations, seeking to construct a relation that is both "weak enough" to allow such changes in multiplicity, and "strong enough" to ensure that $q_1 \sim q_2$ implies $p_1 \cong p_2$.

4.2 Bisimulation Relations

4.2.1 Bisimulations across an isomorphism

Let us first focus on bisimulations between augmentations reaching the same positions. Consider $q, p \in \mathsf{Aug}(A)$ such that $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$. Given $a \in |q|$ and $b \in |p|$, we need a predicate $a \sim b$ expressing that a and b have "the same causal follow-up, up to the multiplicity of Opponent duplications".

In particular, a and b must have "the same pointers" – but that cannot be strictly true since they live in different sets of events!
An idea that might first come to mind is to consider $a \sim^{\varphi} b$ parametrized by φ, asking that the pointers are equal *via* φ. But as the bisimulation unfolds, this requirement is too strong: as seen in the previous example, an isomorphism φ between desequentializations is not enough to ensure that all pointers match *via* φ.

So our actual predicate has form $a \sim^{\varphi}_{\Gamma} b$ where Γ is a *context* stating a correspondence between negative moves established in the bisimulation game so far.

Notation: For any morphism Γ, we write:

- $\mathsf{dom}(\Gamma)$ for its **domain**,
- $\mathsf{cod}(\Gamma)$ for its **codomain**,

i.e. $\Gamma\colon \mathsf{dom}(\Gamma) \to \mathsf{cod}(\Gamma)$.

Definition 4.6 – Context

Consider $q, p \in \mathsf{Aug}_{\bullet}(A)$ with $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$.

A **context** Γ between q and p is a bijection such that:

well-defined: $\mathsf{dom}(\Gamma) \subseteq |q|$ and $\mathsf{cod}(\Gamma) \subseteq |p|$,
negative: $\mathsf{pol}(\mathsf{dom}(\Gamma)) \subseteq \{-\}$,
arena-preserving: for all $a \in \mathsf{dom}(\Gamma)$, $\partial_q(a) = \partial_p(\Gamma(a))$.

In the *negative* condition, we ask for inclusion rather than equality to allow empty contexts. Remark that by arena-preservation, this is equivalent to ask $\mathsf{pol}(\mathsf{cod}(\Gamma)) \subseteq \{-\}$. This ensures that for any context Γ between augmentations q and p, then Γ^{-1} is a context between p and q.

We now give a first notion of bisimulation across augmentations.

Definition 4.7 – Bisimulation (between events)

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$ with $\varphi \colon (\!|q|\!) \cong (\!|p|\!)$.

For any $a \in |q|$, $b \in |p|$ and Γ a context, we say that a context **enables** a, b, noted $\Gamma \vdash (a, b)$, if:

(a) for all $a' \in |q|$, if $a' >_q a$ then $a' \notin \mathsf{dom}(\Gamma)$;
(b) for all $b' \in |p|$, if $b' >_p b$ then $b' \notin \mathsf{cod}(\Gamma)$.

We define a predicate:

$$a \textbf{ and } b \textbf{ are bisimilar via } \varphi \textbf{ with the context } \Gamma,$$

written $a \sim^{\varphi}_{\Gamma} b$, which holds if, firstly:

(1) $\partial_q(a) = \partial_p(b)$;
(2) $\Gamma \vdash (a, b)$.

If a is positive, we additionally require:

(3) if $\mathsf{just}(a) \in \mathsf{dom}(\Gamma)$,
then $\mathsf{just}(b) \in \mathsf{cod}(\Gamma)$ and $\Gamma(\mathsf{just}(a)) = \mathsf{just}(b)$;
(4) if $\mathsf{just}(a) \notin \mathsf{dom}(\Gamma)$,
then $\mathsf{just}(b) \notin \mathsf{cod}(\Gamma)$ and $\varphi(\mathsf{just}(a)) = \mathsf{just}(b)$.

Finally, the following two bisimulation conditions hold inductively:

(5) if $a^+ \rightarrowtriangle_q a'$, then there is $b' \in |p|$ such that $b \rightarrowtriangle_p b'$ and $a' \sim^{\varphi}_{\Gamma \cup \{(a', b')\}} b'$, and symmetrically;
(6) if $a^- \rightarrowtriangle_q a'$, then there is $b' \in |p|$ such that $b \rightarrowtriangle_p b'$ and $a' \sim^{\varphi}_{\Gamma} b'$, and symmetrically.

Reminder: $\mathsf{just}(a)$ is the (unique) $a' \in |q|$ such that $a' \rightarrowtriangle_{(\!|q|\!)} a$.

Remark: Regarding condition (5): $\Gamma \cup \{(a', b')\}$ remains a bijection since $\Gamma \vdash (a, b)$ implies that $a' \notin \mathsf{dom}(\Gamma)$ and $b' \notin \mathsf{cod}(\Gamma)$.

Of particular interest is the case $a \sim^{\varphi}_{\emptyset} b$ over an empty context, written simply $a \sim^{\varphi} b$. From this, we deduce a relation between augmentations.

Definition 4.8 – Bisimulation (between augmentations)

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$ with $\varphi \colon (\!|q|\!) \cong (\!|p|\!)$.

We say that q and p are **bisimilar via** φ, noted $q \sim^{\varphi} p$, if

$$\mathsf{init}(q) \sim^{\varphi} \mathsf{init}(p)\,.$$

Reminder: $\mathsf{init}(q)$ is the initial event of q, *i.e.* the event minimal for $\leq_q$.

Bisimulations allow us to express that two characteristic expansions with isomorphic configurations are "the same". Furthermore, they enjoy equivalence properties:

Lemma 4.9 – Equivalence properties of bisimulations

Consider $q, p, r \in \mathsf{Aug}_\bullet(\mathsf{A})$ with $\varphi \colon (\!|q|\!) \cong (\!|p|\!)$ and $\psi \colon (\!|p|\!) \cong (\!|r|\!)$.

For any events $a \in |q|$, $b \in |p|$, $c \in |r|$ and contexts Γ, Δ, we have:

reflexivity: $a \sim^{\mathsf{id}} a$,
symmetry: if $a \sim^{\varphi}_{\Gamma} b$ then $b \sim^{\varphi^{-1}}_{\Gamma^{-1}} a$,
transitivity: if $a \sim^{\varphi}_{\Gamma} b$ and $b \sim^{\psi}_{\Delta} c$ with $\mathsf{cod}(\Gamma) = \mathsf{dom}(\Delta)$, then $a \sim^{\psi \circ \varphi}_{\Delta \circ \Gamma} c$.

Proof. Immediate by induction. □

Recall the proof sketch for positional injectivity at the end of Subsection 4.1.1. Given two total mia $p_1, p_2 \in \mathsf{Aug}_\bullet(\mathsf{A})$ and two characteristic expansions $q_1 \in \exp_\bullet(p_1)$ and $q_2 \in \exp_\bullet(p_2)$ with $\varphi : (\!|q_1|\!) \cong (\!|q_2|\!)$, we want to prove:

1. if $\varphi : (\!|q_1|\!) \cong (\!|q_2|\!)$ then $q_1 \sim^\varphi q_2$,
2. if $q_1 \sim^\varphi q_2$, then $p_1 \cong p_2$.

We start by proving the second proposition.

4.2.2 Bisimulations between non-isomorphic augmentations

To achieve that, we exploit compositional properties of bisimulations. More precisely, we define bisimulations between augmentations over non-isomorphic configurations, and we show that $q_i \in \exp_\bullet(p_i)$ induces a bisimulation $q_i \sim p_i$. We then find a way to compose

$$p_1 \;\sim\; q_1 \;\sim^\varphi\; q_2 \;\sim\; p_2 \tag{4.1}$$

to deduce $p_1 \sim p_2$, which will imply $p_1 \cong p_2$.

Obviously we cannot expect there to be an isomorphism between $(\!|q_i|\!)$ and $(\!|p_i|\!)$, as characteristic expansions have by construction many more events. Hence we introduce a variant of Definition 4.7 – where we remove condition (4) and ask that all pointers go through Γ:

Definition 4.10 – Bisimulation (for non-iso augmentations)

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$. For any $a \in |q|$, $b \in |p|$ and a context Γ, we say that a **and** b **are bisimilar with the context** Γ, written $a \sim_\Gamma b$, if:

(1) $\partial_q(a) = \partial_p(b)$,
(2) $\Gamma \vdash (a, b)$.

If a is positive, we additionally require:

(3) $\mathsf{just}(a) \in \mathsf{dom}(\Gamma)$ and $\Gamma(\mathsf{just}(a)) = \mathsf{just}(b)$.

Finally, the following two bisimulation conditions hold inductively:

(4) if $a^+ \rightarrowtriangle_q a'$, then there is $b' \in |p|$ such that $b \rightarrowtriangle_p b'$ and $a' \sim_{\Gamma \cup \{(a',b')\}} b'$, and symmetrically,
(5) if $a^- \rightarrowtriangle_q a'$, then there is $b' \in |p|$ such that $b \rightarrowtriangle_p b'$ and $a' \sim_\Gamma b'$, and symmetrically.

We say that q **and** p **are bisimilar**, written $q \sim p$, if:

$$\mathsf{init}(q) \sim_{\{(\mathsf{init}(q), \mathsf{init}(p))\}} \mathsf{init}(p)\,.$$

It may seem confusing that we use the same notation for both kinds of bisimulations. This is justified by the fact that whenever both definitions apply, they coincide:

Lemma 4.11 – Both bisimulations coincide

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$, and $\varphi : (\!|q|\!) \cong (\!|p|\!)$.

Then $q \sim^{\varphi} p$ if and only if $q \sim p$.

Proof. ***If.*** Straightforward from Definition 4.7 and Definition 4.10: case (4) of Definition 4.7 is never used.

Only if. For $a \in |q|$, we define the **negative predecessors** of a in q:

$$[a]^-_q = \{a' \in |q| \mid a' \leq_q a \text{ and } \mathsf{pol}(a') = -\} .$$

Given $a \in |q|$, $b \in |p|$, $\Gamma \vdash (a, b)$, we say that Γ is **complete** if

$$[a]^-_q \subseteq \mathsf{dom}(\Gamma) \quad \text{and} \quad [b]^-_p \subseteq \mathsf{cod}(\Gamma) .$$

For all events $a \in |q|$, $b \in |p|$ and complete context $\Gamma \vdash (a, b)$, if $a \sim^{\varphi}_{\Gamma} b$ then $a \sim_{\Gamma} b$. The proof is immediate by induction: the clause (4) of Definition 4.7 is never used from the hypothesis that Γ is complete. We apply this to the roots of q, p:

$$\mathsf{init}(q) \sim_{\{(\mathsf{init}(q), \mathsf{init}(p))\}} \mathsf{init}(p) ,$$

which is exactly the definition of $q \sim p$. □

Remark: We still very much need to use the $\sim^{\varphi}$ bisimulation sometimes! All isomorphisms between q and p associate $\mathsf{init}(q)$ with $\mathsf{init}(p)$ (because both augmentations are pointed), but they can differ for other events. Expliciting φ will be necessary for some parts of the positional injectivity proof, since we compare all events of the augmentations and not only the roots.

This version of bisimulation also enjoys equivalence properties:

Lemma 4.12 – Equivalence for bisimulations w/o iso

Consider $q, p, r \in \mathsf{Aug}_\bullet(\mathsf{A})$.

For any events $a \in |q|$, $b \in |p|$, $c \in |r|$ and contexts Γ, Δ, we have:

reflexivity: $a \sim_{\mathsf{id}_{[a]^-_q}} a$,
symmetry: if $a \sim_\Gamma b$ then $b \sim_{\Gamma^{-1}} a$,
transitivity: if $a \sim_\Gamma b$ and $b \sim_\Delta c$ with $\mathsf{cod}(\Gamma) = \mathsf{dom}(\Delta)$, then $a \sim_{\Delta \circ \Gamma} c$.

Proof. Similar to the proof for Lemma 4.9. □

Finally this bisimulation allows us to express what we want: the fact that two augmentations are "the same, up to Opponent duplications". In particular, a pointed expansion q of a pointed mia p is bisimilar to p if and only if it is a $-$-obsessional expansion.

Reminder: $q \in \exp(p)$ with the morphism φ is a **$-$-obsessional** expansion if for all $a^+ \in |q|$, if $\varphi(a) \rightarrow_p b'$, then there exists $b \in |q|$ such that $a \rightarrow_q b$ and $\varphi(b) = b'$.

We first state that events of a $-$-obsessional expansion q and their images in p are bisimilar. First, recall that for any mia $p \in \mathsf{Aug}_\bullet(\mathsf{A})$ and $\varphi : q \rightarrow p$, for any event $a \in |q|$, we define the negative predecessors of a in q as:

$$[a]_q^- = \{a' \in |q| \mid a' \leq_q a \text{ and } \mathsf{pol}(a') = -\}\,.$$

From the definition of augmentation morphisms, there is an order-isomorphism:

$$\Gamma_a^\varphi : [a]_q^- \cong [\varphi(a)]_p^-\,.$$

Finally, we define the **co-depth** of a as the maximal length k of a causal chain $a = a_1 \rightarrowtriangle_q \ldots \rightarrowtriangle_q a_k$.

Lemma 4.13 – Bisimulation for −-obsessional expansions

Consider $p \in \mathbf{MIA}_\bullet(\mathsf{A})$ and $q \in \exp_\bullet(\mathsf{A})$ a −-obsessional expansion with the morphism $\varphi : q \to p$.

Then, for any $a \in q$,

$$a \sim_{\Gamma_a^\varphi} \varphi(a)\,.$$

Proof. By induction on the co-depth of $a \in |q|$. We check that $a \sim_{\Gamma_a^\varphi} \varphi(a)$, following Definition 4.10.

First, **(1)** and **(2)** are immediate by the definitions of φ and Γ_a^φ.

(3). If a is positive, then:

$$\begin{cases} \mathsf{just}(a^+) \in [a]_q^- = \mathsf{dom}(\Gamma_a^\varphi)\,, \\ \mathsf{just}(\varphi(a)) \in [\varphi(a)]_p^- = \mathsf{cod}(\Gamma_a^\varphi)\,. \end{cases}$$

Moreover $\mathsf{just}(\varphi(a)) = \varphi(\mathsf{just}(a))$ since φ preserves the static order.

(4). Assume $a^+ \rightarrowtriangle_q b^-$. Then $\varphi(a) \rightarrowtriangle_p \varphi(b)$, and by induction hypothesis we have:

$$b \sim_{\Gamma_b^\varphi} \varphi(b)\,.$$

But $[b^-]_q^- = [a]_q^- \cup \{b\}$ and $[\varphi(b)^-]_p^- = [\varphi(a)]_p^- \cup \{\varphi(b)\}$, so finally:

$$b \sim_{\Gamma_a^\varphi \cup \{(b,\varphi(b))\}} \varphi(b)\,.$$

The same reasoning applies for the symmetric condition. Assume $\varphi(a)^+ \rightarrowtriangle_p b$, then $\varphi^{-1}(b)$ exists by −-obsessionality of q.

(5). Same as for (4), except $[b^+]_q^- = [a]_q^-$ and $[\varphi(b)^+]_p^- = [\varphi(a)]_p^-$.

The same reasoning applies for the symmetric condition. Assume $\varphi(a)^- \rightarrowtriangle_p b$, then $\varphi^{-1}(b)$ exists by +-coveredness of q. □

We can now prove the following proposition:

Proposition 4.14 – Condition of −-obsessionality

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$ with p a mia.

Then, q is a −-obsessional expansion of p if and only if $q \sim p$.

Proof. ***If.*** We construct $\varphi : q \to p$ for all $a \in |q|$ by induction on $\leq_q$. The image is provided by bisimulation, its uniqueness by determinism and $-$-linearity. Condition (4) of bisimulation ensures $-$-obsessionality.

Only if. For $\varphi : q \to p$ and $a \in |q|$, we have

$$a \sim_{\Gamma_a^{\varphi}} \varphi(a)$$

by Lemma 4.13. In particular,

$$\mathsf{init}(q) \sim_{\{(\mathsf{init}(q),\mathsf{init}(p))\}} \mathsf{init}(p)$$

with $\Gamma^{\varphi}_{\mathsf{init}(q)} = \{(\mathsf{init}(q), \mathsf{init}(p))\}$. □

Altogether, we have:

Proposition 4.15

Consider two pointed mia $p_1, p_2 \in \mathbf{MIA}_{\bullet}(\mathsf{A})$ and two characteristic expansions $q_1 \in \exp_{\bullet}(p_1)$ and $q_2 \in \exp_{\bullet}(p_2)$ with an isomorphism $\varphi : (\!|q_1|\!) \cong (\!|q_2|\!)$.

If $q_1 \sim^{\varphi} q_2$, then $p_1 \cong p_2$.

Proof. As characteristic expansions, q_1 and q_2 are $-$-obsessional, so by Proposition 4.14 we have $q_1 \sim p_1$ and $q_2 \sim p_2$. Moreover, by Lemma 4.11, if $q_1 \sim^{\varphi} q_2$ then $q_1 \sim q_2$. By symmetry of $\sim$ (Lemma 4.12), we obtain:

$$p_1 \sim q_1 \sim q_2 \sim p_2\,.$$

Lemma 4.12 allows us to compose bisimulations, giving us $p_1 \sim p_2$ (and $p_2 \sim p_1$).

Since p_2 is a mia and $p_1 \sim p_2$, then by Proposition 4.14 p_1 is a $-$-obsessional expansion of p_2, and there is a (unique) morphism $\varphi : p_1 \to p_2$. Likewise, there exists $\psi : p_2 \to p_1$. But morphisms from an expansion to a mia are unique (Lemma 3.42), so $\varphi \circ \psi = \mathsf{id}_{p_2}$ and $\psi \circ \varphi = \mathsf{id}_{p_1}$, hence $p_1 \cong p_2$. □

Going back again to the proof sketch for positional injectivity, we now only need to prove that two characteristic expansions reaching the same position are bisimilar.

Proof Sketch for Positional Injectivity: Consider two total mia $p_1, p_2 \in \mathsf{Aug}_{\bullet}(\mathsf{A})$ such that $(\!|p_1|\!) = (\!|p_2|\!)$. Let $q_1 \in \exp_{\bullet}(p_1)$ be a characteristic expansion of p_1, then there exists $q_2 \in \exp_{\bullet}(p_2)$ such that $\varphi : (\!|q_1|\!) \cong (\!|q_2|\!)$. Then:
(i) $\varphi : (\!|q_1|\!) \cong (\!|q_2|\!)$ implies $q_1 \sim_{\varphi} q_2$ **(we still need to prove this!)**,
(ii) which implies $p_1 \cong p_2$
(Lemma 4.5 and Proposition 4.15).

4.2.3 Clones

In Section 4.1.2, we introduced *characteristic expansions* which, via duplications with well-chosen cardinalities, constrain the causal structure. More precisely, if p is a mia and $q \in \exp(p)$ is characteristic, one could look at a set of duplicated Player moves in $(\!|q|\!)$ of cardinality n and, decomposing $n = \sum_{i \in I} 2^i$, one could deduce that the causal predecessors of the $\mathbf{q}_j^+$'s are among the forks with cardinality 2^i for $i \in I$. But that is not enough: this does not tell us how to distribute the $\mathbf{q}_j^+$'s to the forks, and not all the choices will work: while the $\mathbf{q}_j^+$'s are copies, their respective causal follow-ups might differ. So the idea is simple: imagine that the causal follow-ups for the $\mathbf{q}_j^+$'s are already reconstructed. Then we may compare them using bisimulation, and replicate the same reasoning as above on bisimulation equivalence classes.

So we are left with the task of leveraging bisimulation to define an adequate equivalence relation on $|q|$. This leads to the notion of **clones**, our last technical tool.

Clones – definitions

We want a relation $a \approx^{\varphi} b$ that will allow a and b (and their follow-ups) to change their pointers through some unspecified Γ.

Definition 4.16 – Pointers-preserving context

Consider $q, p \in \mathsf{Aug}_{\bullet}(\mathsf{A})$ with $\varphi : (\!|q|\!) \cong (\!|p|\!)$ and a context Γ. We say Γ **preserves pointers** if for all $a \in \mathsf{dom}(\Gamma)$, $\varphi(\mathsf{just}(a)) = \mathsf{just}(\Gamma(a))$.

Definition 4.17 – Clone

Consider $q, p \in \mathsf{Aug}_{\bullet}(\mathsf{A})$ with $\varphi : (\!|q|\!) \cong (\!|p|\!)$, and $a \in |q|$, $b \in |p|$.

We say that a and b are **clones** through φ, written $a \approx^{\varphi} b$, if there is a pointer-preserving context Γ such that $a \sim^{\varphi}_{\Gamma} b$.

As $a \approx^{\varphi} b$ quantifies existentially over contexts, compositional properties of clones are more challenging. Nevertheless, via a canonical form for contexts, we show that $\approx$ also enjoy equivalence properties.

Context Properties

First, we prove some properties of contexts. For any event a of an augmentation $q \in \mathsf{Aug}(\mathsf{A})$, we define $\uparrow a$ the set of descendants of a, i.e. $\uparrow a = \{a' \mid a \leq_q a'\}$.

Lemma 4.18 – Matching contexts

Consider $q, p \in \mathsf{Aug}_{\bullet}(\mathsf{A})$ with $\varphi : q \cong p$. Consider $a \sim^{\varphi}_{\Gamma} b$ for some $a \in |q|$, $b \in |p|$ and a context Γ.

Then for any $a' \in \uparrow a$, there exists $b' \in \uparrow b$ such that $a' \sim^{\varphi}_{\Gamma \cup \Delta} b'$, where

$$a = a_0 \rightarrowtriangle_q \dots \rightarrowtriangle_q a' = a_n \,, \qquad b = b_0 \rightarrowtriangle_p \dots \rightarrowtriangle_p b' = b_n \,,$$

and Δ is the context defined as

$$\Delta = \{(a_i, b_i) \mid 0 \leq i \leq n \text{ and } \mathsf{pol}(a_i) = -\} \,.$$

Moreover, if $a \sim^{\varphi}_{\Gamma'} b$ for a context Γ', we also have $a' \sim^{\varphi}_{\Gamma' \cup \Delta} b'$.

Proof. By induction of the co-depth of a'. □

We can now define *minimal contexts*.

Definition 4.19 – Minimal context

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$, $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$, $a \in |q|$, $b \in |p|$ with $a \sim_\Gamma^\varphi b$ for some context Γ. We define $\Gamma_{a,b}$ the **minimal context** for $a \sim_\Gamma^\varphi b$ as the restriction of Γ such that:

$$c \in \mathsf{dom}(\Gamma_{a,b}) \quad \Leftrightarrow \quad \begin{cases} \exists a'^+ \in \uparrow a, \mathsf{just}(a') = c & (i) \\ \Gamma(c) \neq \varphi(c) & (ii) \end{cases}$$

for all $c \in |q|$, and symmetrically the mirror condition applies to any $d \in |p|$.

We can check that this indeed defines a context.

Lemma 4.20 – Minimal contexts are contexts

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$, $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$, $a \in |q|$, $b \in |p|$ with $a \sim_\Gamma^\varphi b$ for some context Γ.

Then $\Gamma_{a,b}$ is a context and $a \sim_{\Gamma_{a,b}}^\varphi b$.

Proof. Immediate by definition. First, any restriction of Γ enables a, b. Next, Γ is only needed for $a'^+ \in \uparrow a$, inductively following conditions (3) and (4) of Definition 4.7, so we only need to keep $c \in \mathsf{dom}(\Gamma)$ verifying condition (i). Finally, if $\Gamma(c) = \varphi(c)$, then we can safely remove $(c, \varphi(c))$ from Γ: whenever c is needed, we use condition (4) instead of (3). □

Moreover, we say $\Gamma_{a,b}$ is *the* minimal context because it is uniquely defined for any bisimilar a, b.

Lemma 4.21 – Minimality and unicity of minimal contexts

Consider $q, p \in \mathsf{Aug}(\mathsf{A})$ with $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$. Consider $a \in |q|$, $b \in |p|$ and Γ, Δ two contexts such that $a \sim_\Gamma^\varphi b$ and $a \sim_\Delta^\varphi b$.

Then $\Gamma_{a,b} = \Delta_{a,b}$. Moreover, $\Gamma_{a,b}$ is the minimal (for inclusion) context such that $a \sim_{\Gamma_{a,b}}^\varphi b$.

Proof. First, we prove that $\Gamma_{a,b} = \Delta_{a,b}$.

Consider $c \in \mathsf{dom}(\Gamma_{a,b})$. Then by (i) there exists $a' \in \uparrow a$ such that $\mathsf{just}(a') = c$. Since $a \sim_\Gamma^\varphi b$, there exists a matching $b' \in \uparrow b$ such that $\mathsf{just}(b) = \Gamma(c)$ (Lemma 4.18). Moreover, by condition (ii), we know $\Gamma(c) \neq \varphi(c)$. If $c \notin \mathsf{dom}(\Delta_{a,b})$, then by Lemma 4.18 and $a \sim_\Delta^\varphi b$ we have $\mathsf{just}(b') = \varphi(c)$, *i.e.* $\varphi(c) = \Gamma_{a,b}(c)$, contradiction. So $c \in \mathsf{dom}(\Delta_{a,b})$, and by Lemma 4.18 and $a \sim_\Delta^\varphi b$ we have $\mathsf{just}(b') = \Delta_{a,b}(c)$, *i.e.* $\Gamma_{a,b}(c) = \Delta_{a,b}(c)$.

Symmetrically, for any $c \in \mathsf{dom}(\Delta_{a,b})$, we have $c \in \mathsf{dom}(\Gamma_{a,b})$ and $\Delta_{a,b}(c) = \Gamma_{a,b}(c)$. Hence, $\Gamma_{a,b} = \Delta_{a,b}$.

We just proved that for *any* context Δ such that $a \sim_\Delta^\varphi b$, we have $\Gamma_{a,b} = \Delta_{a,b}$. Hence $\Gamma_{a,b} \subseteq \Delta$, so $\Gamma_{a,b}$ is minimal for inclusion. □

This lemma allows us to write *the minimal context for* a, b without mentioning Γ.

Clones as equivalence classes

A key notion in the proof of positional injectivity is the notion of clones, a variation of bisimulation. Although the added constraint on contexts makes transitivity more challenging, we can still prove a variation of Lemma 4.9. We use the same notation as for the usual bisimulation: for any a, b events of an augmentation q, $a \approx b$ means $a \approx^{\mathsf{id}} b$.

Lemma 4.22 – Transitivity of $\approx$

Consider q, p, r in $\mathsf{Aug}_\bullet(\mathsf{A})$ with $\varphi : (\!|q|\!) \cong (\!|p|\!)$ and $\psi : (\!|q|\!) \cong (\!|p|\!)$, and $a \in |q|$, $b \in |p|$, $c \in |r|$ such that $a \approx^{\varphi} b$ and $b \approx^{\psi} c$.

Then we also have $a \approx^{\psi \circ \varphi} c$.

Proof. Consider Γ and Δ the minimal contexts such that $a \sim^{\varphi}_{\Gamma} b$ and $b \sim^{\psi}_{\Delta} c$ (unique by Lemma 4.21).

If $\mathsf{cod}(\Gamma) = \mathsf{dom}(\Delta)$, the result is immediate by Lemma 4.9: we get $a \sim^{\psi\circ\varphi}_{\Delta\circ\Gamma} c$ with, for any $d \in \mathsf{dom}(\Delta \circ \Gamma) = \mathsf{dom}(\Gamma)$,

$$\psi(\varphi(\mathsf{just}(d))) = \psi(\mathsf{just}(\Gamma(d))) = \mathsf{just}(\Delta(\Gamma(d)))$$

so $\Delta \circ \Gamma$ preserves pointers, and $a \approx^{\Delta\circ\Gamma} c$.

Now, assume there exists $d \in \mathsf{cod}(\Gamma)$ such that $d \notin \mathsf{dom}(\Delta)$. Since Γ is minimal, there exists $b' \in \uparrow b$ such that $\mathsf{just}(b') = d$. By $b \sim^{\Delta}_{\psi} c$ and Lemma 4.18, there exists a matching $c' \in \uparrow c$ such that $b' \sim^{\psi}_{\Delta\cup\Delta'} c'$, with Δ' mapping negative moves between b and b' to negative moves between c and c'. Since $d \in \mathsf{cod}(\Gamma)$, we do not have $d \geq_p b$, so $d \notin \mathsf{dom}(\Delta')$. Hence, $\mathsf{just}(c') = \psi(d)$ and $\psi(d) \notin \mathsf{cod}(\Delta)$. So we can write

$$b \sim^{\psi}_{\Delta\cup\{(d,\psi(d))\}} c$$

where $\Delta \cup \{(d, \psi(d))\}$ preserves pointers.

Likewise, for any $d \in \mathsf{dom}(\Delta)$ such that $d \notin \mathsf{cod}(\Gamma)$, we have $\Gamma \cup \{(\varphi^{-}1(d), d)\}$ well-defined and pointer preserving, such that

$$a \sim^{\varphi}_{(\Gamma\cup\{(\varphi^{-}1(d),d)\}} b\,.$$

This allows us to define the following pointer-preserving contexts:

$$\begin{aligned}\Gamma' &= \Gamma \cup \{(\varphi^{-1}(d), d) \mid d \in \mathsf{dom}(\Delta), d \notin \mathsf{cod}(\Gamma)\}\\ \Delta' &= \Delta \cup \{(d, \psi(d)) \mid d \in \mathsf{cod}(\Gamma), d \notin \mathsf{dom}(\Delta)\}.\end{aligned}$$

Then $a \sim^{\varphi}_{\Gamma'} b$ and $b \sim^{\psi}_{\Delta'} c$, so by Lemma 4.9, we have $a \sim^{\psi\circ\varphi}_{\Delta'\circ\Gamma'} c$. Moreover, $\Delta' \circ \Gamma'$ preserves pointers, so finally $a \approx^{\psi\circ\varphi} c$. □

This allows us to prove equivalence properties for the clone relation.

Lemma 4.23 – Equivalence for $\approx$

Consider $q, p, r \in \mathsf{Aug}_\bullet(\mathsf{A})$ augmentations, with $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$ and $\psi\colon (\!|p|\!) \cong (\!|r|\!)$, and events $a \in |q|$, $b \in |p|$, $c \in |r|$. Then:

$$\begin{array}{rl} \textit{reflexivity:} & a \approx^{\mathsf{id}} a, \\ \textit{transitivity:} & \text{if } a \approx^{\varphi} b \text{ and } b \approx^{\psi} c, \text{ then } a \approx^{\psi\circ\varphi} c, \\ \textit{symmetry:} & \text{if } a \approx^{\varphi} b \text{ then } b \approx^{\varphi^{-1}} a. \end{array}$$

Proof. ***Reflexivity.*** By reflexivity of $\sim^{\mathsf{id}}$ (Lemma 4.9), $a \sim^{\mathsf{id}} a$, which implies $a \approx^{\mathsf{id}} a$.

Transitivity. See Lemma 4.22.

Symmetry. Immediate by symmetry of $\sim^{\varphi}_{\Gamma}$ (Lemma 4.9): if Γ preserves pointers, so does Γ^{-1}. □

Clones through id in characteristic expansions will be especially interesting, because then we can partition equivalence classes of $\approx^{\mathsf{id}}$ into successors of forks.

Forks and Clone Classes Cardinalities

Lemma 4.24 – Forks generates clones

Consider $p \in \mathbf{MIA}(\mathsf{A})$ and $q \in \exp_\bullet(p)$ a $-$-obsessional expansion.

Then, for all $a_1^-, a_2^- \in X \in \mathsf{Fork}(q)$, for all $a_1^- \rightarrow_q b_1^+$ and $a_2^- \rightarrow_q b_2^+$, we have $b_1 \approx b_2$.

Proof. If $X = \{\mathsf{init}(q)\}$, then $a_1 = a_2 = \mathsf{init}(q)$ and the result is immediate by determinism and reflexivity.

Otherwise, assume $X \neq \{\mathsf{init}(q)\}$ and $a_1 \neq a_2$.

First, let us prove that b_1 and b_2 are bisimilar. Since q is a $-$-obsessional expansion of p, there exists a unique (by Lemma 3.42) morphism $\varphi\colon q \to p$.

Recall Lemma 4.13. Writing

$$\Gamma_i = \Gamma^{\varphi}_{b_i}\colon [b_i]^-_q \cong [\varphi(b_i)]^-_p \qquad \text{for } i = 1, 2\,,$$

we have:

$$b_i \sim_{\Gamma_i} \varphi(b_i) \qquad \text{for } i = 1, 2\,.$$

By $-$-linearity of p, we know that $\varphi(a_1) = \varphi(a_2)$, so $\varphi(b_1) = \varphi(b_2)$ by determinism. Hence by Lemma 4.12, we have

$$b_1 \sim_{\Gamma_2^{-1}\circ\Gamma_1} b_2\,.$$

Writing $\Gamma = \Gamma_2^{-1} \circ \Gamma_1$, it remains to check that Γ preserves pointers. Consider $c \in \mathsf{dom}(\Gamma) = [b_1]^-_q$. If $c = a_1$, then $\Gamma_1(a_1) = \varphi(a_1) = \Gamma_2(a_2)$ by $-$-linearity of p. By courtesy and since $a_1, a_2 \in X \in \mathsf{Fork}(q)$, both have the same pointer $d = \mathsf{just}(a_1) = \mathsf{just}(a_2) = \mathsf{pred}(a_1) = \mathsf{pred}(a_2)$. If $c \neq a_1$, then $c \leq_q d$, so $c \in \mathsf{dom}(\Gamma)_2$ and $\Gamma_1(c) = \varphi(c) = \Gamma_2(c)$, hence $\Gamma(c) = c$. In both cases, Γ preserves pointers, so $b_1 \approx b_2$. □

In particular, if a clone class includes a positive move, it also has all its cousins triggered by the same fork – so clone classes may be partitioned following forks.

Lemma 4.25 – Partition Lemma

Consider $p \in \mathbf{MIA}(\mathsf{A})$ and $q \in \exp_\bullet(p)$ a characteristic expansion. Consider Y a clone class of positive events in $|q|$, with

$$\sharp Y = \sum_{i \in I} 2^i \qquad \text{for } I \subset \mathbb{N} \text{ finite.}$$

Then for all $i \in \mathbb{N}$, we have:

$$i \in I \text{ iff } \exists X_i \in \mathsf{Fork}(q) \text{ such that } \begin{cases} \sharp X_i = 2^i\,, \\ \exists a \in X_i, b \in Y \text{ s.t. } a \rightarrow_q b\,. \end{cases}$$

Moreover, we can partition Y into:

$$Y = \biguplus_{i \in I} Y_i$$

with for all $i \in I$, $\sharp Y_i = 2^i$ and for all $b \in Y_i$, there is a unique $a \in X_i$ such that $a \rightarrow_q b$.

Proof. For any $i \in \mathbb{N}$, we write X_i the fork of q of cardinality 2^i, if it exists. Consider the set:

$$J = \{j \in \mathbb{N} \mid X_j \text{ exists}, \exists a \in X_j, b \in Y \text{ s.t. } a \rightarrow_q b\}\,.$$

Any $b \in Y$ is positive, and so the unique (by determinism) successor of some negative event a. But a appears in some fork $X \in \mathsf{Fork}(q)$, and by Lemma 4.24, all events of X are predecessors of events of Y. So, for any $j \in J$, the set of successors of events of X_j is $Y_j \in Y$, with $\sharp Y_j = \sharp X_j$ by determinism. Finally, we have:

$$Y = \bigcup_{j \in J} Y_j$$

where the union is disjoint since q is forest-shaped. Therefore:

$$\sharp Y = \sum_{j \in J} \sharp Y_j = \sum_{j \in J} \sharp X_j = \sum_{j \in J} 2^j\,.$$

By uniqueness of the binary decomposition of $\sharp Y$, we have $I = J$, which concludes the proof by definition of J. □

4.3 Total MIAs are Positionally Injective in PCG

We now prove the core of the injectivity argument: given two mias p_1, p_2 with characteristic expansions q_1, q_2 and $\varphi\colon (\!|q_1|\!) \cong (\!|q_2|\!)$, we have

Claim 3: for all $a^+ \in |q_1|$, $a \approx^\varphi \varphi(a)$.

Reminder: The **co-depth** of a is the maximal length k of a causal chain

$$a = a_1 \rightarrow_{q_1} \dots \rightarrow_{q_1} a_k\,.$$

The idea of the proof is the following: we reason by induction on the *co-depth* of a, using properties of bisimulations and Lemma 4.25, to prove **Claim 3**. Then we deduce $\mathsf{init}(q_1) \sim^\varphi \mathsf{init}(q_2)$, hence $q_1 \sim^\varphi q_2$.

Proving **Claim 3** requires some care, because cloning is defined via a context and the successors of a might not share the same. Hence we start by defining a canonical form for pointers-preserving contexts.

Lemma 4.26 – Minimal context for clones

Consider $q \in \mathsf{Aug}_\bullet(\mathsf{A})$ and $a, b \in |q|$ such that $a \approx b$.

Then the minimal context for a, b is either empty or $\Gamma\colon \{c\} \cong \{d\}$.

Proof. Assume, seeking a contradiction, that the minimal context Γ has at least two distinct elements $c_1, c_2 \in \mathsf{dom}(\Gamma)$. Remark that since $a \approx b$, there exists Δ a pointers-preserving context such that $a \sim_\Delta b$, and since Γ is a restriction of Δ (see Lemma 4.27), Γ also preserves pointers.

Now, by condition (i) of Definition 4.19, $c_1 \leq_q a$ and $c_2 \leq_q a$. But q is forest shaped, so $c_1 \leq_q c_2$ or $c_2 \leq_q c_1$. *W.l.o.g.*, assume that it is the former. Then by courtesy, $\mathsf{just}(c_1) \leq_q \mathsf{just}(c_2)$ as well, and since $c_1 \neq c_2$, we have:

$$c_1 \leq_q \mathsf{just}(c_2) \tag{4.2}$$

For the same reason, $\Gamma(c_1) \leq_q \Gamma(c_2)$ or $\Gamma(c_2) \leq_q \Gamma(c_1)$.

If it is the latter, this entails that $\mathsf{just}(\Gamma(c_2)) \leq_q \mathsf{just}(\Gamma(c_1))$ by courtesy; *i.e.* since Γ preserves pointers, $\mathsf{just}(c_2) \leq_q \mathsf{just}(c_1)$. So $\mathsf{just}(c_1) = \mathsf{just}(c_2)$, and because $c_1, c_2 \leq_q a$ we have $c_1 = c_2$, contradiction.

So, $\Gamma(c_1) \leq_q \Gamma(c_2)$, and $\Gamma(c_1) \neq \Gamma(c_2)$ by hypothesis. By courtesy, this entails that:

$$\Gamma(c_1) \leq_q \mathsf{just}(\Gamma(c_2)) \tag{4.3}$$

Moreover, Γ preserves pointers, so $\mathsf{just}(c_2) = \mathsf{just}(\Gamma(c_2))$. Hence, (4.3) rewrites to:

$$\Gamma(c_1) \leq_q \mathsf{just}(c_2) \tag{4.4}$$

By forestiality of q, (4.2) and (4.4) implies that c_1 and $\Gamma(c_1)$ are comparable for $\leq_q$. But they are negative and share the same justifier, so they have the same antecedent by courtesy. This implies $c_1 = \Gamma(c_1)$, which contradicts condition (ii) of Definition 4.19. □

Given clones with a context Γ, we can also extend Γ in some ways.

Lemma 4.27 – Extending contexts for clones

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$ two augmentations such that there exists an isomorphism $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$. Consider events $a \in |q|$, $b \in |p|$ and a pointing context Γ such that $a \sim^{\varphi}_{\Gamma} b$.

Then for any $c \in |q|$ such that:

$$\begin{array}{llll} (i) & c \notin \mathsf{dom}(\Gamma) & (iii) & c \notin \uparrow a \\ (ii) & \varphi(c) \notin \mathsf{cod}(\Gamma) & (iv) & \varphi(c) \notin \uparrow b \end{array}$$

we have $a \sim^{\varphi}_{\Gamma \cup \{(c,\varphi(c))\}} b$.

Moreover, for any $c \in |q|$ and $d \in |p|$ such that

$$\begin{array}{llllll} (i) & c \notin \mathsf{dom}(\Gamma) & (iii) & c \notin \uparrow a & (v) & \forall a' \in \uparrow a, \mathsf{just}(a') \neq c \\ (ii) & d \notin \mathsf{cod}(\Gamma) & (iv) & d \notin \uparrow b & (vi) & \forall b' \in \uparrow b, \mathsf{just}(b') \neq d \end{array}$$

then we also have $a \sim^{\varphi}_{\Gamma \cup \{(c,d)\}} b$.

Proof. Straightforward by induction. Either c is never used in the bisimulation (*i.e.* no one in $\uparrow a$ points to c), and we can pair it with any d which is not used either and add (c, d) to Γ (as long as we still have $(\Gamma \cup \{(c, d)\}) \vdash (a, b)$); or it is used with condition *(4)* of Definition 4.7 and we can add $(c, \varphi(c))$ to Γ and use condition *(3)* instead. □

Both those lemmas will help us constructing matching contexts in order to prove **Claim 3**. Before moving on to the proof, we need a last lemma on co-depth of bisimilar events.

Lemma 4.28 – Bisimilar events have the same co-depth

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$ two augmentations such that there exists an isomorphism $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$. Consider events $a \in |q|$, $b \in |p|$ and a pointing context Γ such that $a \sim^{\varphi}_{\Gamma} b$.

Then a and b have the same co-depth.

Proof. Straightforward by induction. □

We now state our main auxiliary lemma:

Lemma 4.29 – Lifing clone classes

Consider $q, p \in \mathsf{Aug}_\bullet(\mathsf{A})$ with $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$. Consider $a^+ \in |q|$ such that $\mathsf{succ}(a) = \bigcup_{i \in I} X_i$, where $I \subseteq \mathbb{N}$ and for $i \in I$,

$$X_i = \{b_{i,1}, \ldots, b_{i,2^i}\} \in \mathsf{Fork}(q)\,.$$

Then we have $a \approx^{\varphi} \varphi(a)$, provided the two conditions hold:

$$\text{if } b_{i,j} \rightarrow_q c_{i,j}, \text{ then } \varphi(b_{i,j}) \rightarrow_p d_{i,j} \text{ and } c_{i,j} \approx^{\varphi} d_{i,j}\,, \tag{4.5}$$

$$\text{if } \varphi(b_{i,j}) \rightarrow_p d_{i,j}, \text{ then } b_{i,j} \rightarrow_q c_{i,j} \text{ and } c_{i,j} \approx^{\varphi} d_{i,j}\,. \tag{4.6}$$

Proof. First, remark that:

$$\partial_q(a) = \partial_p(\varphi(a))\,, \tag{4.7}$$
$$\varphi(\mathsf{just}(a)) = \mathsf{just}(\varphi(a))\,. \tag{4.8}$$

For any $i \in I$ and $1 \leq j \leq 2^i$, we have $b_{i,j} \approx^\varphi \varphi(b_{i,j})$. Indeed:

- if $b_{i,j}$ has no successor, then by 4.6 neither does $\varphi(b_{i,j})$, and $b_{i,j} \sim^\varphi \varphi(b_{i,j})$;
- otherwise, $b_{i,j}$ has a unique (by determinism) successor $c_{i,j}$, and by 4.5 we have $\varphi(b_{i,j}) \multimap_p d_{i,j}$ and $c_{i,j} \approx^\varphi d_{i,j}$.

In both cases, we obtain $b_{i,j} \approx^\varphi \varphi(b_{i,j})$. Let $\Gamma_{i,j}$ be the minimal context for $b_{i,j} \approx^\varphi \varphi(b_{i,j})$.

We wish to take the union of all $\Gamma_{i,j}$ as the context for a and $\varphi(a)$, but this is only possible if they are "compatible": we must ensure that for all $e \in |q|$, $i, k \in I$, $1 \leq j \leq 2^i$ and $1 \leq l \leq 2^k$, if there exists $c'_{i,j} \in \uparrow b_{i,j}$ and $c'_{k,l} \in \uparrow b_{k,l}$ having both e has a justifiter, then their matching $d'_{i,j} \in \uparrow \varphi(b_{i,j})$ and $d'_{k,l} \in \uparrow \varphi(b_{k,l})$ also have the same justifier.

This can only be a problem if $e \in \mathsf{dom}(\Gamma_{i,j})$ or $e \in \mathsf{dom}(\Gamma_{k,l})$, as otherwise both justifiers for $d'_{i,j}$ and $d'_{k,l}$ are $\varphi(e)$. By Lemma 4.26, for all i, j, the context $\Gamma_{i,j}$ has either one or zero element. If all $\Gamma_{i,j}$ are empty, we can directly lift the clone relation to a and $\varphi(a)$.

Otherwise, consider i, j such that:

$$\Gamma_{i,j} \colon \{e_{i,j}\} \cong \{f_{i,j}\}\,.$$

By Definition 4.19, we have:

$$e_{i,j} \in [b_{i,j}]_q^- \quad \text{and} \quad f_{i,j} \in [\varphi(b_{i,j})]_p^-\,.$$

Could we have $f_{i,j} = \varphi(b_{i,j})$? Since $\Gamma_{i,j}$ preseves pointers, $e_{i,j}$ and $f_{i,j}$ have the same justifier through φ; but the only $e \in [b_{i,j}]_q^-$ such that $\varphi(\mathsf{just}(e)) = \mathsf{just}(\varphi(b_{i,j}))$ is $b_{i,j}$, which contradicts minimality of $\Gamma_{i,j}$. Hence we have:

$$f_{i,j} \in [\varphi(a)]_p^-\,.$$

Now, assume that for some k, l, there exists $c'_{k,l} \in \uparrow b_{k,l}$ such that $\mathsf{just}(c'_{k,l}) = e_{i,j}$. Since $b_{k,l} \approx^\varphi \varphi(b_{k,l})$, there is a matching $d'_{k,l} \in \uparrow \varphi(b_{k,l})$ such that:

$$\varphi(\mathsf{just}(e_{i,j})) = \mathsf{just}(\mathsf{just}(d'_{k,l}))\,. \tag{4.9}$$

For $b_{i,j} \sim^\varphi_{\Gamma_{i,j}} \varphi(b_{i,j})$ and $b_{k,l} \sim^\varphi_{\Gamma_{k,l}} \varphi(b_{k,l})$ to be compatible, we need

$$\mathsf{just}(d'_{k,l}) = f_{i,j}\,.$$

Since $\Gamma_{i,j}$ preserves pointers, we have

$$\varphi(\mathsf{just}(e_{i,j})) = \mathsf{just}(f_{i,j})\,. \tag{4.10}$$

Combining Equations 4.9 and 4.10, we obtain

$$\mathsf{just}(\mathsf{just}(d'_{k,l})) = \mathsf{just}(f_{i,j})\,. \tag{4.11}$$

where $\mathsf{just}(d'_{k,l}) \in [d'_{k,l}]^-_p$ and $f_{i,j} \in [\varphi(a)]^-_p$. But $[\varphi(a)]^-_p \subseteq [d'_{k,l}]^-_p$ which is a fully ordered set for $\leq_p$, so $\mathsf{just}(d'_{k,l})$ and $f_{i,j}$ are comparable for $\leq_p$. Moreover, they are negative, so by courtesy

$$\mathsf{just}(\mathsf{just}(d'_{k,l})) = \mathsf{just}(f_{i,j}) \quad \text{iff} \quad \mathsf{pred}(\mathsf{just}(d'_{k,l})) = \mathsf{pred}(f_{i,j})\,.$$

Hence, we have $\mathsf{pred}(\mathsf{just}(d'_{k,l})) = \mathsf{pred}(f_{i,j})$. Since $\mathsf{just}(d'_{k,l})$ and $f_{i,j}$ are comparable, we obtain $\mathsf{just}(d'_{k,l}) = f_{i,j}$.

So all contexts $\Gamma_{i,j}$ are compatible, and we can define:

$$\Gamma = \bigcup_{i,j} \Gamma_{i,j}\,.$$

Via Lemma 4.27, it follows that:

$$\forall i,j, \quad b_{i,j} \sim^{\varphi}_{\Gamma} \varphi(b_{i,j})\,,$$

which entails that $a \sim^{\varphi}_{\Gamma} \varphi(a)$ by two steps of the bisimulation game. Since all $\Gamma_{i,j}$'s preserve pointers, so does Γ; hence $a \approx^{\varphi} \varphi(a)$. □

We are finally able to prove **Claim 3** the core of the injectivity argument.

Lemma 4.30 – Key lemma

Consider $p_1, p_2 \in \mathbf{MIA}(\mathsf{A})$ and $q_1 \in \exp_\bullet(p_1)$, $q_2 \in \exp_\bullet(p_2)$ two characteristic expansions with $\varphi : \llparenthesis q_1 \rrparenthesis \cong \llparenthesis q_2 \rrparenthesis$.

Then for all $a^+ \in |q_1|$, we have $a \approx^{\varphi} \varphi(a)$.

Proof. Recall that the *co-depth* of $a \in |q_i|$ is the maximal length k of a chain $a = a_1 \rightharpoonup_{q_i} \ldots \rightharpoonup_{q_i} a_k$. We show by induction on k the two symmetric properties:

(P_k) for all $a^+ \in |q_1|$ of co-depth $k' \leq k$, we have $a \approx^{\varphi} \varphi(a)$,
(P'_k) for all $a^+ \in |q_2|$ of co-depth $k' \leq k$, we have $a \approx^{\varphi^{-1}} \varphi^{-1}(a)$.

($\mathbf{P_0}$) First, consider $a^+ \in |q_1|$ maximal for $\leq_{q_1}$. By courtesy, a^+ is also maximal for $\leq_{\llparenthesis q_1 \rrparenthesis}$, so $\varphi(a)$ is maximal is q_2 for both $\leq_{\llparenthesis q_1 \rrparenthesis}$ and $\leq_{q_1}$. Since φ also preserves the arena image, we immediately have $a \sim^{\varphi} \varphi(a)$ with the empty context, so $a \approx^{\varphi} \varphi(a)$.

($\mathbf{P'_0}$) By the same reasoning, for any $a^+ \in |q_2|$ maximal for $\leq_{q_2}$, we have $a \approx^{\varphi^{-1}} \varphi^{-1}(a)$.

($\mathbf{P_{k+2}}$) Now, assuming (P_k) and (P'_k), consider $a^+ \in |q_1|$ of co-depth $k+2$. Then the successors of a partition as:

$$\mathsf{succ}(a) = \biguplus_{i \in I} X_i$$

with for all $i \in I$,

$$X_i^1 = \{b_{i,1}, \ldots, b_{i,2^i}\} \in \mathsf{Fork}(q_1)\,.$$

Since configuration isomorphisms preserve causal links from positive to negative moves, the successors of $\varphi(a)$ are:

$$\mathsf{succ}(\varphi(a)) = \biguplus_{i \in I} \varphi(X_i) \quad \text{with } \forall i \in I,\ X_i \in \mathsf{Fork}(q_2)\,.$$

Now, for any $i \in I$, $1 \leq j \leq 2^i$, we claim:

$$\text{if } b_{i,j} \rightarrowtriangle_{q_1} c_{i,c}\text{, then } \varphi(b_{i,j}) \rightarrowtriangle_{q_2} d_{i,j} \text{ and } c_{i,j} \approx^{\varphi} d_{i,j}\,. \tag{4.12}$$

Indeed, consider $Y_{i,j}$ the clone equivalence class of $c_{i,j}$ in q_1. Since the clone relation preserves co-depth, it follows from the induction hypotheses (P_k) and $P'_k)$ and compositional properties of clones (Lemma 4.23) that $\varphi(Y_{i,j})$ is a clone class. Then by the partition lemma (4.25), $\sharp Y_{i,j}$ has 2^i in its binary decomposition – and as φ preserves forks, so does $\sharp\varphi(Y_{i,j})$. So by Lemma 4.25, there is $\varphi(b_{i,j}) \in \varphi(X_i)$ and $d_{i,j} \in \varphi(Y_{i,j})$ such that $\varphi(b_{i,j}) \rightarrowtriangle_{q_2} d_{i,j}$. Since both $\varphi(c_{i,j})$ and $d_{i,j}$ are in $\varphi(Y_{i,j})$, we have $\varphi(c_{i,j}) \approx d_{i,j}$. Moreover, we have $c_{i,j} \approx^{\varphi} \varphi(c_{i,j})$ by induction hypothesis (P_k). By compositional properties of clones (Lemma 4.23), we obtain $c_{i,j} \approx^{\varphi} d_{i,j}$, which concludes the proof of Equation 4.12. Likewise, the mirror property of 4.12 also holds. Having verified all hypotheses for the lifting lemma (4.29), we can now apply it to get $a \approx^{\varphi} \varphi(a)$.

($\mathbf{P'_{k+2}}$) The reasonning is the same as for P_{k+2}.

Conclusion. For $a^+ \in |q_1|$ of any co-depth, $a \approx^{\varphi} \varphi(a)$. □

Remark: Since $\rightarrowtriangle_{q_1}$ alternates polarities and augmentations are $+$-covered, all positive events have even co-depths, so for the induction we go from (P_k) to (P_{k+2}) (same for (P'_k) and (P'_{k+2})).

This lemma gives us the last missing piece to prove positional injectivity:

Theorem 4.31 – Positional Injectivity in PCG

Consider two total $p_1, p_2 \in \mathbf{MIA}(\mathsf{A})$. Then:

$$p_1 \cong p_2 \Leftrightarrow (\!|p_1|\!) = (\!|p_2|\!)\,.$$

Proof. The implication $\Rightarrow$ is immediate by definition.

For the reverse implication, assume $(\!|p_1|\!) = (\!|p_2|\!)$. Consider a characteristic expansion $q_1 \in \mathrm{exp}_\bullet(p_1)$. By hypothesis, there exists $q_2 \in \mathrm{exp}_\bullet(p_2)$ with $\varphi\colon (\!|q_1|\!) \cong (\!|q_2|\!)$. By Lemma 4.5, q_2 also is a characteristic expansion. If both q_i's are empty, there is nothing to prove: we directly have $p_1 = p_2$ the empty augmentation on A. Otherwise, q_1 has an initial event $\mathsf{init}(q_1)$, and since φ preserves minimality,

$$\varphi(\mathsf{init}(q_1)) = \mathsf{init}(q_2)\,.$$

We write $a_i = \mathsf{init}(q_i)$. Augmentations are negative, so both a_i's are negative. By determinism, they have unique successors

$$b_1^+ = \mathsf{succ}(a_1^-) \quad \text{and} \quad b_2^+ = \mathsf{succ}(a_2^-)\,.$$

For $i = 1, 2$, b_i is the only event with its co-depth since all other events except a_i are below it. By Lemma 4.28, it means X_i the clone class of b_i is a singleton $X_i = \{b_i\}$. But φ preserves clone classes by Lemma 4.30, so $\varphi(X_1) = \{\varphi(b_1)\}$ also is a clone class. By the partition lemma (4.25), we obtain $\varphi(b_1) = b_2$. But $b_1 \approx^\varphi \varphi(b_1)$ by Lemma 4.30, so we get:

$$b_1 \approx^\varphi b_2\,.$$

So $b_1 \sim^\varphi b_2$ (with the emply context since for $i = 1, 2$, the only event above b_i is a_i, and we already have $a_2 = \varphi(a_1)$). Therefore,

$$a_1 \sim^\varphi a_2 \quad \textit{i.e.} \quad q_1 \sim^\varphi q_2\,.$$

By Proposition 4.15, we finally obtain $p_1 \cong p_2$. □

4.4 Positional Injectivity in HO

We now come back to our initial **Question 4** from Chapter 3:

Question: are innocent strategies *in* HO *games* positionally injective?

4.4.1 Total Finite Innocent Strategies are Positionally Injective in HO

Using the isomorphisms defined in the previous chapter between PCG and HO, we can easily translate Theorem 4.31 to HO strategies.

Theorem 4.32 – Positional Injectivity in HO

Consider two total, finite innocent strategies σ, τ on an arena A. Then:

$$\sigma = \tau \Leftrightarrow (\!|\sigma|\!) = (\!|\tau|\!)\,.$$

Proof. The first implication $\Rightarrow$ is immediate.

For the reverse implication, assume $(\!|\sigma|\!) = (\!|\tau|\!)$. But $(\!|\sigma|\!) = (\!|\mathsf{MIA}(\sigma)|\!)$ by Proposition 3.48, so by Theorem 4.31 we obtain $\mathsf{MIA}(\sigma) \cong \mathsf{MIA}(\tau)$. By Theorem 3.40, this implies $\sigma = \tau$. □

4.4.2 Beyond Total Finite Strategies

Is it possible to expand this result to partial or infinite strategies?

Our proof method requires totality to ensure that being a characteristic expansion is a property of the position of an augmentation, and finiteness to be able to reason co-inductively on characteristic expansions.

We do not know if positional injectivity still holds for total infinite strategies, or for partial finite strategies; however we do know that partial infinite strategies in general are *not* positionally injective.

Consider the infinitary terms

$$f : \alpha \to \alpha \to \alpha \vdash T_1, T_2, L, R : \alpha$$

recursively defined as

$$T_1 = f\ T_2\ R, \qquad T_2 = f\ L\ T_1, \qquad L = f\ L\ \bot, \qquad R = f\ \bot\ R,$$

in an infinitary simply-typed λ-calculus with divergence $\bot$.

Now, consider $M_1 = \lambda f.\, T_1$ and $M_2 = \lambda f.\, T_2$, and their interpretation as mii's on the arena $[\![(\alpha \to \alpha \to \alpha) \to \alpha]\!]_{\mathsf{PCG}}$. Figure 4.8 shows the arena, using indices to help distinguish between the different moves. Figures 4.9 and 4.10 represent the respective interpretations p_1 and p_2 of M_1 and M_2, with loops indicating regular infinite trees. Clearly M_1 and M_2 are different, and so are their interpretation p_1 and p_2.

We consider positions reached by well-opened plays – or equivalently, by (iso-)expansions of the isogmentations presented in Figures 4.9 and 4.10. Ignoring the initial $\mathbf{q}_4^-$, a position is a multiset of **bricks** as in Figure 4.11, with $i \in \mathbb{N}$ occurrences of $\mathbf{q}_1^-$ and $j \in \mathbb{N}$ occurrences of $\mathbf{q}_2^-$. A brick with $i = j = 0$ is a **leaf**. The position is balanced if it has as many Opponent as Player moves.

Now, any balanced position can be realized in p_1 by first placing bricks with occurrences of both $\mathbf{q}_1^-$ and $\mathbf{q}_2^-$ greedily alongside the *spine* – shown in red in Figures 4.9 and 4.10. At each step, we continue from only one of the copies opened, leaving others dangling. If this gets stuck, apart from leaves we are left with only $\mathbf{q}_1^-$'s, or only $\mathbf{q}_2^-$'s, and in any case there is always a matching non-spine infinite branch available. Finally, leaves can always be placed as their number matches that of trailing negative moves by the balanced hypothesis. The same goes for p_2: any balanced position can be reached with an iso-expansion of p_2.

Moreover, all positions reached by expansions of p_1 or of p_2 are balanced, by determinism and $+$-coveredness.

We obtain that the positions of p_1 and the positions of p_2 both are *exactly* the balanced positions in $[\![(\alpha \to \alpha \to \alpha) \to \alpha]\!]$. Hence,

$$(\!|[\![\lambda f^{\alpha\to\alpha\to\alpha}.\, T_1]\!]_{\mathsf{HO}}|\!) = (\!|[\![\lambda f^{\alpha\to\alpha\to\alpha}.\, T_2]\!]_{\mathsf{HO}}|\!)\,,$$

and positional injectivity fails.

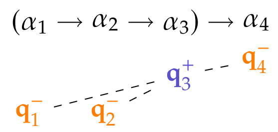


Figure 4.8: $[\![(\alpha \to \alpha \to \alpha) \to \alpha]\!]_{\mathsf{PCG}}$.

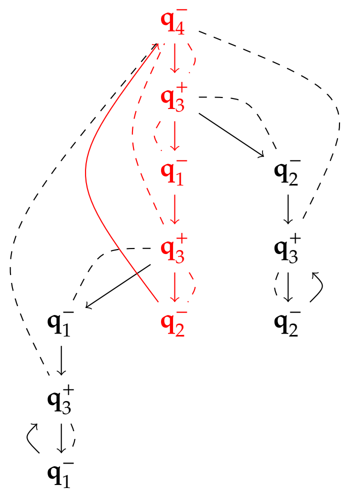


Figure 4.9: $\mathsf{MII}([\![\lambda f^{\alpha\to\alpha\to\alpha}.\, T_1]\!]_{HO})$

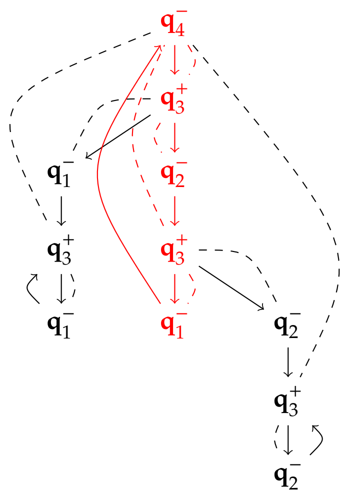


Figure 4.10: $\mathsf{MII}([\![\lambda f^{\alpha\to\alpha\to\alpha}.\, T_2]\!]_{HO})$

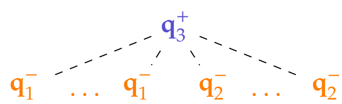


Figure 4.11: Bricks

4.5 Conclusion

Innocent strategies in HO games are not positional, but we show that *total finite* innocent strategies enjoy *positional injectivity* – and likewise, total finite mia's in PCG are positionally injective. However, the property fails in general, for partial infinite innocent strategies.

This result may be useful in the game semantics toolbox: proving two (total, finite) innocent strategies equal now requires only to compare their positions, which can be easier to handle than plays with pointers.

Composition and Resource Calculus Semantics

In this part, we introduce the dynamical *aspect of Pointer Concurrent Games: we define the composition of augmentations and expose the categorical structure of* PCG*. We also study the interpretation of resource calculus in* PCG*.*

In Chapter 5, we start by constructing a bijection between isogmentations and normal resource terms.

In Chapter 6, we define the ***composition*** *of augmentations, and present* PCG *as a category. We also show how this composition coincides with the one from* HO *games, following our previous isomorphism between* PCG *and* HO*.*

In Chapter 7, we introduce ***resource categories****, a new categorical structure that is relevant to obtain a model of the resource calculus. We prove that there is a sound interpretation of resource terms in a resource category. We also investigate the links with differential categories.*

In Chapter 8, we finally show that PCG *indeed forms a resource category, completing the previous isomorphism between resource calculus and games: the correspondence between normal resource terms and isogmentations refines into a denotational interpretation, invariant under reduction, of resource terms as "strategies" – weighted sums of isogmentations.*

Most of this section is adapted from the articles [5] and [3].

[5]: Blondeau-Patissier, Clairambault, and Auclair (2023), 'Strategies as Resource Terms, and Their Categorical Semantics'

[3]: Blondeau-Patissier (2024), 'Resource Categories from Differential Categories'

Augmentations are Normal Resource Terms 5

As stated in Chapter 1, resource terms and plays in HO games are similar: Tsukada and Ong [40] showed that certain normal and η-long resource terms correspond bijectively to plays in HO games, *up to Opponent's scheduling of the independent explorations of separate branches of the term*. This scheduling is formalized by Melliès' homotopy equivalence on plays (see Chapter 3).

[40]: Tsukada and Ong (2016), 'Plays as Resource Terms via Non-idempotent Intersection Types'

Our game model PCG relies on *augmentations*, which correspond to HO plays quotiented by this relation; so it is natural to investigate the relation between augmentations and resource terms. We could try and compose the bijections from Δ to HO and from HO to PCG, but the correspondence between Δ and PCG can actually be studied on its own, in a more direct way than the correspondence presented in [40]. In this chapter, we give the explicit bijection between (normal, η-long) resource terms and isogmentations (isomorphism classes of augmentations).

First, we define an *extensional typed resource calculus* (Section 5.1), a variant of the usual typed resource calculus with typing rules ensuring that normal terms are in η-long form. We give a few additional constructions for PCG in Section 5.2, before constructing the bijection between normal terms and isogmentations in Section 5.3.

5.1 Extensional simply-typed resource calculus

5.1.1 Typing rules

We start by defining an *extensional* simply-typed resource calculus. Indeed, the existing isomorphism from [40] is between quotiented plays and *normal, η-long resource terms*, because game semantics is inherently extensional. Hence, we set typing rules which ensure normal terms are already in an η-long form.

Reminder: Terms are given by the following grammar (see Definition 1.14):

$$s, t, \ldots ::= x \mid \lambda x.s \mid s\ \bar{t}$$
$$\bar{s}, \bar{t}, \ldots ::= [s_1, \ldots, s_n] .$$

Recall the usual grammar of **types**:

$$A, B, C, \ldots ::= \alpha \mid A \to B$$

with a single **base type** α. If $\vec{A} = \langle A_1, \ldots, A_n \rangle$, we write:

$$\vec{A} \to B \stackrel{\text{def}}{=} A_1 \to \cdots \to A_n \to B = A_1 \to (\cdots \to (A_n \to B)\cdots) .$$

Then any type B can be written uniquely as $B = \vec{A} \to \alpha$.

We fix a type for each variable, so that each type has infinitely many variables, and write $x : A$ when A is the type of x. A **typing context** Γ is a finite set of typed variables, written as an enumeration:

$$\Gamma = x_1 : A_1, \ldots, x_n : A_n$$

and abbreviated as $\vec{x} : \vec{A}$.

$$\frac{\Gamma, x : A \vdash_{\mathsf{Tm}} s : B}{\Gamma \vdash_{\mathsf{Tm}} \lambda x.s : A \to B}\ (\text{abs}) \qquad \frac{\Gamma \vdash_{\mathsf{Tm}} s : A \to B \quad \Gamma \vdash_{\mathsf{Bg}} \bar{t} : A}{\Gamma \vdash_{\mathsf{Tm}} s\ \bar{t} : B}\ (\text{app})$$

$$\frac{\Gamma, x : \vec{A} \to \alpha \vdash_{\mathsf{Sq}} \vec{t} : \vec{A}}{\Gamma, x : \vec{A} \to \alpha \vdash_{\mathsf{Tm}} x\ \vec{t} : \alpha}\ (\text{var})$$

$$\frac{\Gamma \vdash_{\mathsf{Tm}} s_1 : A \quad \cdots \quad \Gamma \vdash_{\mathsf{Tm}} s_n : A}{\Gamma \vdash_{\mathsf{Bg}} [s_1, \ldots, s_n] : A}\ (\text{bag}) \qquad \frac{\Gamma \vdash_{\mathsf{Bg}} \bar{s}_1 : A_1 \quad \cdots \quad \Gamma \vdash_{\mathsf{Bg}} \bar{s}_n : A_n}{\Gamma \vdash_{\mathsf{Sq}} \langle \bar{s}_1, \ldots, \bar{s}_n \rangle : \langle A_1, \ldots, A_n \rangle}\ (\text{seq})$$

Figure 5.1: Typing rules for the simply-typed η-long resource calculus

We may then also write:

$$\lambda \vec{x}.s \stackrel{\text{def}}{=} \lambda x_1. \ldots \lambda x_n.s\,.$$

We call **resource sequence** any sequence $\vec{s} \in \mathcal{S}[\Delta] = \mathcal{M}_f(\Delta)^*$. Given a term s and a resource sequence $\vec{t} = \langle \bar{t}_1, \ldots, \bar{t}_k \rangle$, we also define the application:

$$s\ \vec{t} \stackrel{\text{def}}{=} s\ \bar{t}_1\ \cdots\ \bar{t}_k = (\cdots(s\ \bar{t}_1)\cdots)\ \bar{t}_k\,.$$

Finally, we give the type system in Figure 5.1. There are three different kind of judgements:

- $\Gamma \vdash_{\mathsf{Tm}} s : A$ for terms,
- $\Gamma \vdash_{\mathsf{Bg}} \bar{s} : A$ for bags,
- $\Gamma \vdash_{\mathsf{Sq}} \vec{s} : \vec{A}$ for sequences.

We request bags to be typed uniformly (all the elements of a bag share the same type) and variables to be fully applied.

For $\mathsf{X} \in \{\mathsf{Tm}, \mathsf{Bg}, \mathsf{Sq}\}$, we write $\mathsf{X}(\Gamma; A)$ for the set of expressions s such that $\Gamma \vdash_{\mathsf{X}} s : A$. We extend the type system to finite sums of terms with:

$$\Gamma \vdash_{\mathsf{X}} \sum_{i \in I} s_i : A \quad \text{ if } \Gamma \vdash_{\mathsf{X}} s_i : A \text{ for each } i \in I\,.$$

We write $\Sigma\mathsf{X}(\Gamma; A)$ for $\Sigma[\mathsf{X}(\Gamma; A)]$.

5.1.2 Reduction and substitution

We extend resource substitution to sequences by setting:

$$\langle \bar{s}_1, \ldots, \bar{s}_n \rangle \langle \bar{t}/x \rangle \stackrel{\text{def}}{=} \sum_{\bar{t} \lhd \bar{t}_1 * \cdots * \bar{t}_n} \langle \bar{s}_1 \langle \bar{t}_1/x \rangle, \ldots, \bar{s}_n \langle \bar{t}_n/x \rangle \rangle\,.$$

Reminder: For a bag $\bar{t}$, the sum indexed over $\bar{t} \lhd \bar{t}_1 * \cdots * \bar{t}_n$ is the sum over all n-partitionings of $\bar{t}$ (see subsection 1.3.1).

This implies:

$$(s\ \vec{u})\langle \bar{t}/x \rangle = \sum_{\bar{t} \lhd \bar{t}_1 * \bar{t}_2} (s\langle \bar{t}_1/x \rangle)\,(\vec{u}\langle \bar{t}_2/x \rangle)\,,$$

which generalizes the application case of Definition 1.15.

This type system enjoys subject reduction with respect to $\rightsquigarrow$. As is usual, the key result for subject reduction is a substitution lemma.

Reminder: $\rightsquigarrow$ is the resource reduction, defined in Figure 1.12.

Lemma 5.1 – Substitution

If $s \in \mathsf{X}(\Gamma, x : B; A)$ and $\bar{t} \in \mathsf{Bg}(\Gamma; B)$ then $s\langle \bar{t}/x\rangle \in \Sigma\mathsf{X}(\Gamma; A)$.

Proof. By mutual induction on the three syntactic cases. □

Lemma 5.2 – Subject reduction

If $S \in \Sigma\mathsf{X}(\Gamma; A)$ and $S \rightsquigarrow S'$ then $S' \in \Sigma\mathsf{X}(\Gamma; A)$.

Proof. We first treat the case of $S = s \in \mathsf{X}(\Gamma; A)$ by induction on the definition of the reduction $s \rightsquigarrow S'$: the case of a redex is by the substitution lemma, and the other cases follow by contextuality. The extension to sums is straightforward. □

5.1.3 Normalisation

Remark: The strong normalization result from Chapter 1 (Theorem 1.16) still holds for this typed setting.

For $\mathsf{X} \in \{\mathsf{Tm}, \mathsf{Bg}, \mathsf{Sq}\}$, we write $\mathsf{X}_{\mathsf{nf}}(\Gamma; A)$ for the elements of $\mathsf{X}(\Gamma; A)$ that are in normal form.

Lemma 5.3 – Typing normal forms

We have $s \in \mathsf{X}_{\mathsf{nf}}(\Gamma; A)$ if and only if $\Gamma \vdash_{\mathsf{X}} s : A$ is derivable without using the application rule (app).

Proof. Given a derivation tree for $\Gamma \vdash_{\mathsf{X}} s : A$ using rule (app) at least once, consider a minimal subderivation with this property: it must have an instance of (app) at its root, and its premises are derived without (app). The left premise must thus be derived by (abs): we have ruled out (app), and the conclusion of (var) is never an arrow type. We thus obtain a redex. □

Reminder: A normal resource term s of type $\vec{A} \to \alpha$ is **η-long** if it has the shape

$$\lambda x_1 \ldots \lambda x_{|\vec{A}|}.t$$

with t a normal term of type α (which must then be a fully applied variable).

This property ensures that all normal resource terms are η-long.

Corollary 5.4

Consider $s \in \mathsf{Tm}_{\mathsf{nf}}(\Gamma; A \to B)$, then we can write:

$$s = \lambda x.t \quad \text{with} \begin{cases} x : A\,, \\ t \in \mathsf{Tm}_{\mathsf{nf}}(\Gamma, x : A; B)\,. \end{cases}$$

For $s \in \mathsf{Tm}_{\mathsf{nf}}(\Gamma; \alpha)$, we can write:

$$s = y\ \vec{u} \quad \text{with} \begin{cases} y : \vec{C} \to \alpha \in \Gamma\,, \\ \vec{u} \in \mathsf{Sq}_{\mathsf{nf}}(\Gamma; \vec{C})\,. \end{cases}$$

5.2 A few additional PCG constructions

Before detailing the isomorphism between PCG and Δ, we need a few additional constructions on arenas and configurations.

5.2.1 Construction on arenas – HomGame

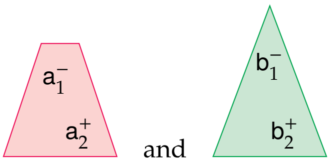


Figure 5.2: Arena A and arena B

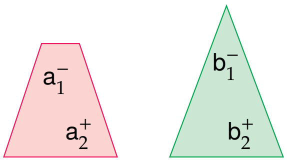


Figure 5.3: Arena $A \otimes B$

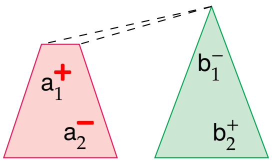


Figure 5.4: Arena $A \Rightarrow B$

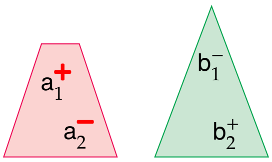


Figure 5.5: Arena $A \vdash B$

In HO games, the categorical structure is obtained *via* the arrow constructor: given two arenas A and B, morphisms from A to B are strategies on the arena $A \Rightarrow B$. However $A \Rightarrow B$ is only defined for B well-opened – otherwise, we lose the tree structure. This arrow construction is needed because strategies can have multiple initial moves: in a non well-opened play, we need the information given by pointers to match moves in A with initial moves in B, and we have to follow pointers in the hiding phase when composing two strategies.

In PCG however, the causal information alone is enough to reconstruct the causal order, and pointers to the initial moves are no longer needed, meaning the composition will be slightly simpler. Hence we introduce the *hom* construction $A \vdash B$, which is very alike $A \Rightarrow B$ except we do not add links between A and B.

Of course, this "simplification" implies some more work when going from PCG to HO games, because we now need to take into account the slightly different categorical structure.

Definition 5.5 – HomGame

Consider arenas A_1 and A_2. Then $A_1 \vdash A_2$ is the arena defined with:

$$
\begin{aligned}
|A_1 \vdash A_2| &\stackrel{\text{def}}{=} |A_1| + |A_2|\,,\\
(i,a) \leq_{A_1 \vdash A_2} (j,b) &\Leftrightarrow (i=j) \text{ and } (a \leq_{A_i} b)\,,\\
\mathrm{pol}_{A_1 \vdash A_2}((1,a)) &= -\,\mathrm{pol}_{A_1}(a)\,,\\
\mathrm{pol}_{A_1 \vdash A_2}((2,a)) &= \mathrm{pol}_{A_2}(a)\,.
\end{aligned}
$$

Then $A_1 \vdash A_2$ is clearly an arena. Remark that if A_1 and A_2 are both negative, then $A_1 \vdash A_2$ is not (because of the minimal events from A_1, which are now positive).

5.2.2 Constructions on configurations

The construction $A_1 \otimes A_2$ can be extended to configurations.

Definition 5.6 – Product of configurations

Consider arenas A_1, A_2 and configurations $x_1 \in \mathrm{Conf}(A_1)$ and $x_2 \in \mathrm{Conf}(A_2)$.

Then $x_1 \otimes x_2$ is the configuration on $A_1 \otimes A_2$ defined with:

$$
\begin{aligned}
|x_1 \otimes x_2| &\stackrel{\text{def}}{=} |x_1| + |x_2|\\
(i,a) \leq_{x_1 \otimes x_2} (j,b) &\Leftrightarrow (i=j) \text{ and } (a \leq_{x_i} b)\\
\partial_{x_1 \otimes x_2}((i,a)) &= (i, \partial_{x_i}(a))\,.
\end{aligned}
$$

Remark that again, this construction extends to the n-ary product in the obvious way.

Likewise, the construction $\mathsf{A}_1 \vdash \mathsf{A}_2$ can be extended to configurations.

Definition 5.7 – Configuration $x_1 \vdash x_2$

Consider arenas A_1, A_2 and configurations $x_1 \in \mathsf{Conf}(\mathsf{A}_1)$ and $x_2 \in \mathsf{Conf}(\mathsf{A}_2)$.

Then $x_1 \vdash x_2$ is the configuration on $\mathsf{A}_1 \vdash \mathsf{A}_2$ defined with:

$$\begin{array}{rcl} |x_1 \vdash x_2| & \stackrel{\text{def}}{=} & |x_1| + |x_2| \\ (i, a) \leq_{x_1 \vdash x_2} (j, b) & \Leftrightarrow & (i = j) \text{ and } (a \leq_{x_i} b) \\ \partial_{x_1 \vdash x_2}((i, a)) & = & (i, \partial_{x_i}(a)) \,. \end{array}$$

One can easily check that $x_1 \vdash x_2 \in \mathsf{Conf}(\mathsf{A}_1 \vdash \mathsf{A}_2)$. Again, remark that $x_1 \vdash x_2$ is no longer negative if x_1, x_2 are negative. Remark that here, both constructions $\otimes$ and $\vdash$ are almost identical: the only difference is the destination arena, where $\mathsf{A}_1 \otimes \mathsf{A}_2$ preserves polarities of both A_1 and A_2 while $\mathsf{A}_1 \vdash \mathsf{A}_2$ inverses polarities for the events occurring in A_1.

Both $\otimes$ and $\vdash$ clearly preserve isomorphisms.

Lemma 5.8 – $\otimes$ and $\vdash$ preserve isomorphisms

Consider arenas A_1, A_2 and configurations $x_i, y_i \in \mathsf{Conf}(\mathsf{A}_i)$ with $x_i \cong y_i$ for $i = 1, 2$. Then,

$$x_1 \otimes x_2 \cong y_1 \otimes y_2 \qquad \text{and} \qquad x_1 \vdash x_2 \cong y_1 \vdash y_2 \,.$$

Proof. Fixing the configuration isomorphisms

$$\varphi_i : x_i \cong y_i \qquad \text{for } i = 1, 2,$$

we construct

$$\begin{array}{rcrcl} \varphi_1 + \varphi_2 & : & |x_1| + |x_2| & \to & |y_1| + |y_2| \\ & & (i, e) & \mapsto & (i, \varphi_i(e)) \end{array}$$

which is a configuration isomorphism for both constructions. □

5.3 The isomorphism

Now we can recast Tsukada and Ong's correspondence as a bijection between normal resource terms in this extensional setting and isogmentations. We first show how the structure of each syntactic kind is reflected by isogmentations of the appropriate type: in particular, terms will be mapped to pointed isogmentations, and bags to general isogmentations.

Reminder: o is the arena with a single (negative) move.

Notation: For any tuple of arenas $\vec{\mathsf{A}} = \langle \mathsf{A}_1, \dots, \mathsf{A}_n \rangle$, we write:

$$\vec{\mathsf{A}} \Rightarrow \mathsf{o} \stackrel{\text{def}}{=} \mathsf{A}_1 \Rightarrow \dots \Rightarrow \mathsf{A}_n \Rightarrow \mathsf{o} \qquad \text{and} \qquad \vec{\mathsf{A}}^{\otimes} \stackrel{\text{def}}{=} \mathsf{A}_1 \otimes \dots \otimes \mathsf{A}_n \,.$$

5.3.1 Types and contexts

We start by giving an interpretation for types:

$$\llbracket \alpha \rrbracket \stackrel{\text{def}}{=} \mathsf{o}$$
$$\llbracket \langle A_1, \ldots, A_n \rangle \rrbracket \stackrel{\text{def}}{=} \llbracket A_1 \rrbracket \otimes \cdots \otimes \llbracket A_n \rrbracket$$
$$\llbracket A \to B \rrbracket \stackrel{\text{def}}{=} \llbracket A \rrbracket \Rightarrow \llbracket B \rrbracket$$

For contexts, we set $\llbracket \Gamma \rrbracket \stackrel{\text{def}}{=} \bigotimes_{(x:A)\in\Gamma} \llbracket A \rrbracket$.

5.3.2 Resource sequences

To reflect the syntactic formation rule for sequences, we show that any isogmentation on an arena $\mathsf{G} \vdash \mathsf{A}_1 \otimes \ldots \otimes \mathsf{A}_n$ can be decomposed in a tuple of isogmentations on the $\mathsf{G} \vdash \mathsf{A}_i$'s – and reciprocally.

Reminder:

$$\frac{\Gamma \vdash_{\mathsf{Bg}} \bar{s}_1 : A_1 \quad \cdots \quad \Gamma \vdash_{\mathsf{Bg}} \bar{s}_n : A_n}{\Gamma \vdash_{\mathsf{Sq}} \langle \bar{s}_1, \ldots, \bar{s}_n \rangle : \langle A_1, \ldots, A_n \rangle}\ (\text{seq})$$

Definition 5.9 – Tupling of augmentations

Consider negative arenas G and $\vec{\mathsf{A}} = \langle \mathsf{A}_1, \ldots, \mathsf{A}_n \rangle$; and augmentations $q_i \in \mathsf{Aug}(\mathsf{G} \vdash \mathsf{A}_i)$ for $1 \leq i \leq n$.

We set $\vec{q} = \langle q_i \mid 1 \leq i \leq n \rangle_{\mathsf{Aug}} \in \mathsf{Aug}(\mathsf{G} \vdash \vec{\mathsf{A}}^{\otimes})$ with

$$|\vec{q}| \stackrel{\text{def}}{=} \sum_{i=1}^{n} |q_i|\,, \qquad \begin{cases} \partial_{\vec{q}}(i,e) \stackrel{\text{def}}{=} (1,\mathsf{g}) & \text{if } \partial_{q_i}(e) = (1,\mathsf{g}),\\ \partial_{\vec{q}}(i,e) \stackrel{\text{def}}{=} (2,(i,\mathsf{a})) & \text{if } \partial_{q_i}(e) = (2,\mathsf{a}),\end{cases}$$

with the two orders $\leq_{\vec{q}}$ and $\leq_{(\!|\vec{q}|\!)}$ inherited.

It is immediate that this construction preserves isomorphisms, so that it extends to isogmentations.

Proposition 5.10 – Tuplings of isogmentations

The previous construction on augmentations induces a bijection

$$\langle -, \ldots, - \rangle_{\mathsf{Isog}} : \prod_{i=1}^{n} \mathsf{Isog}(\mathsf{G} \vdash \mathsf{A}_i) \cong \mathsf{Isog}(\mathsf{G} \vdash \vec{\mathsf{A}}^{\otimes})\,.$$

Proof. ***Injective.*** As an isomorphism must preserve $\twoheadrightarrow$ and display maps, any isomorphism

$$\varphi : \langle q_i \mid i \in I \rangle_{\mathsf{Aug}} \cong \langle p_i \mid i \in I \rangle_{\mathsf{Aug}}$$

decomposes uniquely into a sequence of $\varphi_i : q_i \cong p_i$, as required.

Reminder: An augmentation q is:

- **forestial:** both $\langle |q|, \leq_{(\!|q|\!)} \rangle$ and $\langle |q|, \leq_q \rangle$ are finite forests;
- **negative:** if a is minimal for $\leq_q$, then $\mathsf{pol}(a) = -$.

Surjective. Consider $q \in \mathsf{Aug}(\mathsf{G} \vdash \vec{\mathsf{A}}^{\otimes})$. By *forestiality*, any $a \in |q|$ has a unique minimal antecedent, sent by the display map (via *negativity*) to one of the A_i's – we say that a is *above* A_i. Defining accordingly q_i as q restricted to the events above A_i, we easily construct an isomorphism $q \cong \langle q_i \mid 1 \leq i \leq n \rangle_{\mathsf{Aug}}$ as required. □

5.3.3 Resource bags

The next step is to reflect the typing rule for bags, by showing that isogmentations can be seen as *bags of pointed isogmentations*.

Reminder:

$$\frac{\Gamma \vdash_{\mathsf{Tm}} s_1 : A \quad \cdots \quad \Gamma \vdash_{\mathsf{Tm}} s_n : A}{\Gamma \vdash_{\mathsf{Bg}} [s_1, \ldots, s_n] : A} \text{(bag)}$$

Definition 5.11 – Bag of augmentations

Consider negative arenas G and A, and $q_1, q_2 \in \mathsf{Aug}(\mathsf{G} \vdash \mathsf{A})$.

We set $q_1 * q_2 \in \mathsf{Aug}(\mathsf{G} \vdash \mathsf{A})$ with:

- events $|q_1 * q_2| = |q_1| + |q_2|$,
- display map $\partial_{q_1 * q_2}(i, a) = \partial_{q_i}(a)$,
- the two orders $\leq_{q_1 * q_2}$ and $\leq_{\llparenthesis q_1 * q_2 \rrparenthesis}$ inherited.

This generalizes to an n-ary operation $\Pi_{\mathsf{Aug}}(-)$ in the obvious way, which preserves isomorphisms. The operation induced on isogmentations, denoted by $\Pi_{\mathsf{Isog}}(-)$, is associative and admits as neutral element the empty isogmentation $0 \in \mathsf{Isog}(\mathsf{G} \vdash A)$ with (a unique representative $0 \in \mathsf{Aug}(\mathsf{G} \vdash A)$ with) no event.

Proposition 5.12 – Bags and pointedness

The previous construction on augmentations induces a bijection

$$\Pi_{\mathsf{Isog}}(-) \colon \mathscr{M}_f(\mathsf{Isog}_\bullet(\mathsf{G} \vdash \mathsf{A})) \cong \mathsf{Isog}(\mathsf{G} \vdash \mathsf{A})\,.$$

Proof. ***Injective.*** Consider $q_1, \ldots, q_n, p_1, \ldots, p_m \in \mathsf{Isog}(\mathsf{G} \vdash \mathsf{A})$. Because the q_i's and p_i's are pointed and isomorphisms preserve the forest structure, an isomorphism $\varphi \colon q_1 * \cdots * q_n \cong p_1 * \cdots * p_m$ forces $m = n$ and induces a permutation π on n with a family of isomorphisms $\varphi_i \colon q_i \cong p_{\pi(i)}$ for $1 \leq i \leq n$. This implies:

$$\Pi_{\mathsf{Isog}}[\overline{q_i} \mid 1 \leq i \leq n] = \Pi_{\mathsf{Isog}}[\overline{p_i} \mid 1 \leq i \leq n]\,.$$

Surjective. As any $q \in \mathsf{Aug}(\mathsf{G} \vdash \mathsf{A})$ is finite, it has a finite set I of initial moves. As q is *forestial*, any $a \in |q|$ is above exactly one initial move. For $i \in I$, we write $q_i \in \mathsf{Aug}_\bullet(\mathsf{G} \vdash \mathsf{A})$ the restriction of q above i; then $q \cong q_1 * \cdots * q_n$ as required. □

5.3.4 Currying

Reminder:

$$\frac{\Gamma, x : A \vdash_{\mathsf{Tm}} s : B}{\Gamma \vdash_{\mathsf{Tm}} \lambda x.s : A \to B} \text{(abs)}$$

For the typing rule for abstractions, we need a bijection between augmentations of $\mathsf{G} \otimes \mathsf{A} \vdash \mathsf{B}$ and augmentations of $\mathsf{G} \vdash \mathsf{A} \Rightarrow \mathsf{B}$. These two arenas are *almost* identical; the events are the same (up to the tags), but $\mathsf{G} \vdash \mathsf{A} \Rightarrow \mathsf{B}$ adds links between events of A and events of B. Thankfully, given an augmentation $q \in \mathsf{Aug}(\mathsf{G} \otimes \mathsf{A} \vdash \mathsf{B})$, the forestial structure of q ensures that these links can be uniquely constructed when turning q into an augmentation of $\mathsf{G} \vdash \mathsf{A} \Rightarrow \mathsf{B}$.

Lemma 5.13 – Unique initial ancestor

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$. For every $a \in |q|$, there exists a unique $b \in \min_{\leq_q}(q)$ such that $b \leq_q a$.

This event b is called the **initial ancestor of** a, denoted by $\mathsf{init}(a)$.

Proof. $\leq_q$ is a finitary forest. □

Definition 5.14 – Currying of augmentation

Consider negative arenas G, A and B. We have a bijection

$$\Lambda^{\mathsf{Aug}}_{\mathsf{G},\mathsf{A},\mathsf{B}} : \mathsf{Aug}(\mathsf{G} \otimes \mathsf{A} \vdash \mathsf{B}) \cong \mathsf{Aug}(\mathsf{G} \vdash \mathsf{A} \Rightarrow \mathsf{B})$$

leaving the augmentation unchanged except for the display map, which is reassigned following:

$$\begin{array}{rcll} \partial_{\Lambda(q)}(a) & = & (1, \mathsf{b}) & \text{if } \partial_q(a) = (1,(1,\mathsf{b})) \\ \partial_{\Lambda(q)}(a) & = & (2,(2,\mathsf{b})) & \text{if } \partial_q(a) = (2,\mathsf{b}) \\ \partial_{\Lambda(q)}(a) & = & (2,(1,(\mathsf{b},\mathsf{c}))) & \text{if } \partial_q(a) = (1,(2,\mathsf{c})) \\ & & & \text{and } \partial_q(\mathsf{init}(a)) = (2,\mathsf{b}); \end{array}$$

and for the static order, which is likewise completed with static links between events in A and B:

$$\begin{array}{rclll} a' & \leq_{(\!|\Lambda(q)|\!)} & a & \text{iff} & (a' \leq_{(\!|q|\!)} a) \\ & & & & \text{or } \big(\partial_q(a) = (1,(2,\mathsf{a})) \text{ and } a' = \mathsf{init}(a)\big)\,. \end{array}$$

Remark: If $\partial_q(a) = (1,(2,\mathsf{a}))$, then $\partial_q(\mathsf{init}(a)) = (2,\mathsf{b})$ by negativity of q.

Since isomorphisms of augmentations are order-isomorphisms and preserve display maps, the definition of this bijection is obviously compatible with isomorphisms. Moreover, the causal order is unchanged, so in particular it preserves well-openedness.

Proposition 5.15 – Currying of isogmentations

The previous bijection defined on augmentations induces bijections:

$$\Lambda^{\mathsf{Isog}}_{\Gamma,\mathsf{A},\mathsf{B}} : \mathsf{Isog}(\Gamma \otimes \mathsf{A} \vdash \mathsf{B}) \cong \mathsf{Isog}(\Gamma \vdash \mathsf{A} \Rightarrow \mathsf{B})$$

$$\Lambda^{\mathsf{Isog}_\bullet}_{\Gamma,\mathsf{A},\mathsf{B}} : \mathsf{Isog}_\bullet(\Gamma \otimes \mathsf{A} \vdash \mathsf{B}) \cong \mathsf{Isog}_\bullet(\Gamma \vdash \mathsf{A} \Rightarrow \mathsf{B})$$

5.3.5 Head occurrence

Reminder:

$$\frac{\Gamma, x : \vec{A} \to \alpha \vdash_{\mathsf{Sq}} \vec{t} : \vec{A}}{\Gamma, x : \vec{A} \to \alpha \vdash_{\mathsf{Tm}} x\,\vec{t} : \alpha}\ (\text{var})$$

Reminder: o is the arena with exactly one negative move, written $\mathbf{q}^-$.

By Lemma 5.3, the only remaining case for typed normal forms is (var).

Consider $\mathsf{G} = \mathsf{A}_1 \otimes \ldots \otimes \mathsf{A}_n$, where each A_i is a negative arena of the shape $\mathsf{A}_i = \vec{\mathsf{B}_i} \Rightarrow \mathsf{o} \cong \vec{\mathsf{B}}_i^{\otimes} \Rightarrow \mathsf{o}$ with $\vec{\mathsf{B}_i} = \langle \mathsf{B}_{i,1}, \ldots, \mathsf{B}_{i,p_i} \rangle$. Given an augmentation $q \in \mathsf{Aug}(\mathsf{G} \vdash \vec{B_i}^{\otimes})$, we construct **the i-lifting** of q:

$$\Box_i(q) \in \mathsf{Aug}_\bullet(\mathsf{G} \vdash \mathsf{o})\,.$$

Intuitively, $\square_i(q)$ is the augmentation which:

- starts by $\mathbf{q}^-$ the initial Opponent move of $\mathsf{G} \vdash \mathsf{o}$,
- plays $\mathbf{q}_i^+$ the initial move from A_i (which is negative in A_i by negativity, thus positive in $\mathsf{G} \vdash \mathsf{o}$),
- then proceeds as q: writing $a_1^-, \dots, a_k^-$ the minimal moves of q, they must be mapped to initial events in $\mathsf{G} \vdash \vec{\mathsf{B}}_i^{\otimes}$, hence to initial events of $\vec{\mathsf{B}}_i^{\otimes}$ by negativity of q; so they can be played in the i-th component of G.

We obtain a pointed augmentation on $\mathsf{G} \vdash \mathsf{o}$, depicted in Figure 5.6.

More formally:

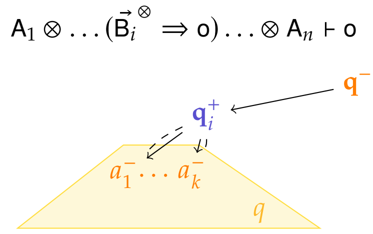


Figure 5.6: $\square_i(q)$.

Definition 5.16 – i-lifting of an augmentation

Consider $\mathsf{G} = \mathsf{A}_1 \otimes \dots \otimes \mathsf{A}_n$, where each A_i is a negative arena of the shape $\mathsf{A}_i = \vec{\mathsf{B}}_i \Rightarrow \mathsf{o} \cong \vec{\mathsf{B}}_i^{\otimes} \Rightarrow \mathsf{o}$ with $\vec{\mathsf{B}}_i = \langle \mathsf{B}_{i,1}, \dots, \mathsf{B}_{i,p_i} \rangle$.

Consider also an augmentation $q \in \mathsf{Aug}(\mathsf{G} \vdash \vec{B}_i^{\otimes})$.

The **i-lifting of** q, written $\square_i(q) \in \mathsf{Aug}_{\bullet}(\mathsf{G} \vdash \mathsf{o})$, is defined with:

- events $|\square_i(q)| \overset{\text{def}}{=} |q| \uplus \{\ominus, \oplus\}$,
- static order $\leq_{(\!|\square_i(q)|\!)}$ the least partial order containing:

$$\begin{array}{lcll} (1,a) & \leq_{(\!|\square_i(q)|\!)} & (1,b) & \text{if } a \leq_{(\!|q|\!)} b, \\ (2,\oplus) & \leq_{(\!|\square_i(q)|\!)} & (1,a) & \text{if } \partial_q(a) = (2,\mathsf{b}), \end{array}$$

- causality order $\leq_{\square_i(q)}$ the order $\leq_q$ prefixed with $\ominus \twoheadrightarrow \oplus$, *i.e.* for all $a \leq_q b \in |q|$ and $e \in \{\oplus, \ominus\}$,

$$(1,a) \leq_{\square_i(q)} (1,b)\,; \quad (2,e) \leq_{\square_i(q)} (1,a)\,; \quad (2,\ominus) <_{\square_i(q)} (2,\oplus)\,,$$

- display map $\partial_{\square_i(q)}$ the map given by:

$$\begin{array}{lcll} \partial_{\square_i(q)}((2,\ominus)) & \overset{\text{def}}{=} & (2,\mathbf{q}), & \\ \partial_{\square_i(q)}((2,\oplus)) & \overset{\text{def}}{=} & (1,(i,(2,\mathbf{q}))), & \\ \partial_{\square_i(q)}((1,a)) & \overset{\text{def}}{=} & (1,\mathsf{a}) & \text{if } \partial_q(a) = (1,\mathsf{a}), \\ \partial_{\square_i(q)}((1,b)) & \overset{\text{def}}{=} & (1,(i,(1,\mathsf{b}))) & \text{if } \partial_q(b) = (2,\mathsf{b}). \end{array}$$

Recall that $\mathbf{q}$ is the only event of o.
We keep the isomorphism $\mathsf{A}_i \cong \vec{\mathsf{B}}_i^{\otimes} \Rightarrow \mathsf{o}$ implicit whenever we write moves in A_i, so that we can consider $(i,(2,\mathbf{q})) \in |\mathsf{G}|$, and $(i,(1,\mathsf{b})) \in |\mathsf{G}|$ for any $\mathsf{b} \in |\vec{\mathsf{B}}_i^{\otimes}|$.

This construction defines an augmentation on $\mathsf{G} \vdash \mathsf{o}$, with a unique initial event $(2,\ominus)$. It again preserves isomorphisms, thus extending to isogmentations.

Proposition 5.17 – Lifting of isogmentations

The previous construction on augmentations induces a bijection

$$\square_{-}(-) \colon \{1,\dots,n\} \times \sum_{1 \leq i \leq n} \mathsf{Isog}(\mathsf{G} \vdash \vec{\mathsf{B}}_i^{\otimes}) \cong \mathsf{Isog}_{\bullet}(\mathsf{G} \vdash \mathsf{o})\,.$$

Proof. ***Injective.*** Given isomorphic $\square_i(q')$ and $\square_j(p')$, we obviously have $i = j$ since isomorphisms of augmentations preserve display maps; and the isomorphism decomposes into $q' \cong p'$ as required.

$$\begin{aligned}
\|\Gamma \vdash_{\mathsf{Tm}} \lambda x.s : A \to B\|_{\mathsf{Tm}} &\overset{\text{def}}{=} \Lambda^{\mathsf{Isog}_\bullet}_{[\![\Gamma]\!],[\![A]\!],[\![B]\!]}(\|\Gamma, x : A \vdash_{\mathsf{Tm}} s : B\|_{\mathsf{Tm}}) \\
\|\Gamma \vdash_{\mathsf{Tm}} x\,\vec{t} : \alpha\|_{\mathsf{Tm}} &\overset{\text{def}}{=} \square_i(\|\Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{A}\|_{\mathsf{Sq}}) \\
\|\Gamma \vdash_{\mathsf{Bg}} [s_1, \ldots, s_n] : A\|_{\mathsf{Bg}} &\overset{\text{def}}{=} \Pi_{\mathsf{Isog}}[\ \|\Gamma \vdash_{\mathsf{Tm}} s_i : A\|_{\mathsf{Tm}} \mid 1 \leq i \leq n\ ] \\
\|\Gamma \vdash_{\mathsf{Sq}} \langle \bar{s}_1, \ldots, \bar{s}_n\rangle : \vec{A}\|_{\mathsf{Sq}} &\overset{\text{def}}{=} \langle\ \|\Gamma \vdash_{\mathsf{Bg}} \bar{s}_i : A_i\|_{\mathsf{Bg}} \mid 1 \leq i \leq n\ \rangle_{\mathsf{Isog}}
\end{aligned}$$

Figure 5.7: Isomorphism for normal forms of the resource calculus

Surjective. Any $q \in \mathsf{Isog}_\bullet(\mathsf{G} \vdash \mathsf{o})$ has a unique initial move, which is negative hence cannot be maximal by +-*covered*. By *determinism*, there is a unique subsequent Player move, displayed to the initial move of some A_i. The subsequent moves directly inform $q' \in \mathsf{Aug}(\mathsf{A} \vdash \vec{\mathsf{B}}_i^{\otimes})$ such that $q \cong \square_i(q')$. □

5.3.6 The isomorphism

Putting together the above results, we may now deduce:

Theorem 5.18 – Bijection (for typed resource calculus)

For Γ a context and A a type, there are bijections:

$$\begin{aligned}
\|-\|_{\mathsf{Tm}} &: \mathsf{Tm}_{\mathsf{nf}}(\Gamma; A) \cong \mathsf{Isog}_\bullet([\![\Gamma]\!] \vdash [\![A]\!]) \\
\|-\|_{\mathsf{Bg}} &: \mathsf{Bg}_{\mathsf{nf}}(\Gamma; A) \cong \mathsf{Isog}([\![\Gamma]\!] \vdash [\![A]\!]) \\
\|-\|_{\mathsf{Sq}} &: \mathsf{Sq}_{\mathsf{nf}}(\Gamma; \vec{A}) \cong \mathsf{Isog}([\![\Gamma]\!] \vdash [\![\vec{A}]\!])\,.
\end{aligned}$$

Proof. The three functions are defined by mutual induction using Propositions 5.10, 5.12, 5.15 and 5.17, as in Figure 5.7 (where the index i for the head variable case is the index of $(x : \vec{A} \to \alpha)$ in Γ).

Injectivity. Directly by induction on the syntax, using the injectivity of each construction.

Surjectivity. By induction on the size (*i.e.* the number of events) of augmentations, the syntactic kind (considering $\mathsf{Tm} < \mathsf{Bg} < \mathsf{Sq}$), and also the type A in the Tm case:

- the decomposition provided by Proposition 5.17 yields augmentations of strictly smaller size (we remove the two initial moves);
- the bijection of Proposition 5.15 preserves the size of augmentations and stays in the kind Tm, but yields a smaller output type;
- the remaining two decompositions do not increase the size and yield a lower kind. □

Hence we have an explicit bijection $\|-\|_{\mathsf{Tm}}$ between normal resource terms and isogmentations of PCG.

Example: Consider the following sequent:

$$x : \alpha \vdash_{\mathsf{Tm}} \lambda y.y\,[x, x]\,[] : (\alpha \to (\alpha \to \alpha) \to \alpha) \to \alpha$$

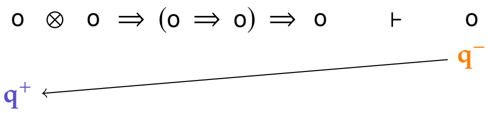
with the typing derivation (setting $\Gamma := x : \alpha, y : \alpha \to (\alpha \to \alpha) \to \alpha$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{\Gamma \vdash_{\mathsf{Tm}} x : \alpha}\,(\text{var}) \quad \overline{\Gamma \vdash_{\mathsf{Tm}} x : \alpha}\,(\text{var})}{\Gamma \vdash_{\mathsf{Bg}} [x, x] : \alpha}\,(\text{bag}) \qquad \overline{\Gamma \vdash_{\mathsf{Bg}} [] : \alpha \to \alpha}\,(\text{bag})}{\Gamma \vdash_{\mathsf{Sq}} \langle [x, x], [] \rangle : \langle \alpha, \alpha \to \alpha \rangle}\,(\text{seq})}{x : \alpha, y : \alpha \to (\alpha \to \alpha) \to \alpha \vdash_{\mathsf{Tm}} y\, [x, x]\, [] : \alpha}\,(\text{var})}{x : \alpha \vdash_{\mathsf{Tm}} \lambda y. y\, [x, x]\, [] : (\alpha \to (\alpha \to \alpha) \to \alpha) \to \alpha}\,(\text{abs})$$

We construct its interpretation step-by-step.

Typing rule (var) with a head variable of type α. Since x is applied to an empty sequence, we are in a special case of i-lifting:

$\|\Gamma \vdash_{\mathsf{Tm}} x : \alpha\|_{\mathsf{Tm}} =$

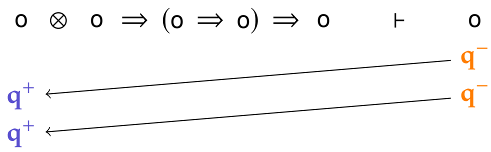


Typing rules (bag). For the empty bag, we simply have:

$$\|\Gamma \vdash_{\mathsf{Bg}} [] : \alpha \to \alpha\|_{\mathsf{Bg}} = 0 \in \mathsf{Isog}([\![\Gamma]\!] \vdash o \Rightarrow o)\,.$$

For the other one, we have:

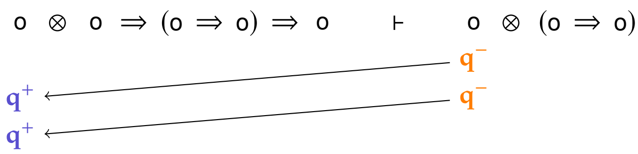
$\|\Gamma \vdash_{\mathsf{Bg}} [x, x] : \alpha\|_{\mathsf{Tm}} =$



Typing rule (seq). From the tupling isomorphism, we have

$$\begin{aligned} & \|\Gamma \vdash_{\mathsf{Sq}} \langle [x, x], [] \rangle : \langle \alpha, \alpha \to \alpha \rangle\|_{\mathsf{Sq}} \\ = \; & \langle\, \|\Gamma \vdash_{\mathsf{Bg}} [x, x] : \alpha\|_{\mathsf{Bg}}, \|\Gamma \vdash_{\mathsf{Bg}} [] : \alpha \to \alpha\|_{\mathsf{Bg}} \rangle_{\mathsf{Isog}} \end{aligned}$$

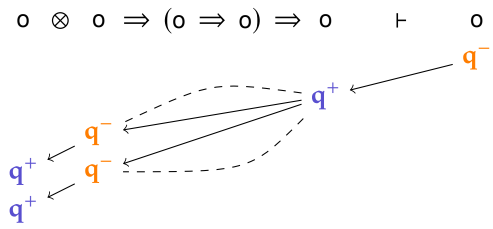
which gives us the following isogmentation:



Typing rule (var). We lift the previous isogmentation:

$$\begin{aligned} & \|\Gamma \vdash_{\mathsf{Tm}} y\, [x, x]\, [] : \alpha\|_{\mathsf{Tm}} \\ = \; & \square_2(\|\Gamma \vdash_{\mathsf{Sq}} \langle [x, x], [] \rangle : \langle \alpha, \alpha \to \alpha \rangle\|_{\mathsf{Sq}}) \end{aligned}$$

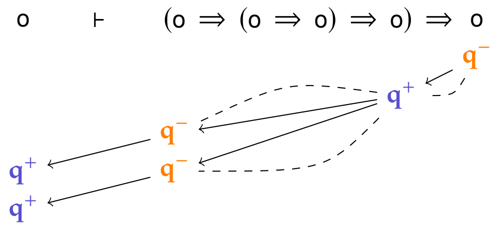
which gives us:



Typing rule (abs). Finally, we apply the currying isomorphism:

$$\begin{aligned} & \|x : \alpha \vdash_{\mathsf{Tm}} \lambda y.y\,[x,x]\,[] : (\alpha \to (\alpha \to \alpha) \to \alpha) \to \alpha\|_{\mathsf{Tm}} \\ = \;& \Lambda^{\mathsf{Isog}}_{o,o\Rightarrow(o\Rightarrow o)\Rightarrow o,o}(\|\Gamma \vdash_{\mathsf{Tm}} y\,[x,x]\,[] : \alpha\|_{\mathsf{Tm}}) \end{aligned}$$

and we obtain:



5.4 Conclusion

Hence, we have a direct isomorphism between normal resource terms and isogmentations. However, we can do better: in the next chapters, we study the categorical structure of PCG, in order to extend this first correspondence into a sound interpretation of resource terms in PCG.

Composition and Categorical Structure 6

Now that we have established a first link between the resource calculus and pointer concurrent games, we expand our game model with the notion of *composition*. Indeed, one of the advantages of game semantics is its compositional aspect, representing how programs interact which each other.

Defining composition for augmentations is tricky, because both augmentations need to agree on the events occurring in the shared arena component; thus, Section 6.1 features a detailed presentation of the construction. In doing so, we find out that the composition of two augmentations must (in general) produce *several* augmentations – that is, we need to consider sums of augmentations rather than single augmentations. This leads us to the definition of PCG *strategies*, which are sums of augmentations (similar to how HO strategies are sets of plays). Section 6.2 defines strategies and in particular copycat strategies, which will be the identity morphisms for PCG – the SMCC obtained with negative arenas as objects and strategies as morphisms, presented in Section 6.4. Finally, we show in Section 6.5 that our notion of composition is compatible with the composition in HO games.

6.1 Composition for augmentations

In all this section, we fix A, B and C negative arenas. We start by defining the *composition* of augmentations: how do two augmentations – say, the ones from Figures 6.1 and 6.2 – interact with each other?

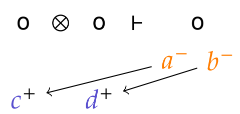


Figure 6.1: $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, with $\mathsf{A} = o \otimes o$ and $\mathsf{B} = o$.

As for HO games, we first define the *interaction* of two augmentations, then we hide the events occurring in the shared arena to obtain the composition. However, because PCG augmentations are not linear, there may be several ways to "match" events occurring in the shared arena component – hence we need to first fix an isomorphism between those events.

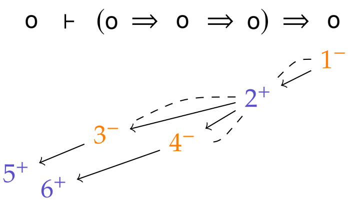


Figure 6.2: $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, with $\mathsf{B} = o$ and $\mathsf{C} = (o \Rightarrow o \Rightarrow o) \Rightarrow o$.

6.1.1 Interaction via an isomorphism

Consider two augmentations $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ and $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$. Inuitively, we can only compose q and p provided "they reach the same state on B", so we first extract the "state they reach" via their desequentializations: let us write $|x_{q \upharpoonright \mathsf{A}}|$ for the events of q that display to A and $|x_{q \upharpoonright \mathsf{B}}|$ for those that display to B – these inform $x_{q \upharpoonright \mathsf{A}} \in \mathsf{Conf}(\mathsf{A})$ and $x_{q \upharpoonright \mathsf{B}} \in \mathsf{Conf}(\mathsf{B})$ and likewise for p.

Remark: Up to isomorphism of augmentations, we may consider

$$(\!|q|\!) = x_{q \upharpoonright \mathsf{A}} \vdash x_{q \upharpoonright \mathsf{B}}\,,$$

but we need not assume that.

But what does it mean to "reach the same state"? In general requiring $x_{q \upharpoonright \mathsf{B}} = x_{p \upharpoonright \mathsf{B}}$ is meaningless, since this data should really be considered up to isomorphism. States in B are not configurations, but *positions* – symmetry classes of configurations. Thus q and p are **compatible** if $x_{q \upharpoonright \mathsf{B}}$ and $x_{p \upharpoonright \mathsf{B}}$ are **symmetric**, *i.e.* if there is a symmetry $\varphi : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$.

Notation: we write $x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$ for the induced equivalence.

For instance, in Figures 6.1 and 6.2, we have

$$|x_{\mathsf{B}\upharpoonright q}| = \{a, b\} \qquad \text{and} \qquad |x_{\mathsf{B}\upharpoonright q}| = \{5, 6\},$$

where all four events maps to the same arena move $\mathbf{q}^-$ (the only move of the singleton arena $\mathsf{B} = \mathsf{o}$). Hence, we have two symmetries:

$$\varphi = \{(a \mapsto 5), (b \mapsto 6)\} \qquad \text{and} \qquad \psi = \{(a \mapsto 6), (b \mapsto 5)\}\,.$$

Composing q and p means constructing an augmentation on $\mathsf{A} \vdash \mathsf{C}$, resulting from the interaction of q and p. However, the behavior of this interaction depends on the choice of symmetry – actually, we shall see that different symmetries may lead to different augmentations! Hence, we start by defining the interaction of two compatible augmentations *along with a mediating symmetry*.

Definition 6.1 – Interaction via a symmetry

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ and $\varphi : x_{q\upharpoonright\mathsf{B}} \cong_{\mathsf{B}} x_{p\upharpoonright\mathsf{B}}$.

The **interaction** $p \circledast_\varphi q$ is the pair $\langle |p \circledast_\varphi q|, \leq_{p\circledast_\varphi q}\rangle$ with:

- the set $|p \circledast_\varphi q| \overset{\text{def}}{=} |q| + |p|$,
- the binary relation $\leq_{p\circledast_\varphi q}$ on $|p \circledast_\varphi q|$ defined as the transitive closure of $\rhd \overset{\text{def}}{=} \rhd_q \cup \rhd_p \cup \rhd_\varphi$,

with

$$\begin{array}{rcl} \rhd_q & = & \{((1,e),(1,f)) \mid e <_q f\}\,, \\ \rhd_p & = & \{((2,e),(2,f)) \mid e <_p f\}\,, \\ \rhd_\varphi & = & \{((1,e),(2,\varphi(e))) \mid e \in |x_{q\upharpoonright\mathsf{B}}| \wedge \mathsf{pol}_{A\vdash B}(\partial_q(e)) = +\} \\ & \cup & \{((2,e),(1,\varphi^{-1}(e))) \mid e \in |x_{p\upharpoonright\mathsf{B}}| \wedge \mathsf{pol}_{B\vdash C}(\partial_p(e)) = +\}\,. \end{array}$$

For instance, we can construct $p \circledast_\varphi q$ with the augmentations and symmetry from above:

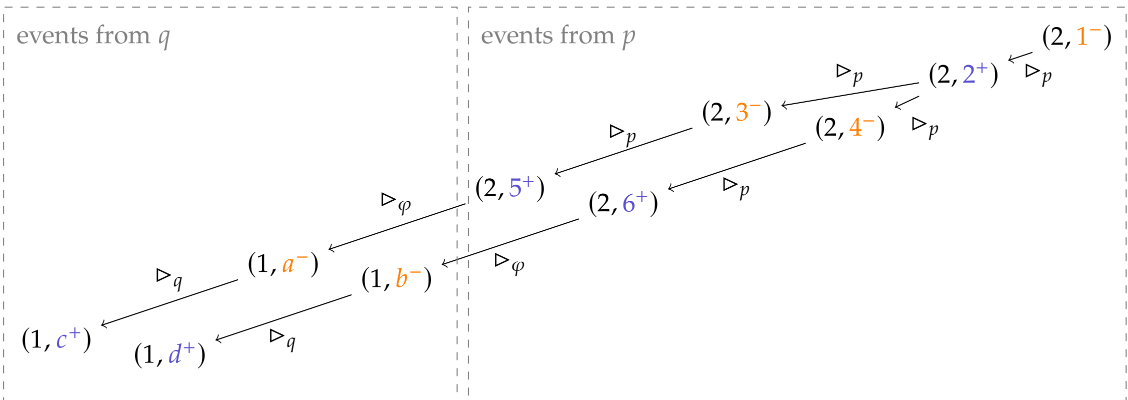


Figure 6.3: $p \circledast_\varphi q$, where we annotate each immediate causality arrow with the relation it comes from (between $\rhd_q$, $\rhd_p$ and $\rhd_\varphi$).

Now, before going any further, we need to check that $\rhd$ is acyclic: remember that we want to hide events of the *interaction* occurring in B to obtain an *augmentation* on $\mathsf{A} \vdash \mathsf{C}$, with a partial order.

Remark: for $e \in |p \circledast_\varphi q|$, then e **occurs in** A if and only if it has form $(1, e')$ with $\partial_q(e') = (1, \mathsf{a})$; and e **occurs in** C if and only if it has form $(2, e')$ with $\partial_p(e') = (2, \mathsf{c})$. Otherwise, it **occurs in** B.

Proof sketch: We start by proving that if $\rhd$ has a cycle, then it has a cycle occurring entirely in B (Lemma 6.2), without minimal events in B (Lemma 6.4). Focusing on such a cycle, we exhibit a contradiction in Lemma 6.6. The proof is a direct adaptation of a similar fact in concurrent games on event structures [12, Lemma 7.6].

[12]: Castellan and Clairambault (2021), *Disentangling Parallelism and Interference in Game Semantics*

Lemma 6.2 – A cycle must occur in B

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ and $\varphi : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$.

If $\rhd$ has a cycle in $p \circledast_\varphi q$, then it has a cycle entirely in B.

Proof. First, observe that $\rhd$ has no direct link between A and C. Consider a cycle

$$e_1 \rhd \ldots \rhd e_n \rhd e_1\,.$$

Note that this cycle must pass through B; otherwise, it is entirely in A or entirely in C, making $\rhd_q$ or $\rhd_p$ cyclic, contradiction.

Now, consider a section

$$e_i^{\mathsf{B}} \rhd e_{i+1}^{\mathsf{A}} \rhd \ldots e_j^{\mathsf{A}} \rhd e_{j+1}^{\mathsf{B}}$$

Notation: We write e^{A} (resp. e^{B}, e^{C}) if the event e occurs in A (resp. B, C).

with the segment $e_{i+1} \rhd \ldots \rhd e_j$ entirely in A. By definition, we must have

$$e_i^{\mathsf{B}} \rhd_q e_{i+1}^{\mathsf{A}} \rhd_q \ldots \rhd_q e_j^{\mathsf{A}} \rhd_q e_{j+1}^{\mathsf{B}}\,,$$

so that $e_i^{\mathsf{B}} \rhd_q e_{j+1}^{\mathsf{B}}$ by transitivity of $\rhd_q$. Hence, a segment of the cycle in A may be removed, preserving the cycle. Symmetrically, any segment in C may be removed, yielding a cycle within B. □

Hence, we restrict our attention to cycles entirely within B. Given

$$e_1 \rhd \ldots \rhd e_n \rhd e_1$$

a cycle, we call n its **length**. We show that this cycle can also be assumed not to contain any element minimal in B.

First, remark that if e is minimal in B, then any event greater than e for the causal order is also greater for the static order.

Lemma 6.3 – Minimality in B

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ and $e, e' \in |q|$ such that $e <_q e'$, with e' occurring in B and $\partial_q(e)$ minimal in B. Then, $e <_{(\!|q|\!)} e'$.

Proof. Since $(\!|q|\!)$ is a forest, there is a unique $f \leq_{(\!|q|\!)} e'$ such that f is minimal for $\leq_{(\!|q|\!)}$. By *minimality-respecting* of $(\!|q|\!)$, $\partial_q(f)$ is minimal in $\mathsf{A} \vdash \mathsf{B}$. By construction, f must occur in B, so by negativity of B, f is negative. Hence by *rule-abidingness* and *courtesy* of q, f is minimal for $\leq_q$. Since q is a forest, it follows that $f = e$. □

Reminder:
minimality-respecting:

$$e \in \min(\leq_{(\!|q|\!)}) \Leftrightarrow \partial_q(e) \in \min(\leq_{\mathsf{A} \vdash \mathsf{B}})\,;$$

rule-abidingness:

if $a \leq_{(\!|q|\!)} b$, then $a \leq_q b$;

courtesy:

$$(e^+ \rightarrow_q f \text{ or } e \rightarrow_q f^-) \Rightarrow e \rightarrow_{(\!|q|\!)} f\,.$$

Exploiting that, we prove that if a cycle exists, then there is a cycle without any minimal event in B.

Lemma 6.4 – Cycle without minimal event

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ and $\varphi : x_{q\upharpoonright\mathsf{B}} \cong_{\mathsf{B}} x_{p\upharpoonright\mathsf{B}}$. If $\rhd$ has a cycle, then it has one entirely in B and without minimal event in B.

Proof. By Lemma 6.2, assume the cycle is entirely in B.

Consider a cycle

$$e_1 \rhd \ldots \rhd e_n \rhd e_1$$

entirely in B of minimal length. Seeking a contradiction, consider e_i minimal in B. Assume first it is in B^r. Since B is negative, we cannot have $e_{i-1} \rhd_\varphi e_i$. Hence, if e_{i+1} is also in B^r, we have

Notation: if e occurs in B, we say it **occurs in** B^1 if it has the form $(1, e')$ for $e' \in |q|$, and **occurs in** B^r otherwise.

$$e_{i-1} \rhd_p e_i \rhd_p e_{i+1}$$

so $e_{i-1} \rhd_p e_{i+1}$, shortening the cycle and contradicting its minimality. So e_{i+1} is in B^1. But then $e_i \rhd_\varphi e_{i+1}$, and e_i is in B^r, so we have $e_i = (2, f_i)$ with $\mathsf{pol}_{\mathsf{B}\vdash\mathsf{C}}(\partial_p(f_i)) = +$. Hence $e_i \rhd_\varphi e_{i+1}$ implies $e_{i+1} = (1, \varphi^{-1}(f_i))$. As an isomorphism of configurations, φ preserves the display to B, so e_{i+1} is minimal in B^1.

In all cases, the cycle contains a minimal element of B^1. Call it e_i, then

$$e_i \rhd_q e_{i+1} \rhd_q \ldots \rhd_q e_j \rhd_\varphi e_{j+1}$$

where all relations in between e_i and e_j are in $\rhd_q$ (by definition, only those can apply until we jump to B^r via $\rhd_\varphi$), and where by definition, $\varphi(e_j) = e_{j+1}$. By transitivity, $e_i \rhd_q e_j$. But by Lemma 6.3, this entails $e_i <_{(\!|p \circledast_\varphi q|\!)} e_j$. Now as φ is an isomorphism of configurations, this implies $\varphi(e_i) <_{(\!|p \circledast_\varphi q|\!)} \varphi(e_j)$, hence $e_{i-1} <_{(\!|p \circledast_\varphi q|\!)} e_{j+1}$. By *rule-abiding*, this entails $e_{i-1} \rhd_p e_{j+1}$. But this means that the segment $e_i \ldots e_j$ may be removed from the cycle, contradicting the minimality of the later. □

Notation: We define $\leq_{(\!|p \circledast_\varphi q|\!)}$ with

$$\begin{array}{lll} (1, e) \leq_{(\!|p \circledast_\varphi q|\!)} (1, f) & \text{iff} & e \leq_{(\!|q|\!)} f\,, \\ (2, e) \leq_{(\!|p \circledast_\varphi q|\!)} (2, f) & \text{iff} & e \leq_{(\!|p|\!)} f\,. \end{array}$$

So we focus on cycles entirely in B, comprising no minimal event.

Notation: If $e \in |p \circledast_\varphi q|$ is not minimal in $(\!|p \circledast_\varphi q|\!)$, it has a unique predecessor in $(\!|p \circledast_\varphi q|\!)$ called its **justifier** and written $\mathsf{just}(e)$.

Considering whether an event occurs in B^1 or in B^r is not precise enough: we also need to consider its polarity, since $\rhd_\varphi$ is defined taking into account both the arena side and the polarity of events.

Notation: We say e occurring in B **has polarity** $\mathtt{l}$ if it has the form $(1, e')$ with $\mathsf{pol}_{\mathsf{A}\vdash\mathsf{B}}(\partial_q(e')) = +$, **has polarity** $\mathtt{r}$ if it has the form $(2, e')$ with $\mathsf{pol}_{\mathsf{B}\vdash\mathsf{C}}(\partial_p(e')) = +$, and **has polarity** φ otherwise. We may then write e^1, e^r or e^φ instead of e, depending on its polarity.

Lemma 6.5 – Deadlock-free auxiliary lemma

We have the following properties:

$$\begin{array}{ll} (1) & \text{if } e \rhd_q f^{\varphi}, \text{ then } e \rhd_q^* \mathsf{just}(f), \\ (2) & \text{if } e \rhd_p f^{\varphi}, \text{ then } e \rhd_p^* \mathsf{just}(f), \end{array}$$

where the events are annotated with their assumed polarity.

Proof. ***(1)*** We must have $e = (1, e')$ and $f = (1, f')$ with $e' <_q f'$, with f' negative. By *rule-abiding* and *courtesy*, $\mathsf{just}(f') \rightarrowtriangle_q f'$. Since q is a forest, $e' \leq_q \mathsf{just}(f')$.

(2) Symmetric. □

Notation: For any $e \in |p \circledast_\varphi q|$, its **depth**, written $\mathsf{depth}(e)$, is 0 if e is minimal in $(\!|p \circledast_\varphi q|\!)$, otherwise $\mathsf{depth}(\mathsf{just}(e)) + 1$.

We finally prove the *deadlock-free lemma*:

Lemma 6.6 – Deadlock-free lemma

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ and $\varphi : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$.

Then $\rhd$ is acyclic.

Proof. Seeking a contradiction, assume there is a cycle. By Lemma 6.4 it is entirely in B, without a minimal event in B. Writing it $\rho = e_1 \rhd \ldots \rhd e_n \rhd e_1$, its **depth** is

$$\mathsf{depth}(\rho) = \sum_{i=1}^{n} \mathsf{depth}(e_i)\,,$$

and *w.l.o.g.* assume ρ minimal for the product order on pairs (n, d) where $d = \mathsf{depth}(\rho)$ and n is its length. We notice that ρ has no consecutive $\rhd_q$ or $\rhd_p$ – or we shorten the cycle by transitivity, breaking minimality. It also has no consecutive $\rhd_\varphi$ by definition. This entails that $n = 4k$, with *w.l.o.g.*

$$e_{4i} \rhd_q e_{4i+1} \rhd_\varphi e_{4i+2} \rhd_p e_{4i+3} \rhd_\varphi e_{4i+4}\,.$$

Then for all i, e_{4i+1} has polarity $\mathtt{l}$. Otherwise, it has polarity φ, making $e_{4i+1} \rhd_\varphi e_{4i+2}$ impossible. Likewise, e_{4i+3} has polarity $\mathtt{r}$, while e_{4i+2} and e_{4i+4} have polarity φ.

We claim $\mathsf{just}(e_{4i+1}) \rhd_q \mathsf{just}(e_{4i})$. Indeed $\mathsf{just}(e_{4i+1}) \rightarrowtriangle_{(\!|p \circledast_\varphi q|\!)} e_{4i+1}$ by definition; and by *rule-abiding* this entails that $\mathsf{just}(e_{4i+1}) \rhd_q e_{4i+1}$. Since q is a forest, that makes $\mathsf{just}(e_{4i+1})$ comparable with e_{4i} for $\rhd_q^*$. If $\mathsf{just}(e_{4i+1}) = e_{4i}$, then

$$e_{4i-1} \rhd_p e_{4i+2}$$

since φ is a symmetry and by *rule-abiding* – but this allows us to shorten the cycle, contradicting its minimality.

Likewise, if $e_{4i} \rhd_q \mathsf{just}(e_{4i+1})$, then

$$e_{4i} \rhd_q \mathsf{just}(\mathsf{just}(e_{4i+1}))$$

by Lemma 6.5 – we cannot have an equality as they have distinct polarities. But then

$$e_{4i} \rhd_q \mathsf{just}(\mathsf{just}(e_{4i+1})) \rhd_q \mathsf{just}(\mathsf{just}(e_{4i+2})) \rhd_p e_{4i+3}$$

yielding a cycle with the same length but strictly smaller depth, absurd. The last case remaining has $\mathsf{just}(e_{4i+1}) \rhd_q e_{4i}$, but so $\mathsf{just}(e_{4i+1}) \rhd_q \mathsf{just}(e_{4i})$ by Lemma 6.5 (again, the equality is impossible for polarity reasons).

With the same reasoning, $\mathsf{just}(e_{4i+3}) \rhd_p \mathsf{just}(e_{4i+2})$; and $\mathsf{just}(e_{4i+2}) \rhd_\varphi \mathsf{just}(e_{4i+1})$ and $\mathsf{just}(e_{4i+4}) \rhd_\varphi \mathsf{just}(e_{4i+3})$ by definition. So we can replace the whole cycle with

$$\mathsf{just}(e_{4i+4}) \rhd_\varphi \mathsf{just}(e_{4i+3}) \rhd_p \mathsf{just}(e_{4i+2}) \rhd_\varphi \mathsf{just}(e_{4i+1}) \rhd_q \mathsf{just}(e_{4i})$$

reversing directions, with the same length but strictly smaller depth, contradiction. □

Since $\rhd$ is acyclic, the binary relation $\leq_{p \circledast_\varphi q}$ also is acyclic – which will allow us to extract an *augmentation* $r \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{C})$ with a partial order $\leq_r$ from $p \circledast_\varphi q$.

Proposition 6.7 – Interaction is acyclic

The interaction $p \circledast_\varphi q$ is a partially ordered set.

Proof. By Lemma 6.6, $\rhd$ is *acyclic*. Therefore, its transitive closure is a partial order. □

6.1.2 Composition via an isomorphism

Now, we define the composition: as in HO games, we *hide* the events occurring in B and only keep events occurring in A and C.

Definition 6.8 – Composition via an isomorphism

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ and $\varphi : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$.

The **composition of** q **with** p **according to** φ, noted $p \odot_\varphi q$, is the tuple $\langle |p \odot_\varphi q|, \leq_{(\!|p \odot_\varphi q|\!)}, \leq_{p \odot_\varphi q}, \partial_{p \odot_\varphi q} \rangle$ defined with:

$$\begin{array}{rcl} |p \odot_\varphi q| & = & |x_{q \upharpoonright \mathsf{A}}| + |x_{p \upharpoonright \mathsf{C}}| \\ \leq_{(\!|p \odot_\varphi q|\!)} & = & \leq_{x_{q \upharpoonright \mathsf{A}} \vdash x_{p \upharpoonright \mathsf{C}}} \\ e \leq_{p \odot_\varphi q} f & \text{iff} & e \leq_{p \circledast_\varphi q} f \\ \partial_{p \odot_\varphi q} & : & (1, e) \mapsto \partial_q((1, e)) \\ & & (2, e) \mapsto \partial_p((2, e)) \end{array}$$

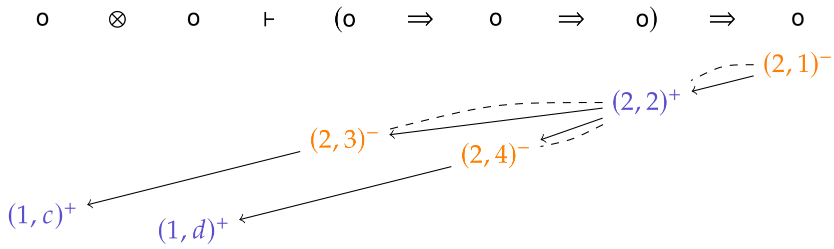


Figure 6.4: Composition $p \odot_\varphi q$

Example: Recall the augmentations from Figures 6.1 and 6.2, with the isomorphism

$$\varphi = \{(a \mapsto 5), (b \mapsto 6)\}\,.$$

The composition $p \odot_\varphi q$ is represented in Figure 6.4.

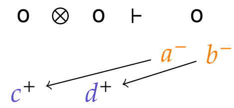


Figure 6.1: $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, with $\mathsf{A} = o \otimes o$ and $\mathsf{B} = o$.

In order to show that this composition is well-behaved, we need to characterise immediate causal dependency in the interaction. It turns out that this is very constrained – this is detailed in two lemmas: the first, for forward causality, follows.

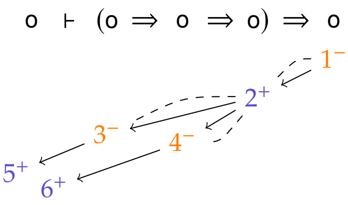


Figure 6.2: $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, with $\mathsf{B} = o$ and $\mathsf{C} = (o \Rightarrow o \Rightarrow o) \Rightarrow o$.

Lemma 6.9 – Forward causality

If $e = (1, e') \rightarrowtriangle_{p\circledast_\varphi q} f$ for $e' \in |q|$, then we have:

(1) If e' is negative in q, then $f = (1, f')$ and $e' \rightarrowtriangle_q f'$;
(2) If e' is positive in q and occurs in A, then $f = (1, f')$ and $e' \rightarrowtriangle_q f'$;
(3) If e' is positive in q and occurs in B, then $f = (2, \varphi(e'))$,

and symmetrically for $e = (2, e') \rightarrowtriangle_{p\circledast_\varphi q} f$ for $e' \in |p|$.

Proof. Any immediate causal link must originate from one of the clauses of the relation $\rhd$ above.

For ***(1)***, for polarity reasons it can only be $(1, e) \rhd_q f'$ so that $f' = (1, f)$ with $e <_q f$, and furthermore we must have $e \rightarrowtriangle_q f$ or that would immediately contradict $e \rightarrowtriangle_{p\circledast_\varphi q} f$.

For ***(2)***, similarly only the clause $\rhd_q$ may apply.

For ***(3)***, we must show that $e \rhd_q f$ is impossible. If that was the case, then $f = (1, f')$ with $e' <_q f'$, where we must have $e' \rightarrowtriangle_q f'$ (or contradict $e \rightarrowtriangle f$). But as e' is positive, by *courtesy* we have $e' \rightarrowtriangle_{(\!|q|\!)} f'$, and thus $\varphi(e') \rightarrowtriangle_{(\!|p|\!)} \varphi(f')$ as φ is an order-isomorphism. And by *rule-abiding*, that entails $\varphi(e') <_p \varphi(f')$, so that altogether we have

$$e \rhd_\varphi (2, \varphi(e')) \rhd_p (2, \varphi(f')) \rhd_\varphi f$$

contradicting the fact that $e \leq_{p\circledast_\varphi q} f$. □

Symmetrically, in the "backward" direction, we have:

Lemma 6.10 – Backward causality

If $e \rightarrowtriangle_{p \circledast_\varphi q} (1, f') = f$ for $f' \in |q|$, then we have:

(1) If f' is positive in q, then $e = (1, e')$ and $e' \rightarrowtriangle_q f'$;
(2) If f' is negative in q and occurs in A, then $e = (1, e')$ and $e' \rightarrowtriangle_q f'$;
(3) If f' is negative in q and occurs in B, then $e = (1, \varphi^{-1}(e'))$,

and symmetrically for $e \rightarrowtriangle_{p \circledast_\varphi q} (2, f')$ for $f' \in |p|$.

Proof. Analogous to the proof of lemma 6.9. □

We can now prove that the composition is an augmentation.

Proposition 6.11 – Composition via an isomorphism

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ and $\varphi \colon x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$.

Then the composition $p \odot_\varphi q$ is an augmentation on $\mathsf{A} \vdash \mathsf{C}$.

Proof. Let us call an event $e \in |p \circledast_\varphi q|$ **visible** if it appears in $p \odot_\varphi q$, **hidden** otherwise. For brevity, we write $r = p \odot_\varphi q$ and $\hat{r} = p \circledast_\varphi q$.

First, we check that $(\!|r|\!) = \langle |r|, \leq_{(\!|r|\!)}, \partial_r \rangle$ is a configuration. This is clear by construction:

$$(\!|r|\!) = x_{q \upharpoonright \mathsf{A}} \vdash x_{p \upharpoonright \mathsf{C}} \in \mathsf{Conf}(\mathsf{A} \vdash \mathsf{B})\,.$$

Then, we check that r is an augmentation.

Forestiality. An event $e \in |r|$ has at most one causal predecessor, by Lemma 6.10; and $\leq_r$ is acyclic by Proposition 6.7.

Rule-abidingness. Immediate from the definition and the fact that q and p are rule-abiding.

Courtesy. Consider $e \rightarrowtriangle_r f$: this implies a sequence

$$e \rightarrowtriangle_{\hat{r}} e_1 \rightarrowtriangle_{\hat{r}} \ldots \rightarrowtriangle_{\hat{r}} e_k \rightarrowtriangle_{\hat{r}} f$$

in $\hat{r}$, where $e_1, \ldots, e_k$ are hidden.

If e is positive, then e_1 cannot be hidden by Lemma 6.9, so $k = 0$ and $e \rightarrowtriangle_{\hat{r}} f$. As both e and f are visible, this is only possible if they both come from q or they both come from p. In any case, $e \rightarrowtriangle_{(\!|r|\!)} f$ by courtesy of q or p.

Likewise, if f is negative, we use Lemma 6.10 to show that $e \rightarrowtriangle_{(\!|r|\!)} f$.

Determinism. Since q and p are deterministic, and $\triangleright_\varphi$ does not branch, Lemma 6.9 entails that $\leq_{\hat{r}}$ can only branch at negative visible events, from which it follows that r is deterministic.

Negativity. An event minimal in $p \odot_\varphi q$ must come from p, must occur in C hence be visible, and be minimal in p, hence negative.

+-coveredness. Immediate from the definition and the fact that q and p are +-covered. □

Moreover, this operation preserves isomorphisms. First, remark that for any augmentations $q, p \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ with an isomorphism $\varphi\colon q \cong p$, we know that φ also is a configuration isomorphism and $\varphi\colon (\!|q|\!) \cong (\!|p|\!)$. We note φ_{A} (resp. φ_{B}) the restriction of φ to the events occuring in A (resp. B) – which is well-defined by arena-preservation of φ. Then:

$$\varphi_{\mathsf{A}}\colon x_{q\restriction\mathsf{A}} \cong_{\mathsf{A}} x_{p\restriction\mathsf{A}} \qquad \text{and} \qquad \varphi_{\mathsf{B}}\colon x_{q\restriction\mathsf{B}} \cong_{\mathsf{B}} x_{p\restriction\mathsf{B}}\,.$$

Lemma 6.12 – Composition preserves isomorphism

Consider $q, q' \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ and $p, p' \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, with the augmentation isomorphisms :

$$\varphi\colon q \cong q' \qquad \text{and} \qquad \psi\colon p \cong p'\,,$$

and the configuration isomorphisms :

$$\theta\colon x_{q\restriction\mathsf{B}} \cong_{\mathsf{B}} x_{p\restriction\mathsf{B}} \qquad \text{and} \qquad \theta'\colon x_{q'\restriction\mathsf{B}} \cong_{\mathsf{B}} x_{p'\restriction\mathsf{B}}\,,$$

such that the diagram of Figure 6.5 commutes.

Then, we have an augmentation isomorphism:

$$\psi \odot_{\theta,\theta'} \varphi\colon p \odot_\theta q \cong p' \odot_{\theta'} q'\,.$$

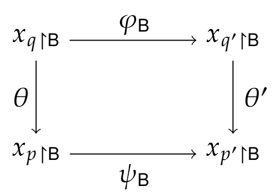


Figure 6.5: Congruence

Proof. We set the bijection

$$\begin{array}{rcrcl} \psi \circledast_{\theta,\theta'} \varphi & : & |p \circledast_\theta p| & \cong & |p' \circledast_{\theta'} q'| \\ & & (1, e) & \mapsto & (1, \varphi(e)) \\ & & (2, f) & \mapsto & (2, \psi(f)) \end{array}$$

which sends $\rhd_q$ to $\rhd_{q'}$ and $\rhd_p$ to $\rhd_{p'}$ by definition of φ and ψ. Likewise, it sends $\rhd_\theta$ to $\rhd_{\theta'}$ by hypothesis (Figure 6.5). Clearly, the symmetric statement holds for the inverse.

Therefore, it is clear that $\psi \circledast_{\theta,\theta'} \varphi$ restricts to

$$\psi \odot_{\theta,\theta'} \varphi\colon p \odot_\theta q \cong p' \odot_{\theta'} q'$$

as required. □

This allows us to extend the definition of composition to isogmentations: for any $\mathsf{q} \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ and $\mathsf{p} \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, if $\varphi\colon x_{\underline{\mathsf{q}}\restriction\mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}}\restriction\mathsf{B}}$, we define

$$\mathsf{p} \odot_\varphi \mathsf{q} \stackrel{\text{def}}{=} \overline{\underline{\mathsf{q}} \odot_\varphi \underline{\mathsf{p}}}\,.$$

For now the choice of representatives still matters because of φ, but we shall see in the next section that we can actually define a composition of isogmentations which does not depend on the representatives, by summing over all symmetries.

6.1.3 Composing isogmentations

We now have a definition of composition of augmentations *according to an isomorphism* – but what about *composition of augmentations/isogmentations in general*? Indeed, the composition of q and p is only defined once we have fixed a mediating $\varphi : x_{q\restriction B} \cong_B x_{p\restriction B}$, which is not necessarily unique. Worse, the result of composition depends on the choice of φ: if Figure 6.3 was constructed with the symmetry $\psi = \{(a \mapsto 6), (b \mapsto 5)\}$, we would get the alternative interaction of Figure 6.6.

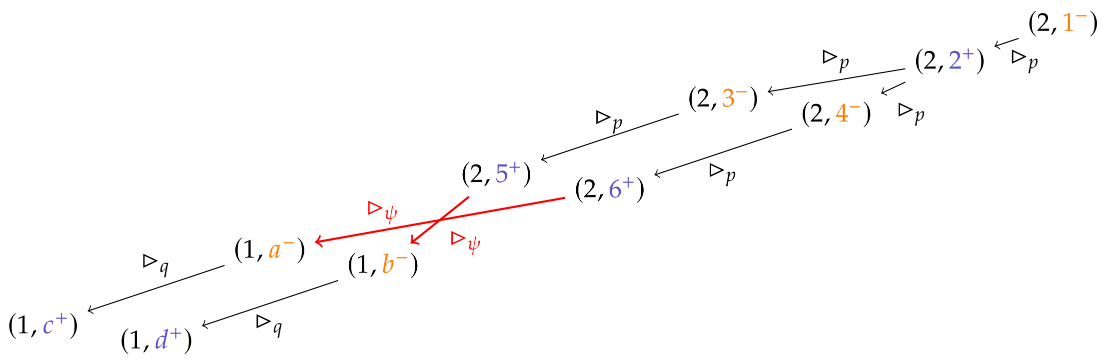


Figure 6.6: $p \circledast_\psi q$, where we label each immediate causality arrow with the relation it comes from (between $\rhd_q$, $\rhd_p$ and $\rhd_\psi$).

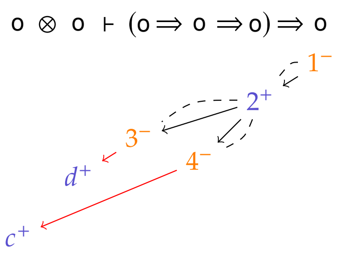


Figure 6.7: $p \odot_\psi q$ (where we dropped the tags from events' names for the sake of brevity).

The corresponding augmentation $p \odot_\psi q$ is shown in Figure 6.7; the two augmentations $p \odot_\varphi q$ and $p \odot_\psi q$ are clearly different – worse, they are not even isomorphic!

This is reminiscent of the behaviour of resource substitution. Consider for example the term

$$M = \lambda f.f\ [x][x]\,.$$

The *substitution* $M\langle [y,z]/x\rangle$ yields *two different resource terms*:

$$(\lambda f.f\ [x][x])\ \langle [y,z]/x\rangle = \lambda f.f\ [y]\ [z] + \lambda f.f\ [z]\ [y]\,,$$

which is analogous to how the composition of q and p yields two different, non-isomorphic augmentations. As substitution of resource terms yields *sums* of resource terms, this suggests that composition of isogmentations should produce *sums* of isogmentations.

Definition 6.13 – Composition of isogmentations

Consider $\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})$ and $\mathsf{p} \in \mathsf{Isog}(\mathsf{B} \vdash \mathsf{C})$.

Their **composition** $\mathsf{p} \odot \mathsf{q}$ is defined as:

$$\mathsf{p} \odot \mathsf{q} \overset{\text{def}}{=} \sum_{\varphi :\, x_{\underline{\mathsf{q}}\restriction \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}}\restriction \mathsf{B}}} \mathsf{p} \odot_\varphi \mathsf{q}\,.$$

For this definition to make sense, we want the composition to be compatible with isomorphisms, so that the isogmentations obtained by the composition *do not* depend on the choice of representatives.

Lemma 6.14 – Composition does not depend on representatives

Consider $\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})$ and $\mathsf{p} \in \mathsf{Isog}(\mathsf{B} \vdash \mathsf{C})$.

For any $q \in \mathsf{q}$ and $p \in \mathsf{p}$, we have:

$$\mathsf{p} \odot \mathsf{q} = \sum_{\theta : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}} \overline{p \odot_\theta q}\,.$$

Proof. Fix any isomorphisms $\varphi : \underline{\mathsf{q}} \cong q$ and $\psi : \underline{\mathsf{p}} \cong p$, projected to

$$\varphi_{\mathsf{B}} : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{q \upharpoonright \mathsf{B}} \quad \text{and} \quad \psi_{\mathsf{B}} : x_{\underline{\mathsf{p}} \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}\,.$$

Writing $[x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}]$ for the set of isomorphisms $\theta : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$, we define the bijection:

$$\begin{array}{rcrcl} \Omega & : & [x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}} \upharpoonright \mathsf{B}}] & \cong & [x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}] \\ & & \theta & \mapsto & \psi_{\mathsf{B}} \circ \theta \circ \varphi_{\mathsf{B}}^{-1} \end{array}.$$

Then, for any $\theta : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}} \upharpoonright \mathsf{B}}$, we can apply Lemma 6.12 (the diagram of Figure 6.5 commutes by definition of Ω) to obtain

$$\underline{\mathsf{p}} \odot_\theta \underline{\mathsf{q}} \;\cong\; p \odot_{\Omega(\theta)} q\,. \tag{6.1}$$

Now, we calculate:

$$\begin{array}{rcll} \mathsf{p} \odot \mathsf{q} & = & \sum_{\theta : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}} \upharpoonright \mathsf{B}}} \overline{\underline{\mathsf{p}} \odot_\theta \underline{\mathsf{q}}} & \text{(Definition 6.13)} \\ & = & \sum_{\theta : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}} \upharpoonright \mathsf{B}}} \overline{p \odot_{\Omega(\theta)} q} & \text{(Equation 6.1)} \\ & = & \sum_{\theta' : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}} \overline{p \odot_{\theta'} q} & (\Omega \text{ is a bijection}) \end{array}$$

as required. □

This allows us to define, in general, the composition of two augmentations $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ and $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ as:

$$p \odot q \overset{\text{def}}{=} \overline{q} \odot \overline{p} = \sum_{\theta : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}} \overline{p \odot_\theta q} \tag{6.2}$$

and we shall often use this equation to move between augmentations and isogmentations – remark that the composition of augmentations is always a sum of *isogmentations*.

6.2 Strategies and identities

6.2.1 Strategies

Recall that in HO games, strategies are sets of plays – similarly, in PCG strategies are *weighted sums of isogmentations*.

Definition 6.15 – Strategy

A **strategy** on arena A is a function $\sigma : \mathsf{Isog}(\mathsf{A}) \to \overline{\mathbb{R}}_+$, where $\overline{\mathbb{R}}_+$ is the completed half-line of non-negative reals. We then write $\sigma : \mathsf{A}$.

Remark: An isogmentation $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$ may be considered as a strategy, with coefficient 1 for q and 0 for any other isogmentation.

We regard $\sigma : \mathsf{A}$ as a weighted sum

$$\sigma = \sum_{\mathsf{q} \in \mathsf{Isog}(\mathsf{A})} \sigma(\mathsf{q}) \cdot \mathsf{q}\,,$$

and we write $\mathsf{supp}(\sigma)$ for its support set:

$$\mathsf{supp}(\sigma) \stackrel{\text{def}}{=} \{\mathsf{q} \in \mathsf{Isog}(\mathsf{A}) \mid \sigma(\mathsf{q}) \neq 0\}\,.$$

We can lift the composition of isogmentations to strategies.

Definition 6.16 – Composition of strategies

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$ and $\tau : \mathsf{B} \vdash \mathsf{C}$.

Their **composition** $\tau \odot \sigma : \mathsf{A} \vdash \mathsf{C}$ is defined via the formula:

$$\tau \odot \sigma \quad \stackrel{\text{def}}{=} \quad \sum_{\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})} \sum_{\mathsf{p} \in \mathsf{Isog}(\mathsf{B} \vdash \mathsf{C})} \sigma(\mathsf{q})\tau(\mathsf{p}) \cdot (\mathsf{p} \odot \mathsf{q})\,.$$

In other words, the coefficient $(\tau \odot \sigma)(\mathsf{r})$ is the sum of $\sigma(\mathsf{q}) \times \tau(\mathsf{p})$ over all triples $\mathsf{q}, \mathsf{p}, \varphi$ such that $\mathsf{r} = \mathsf{p} \odot_\varphi \mathsf{q}$ – there are no convergence issues, as we consider positive coefficients and we have been careful to include $+\infty \in \overline{\mathbb{R}}_+$ in Definition 6.15.

We have now defined *strategies* from A to B, as well as a composition on strategies: we show in Section 6.4 that PCG, the structure formed by negative arenas and strategies between them, is a category.

6.2.2 Identities

But first, we focus on some key strategies: *copycat strategies*, formal sums of specific isogmentations presenting typical copycat behaviour, which will act as identities in PCG.

We start by defining their concrete representatives.

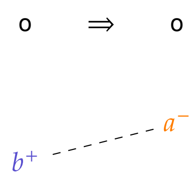


Figure 6.8: A configuration $x \in \mathsf{Conf}(\mathsf{A})$, with $\mathsf{A} = \mathsf{o} \Rightarrow \mathsf{o}$.

Definition 6.17 – Copycat augmentation on x

Consider $x \in \mathsf{Conf}(\mathsf{A})$ on negative arena A. The augmentation $\mathbb{C}_x \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{A})$, called the **copycat augmentation on** x, is defined with

- $(\!|\mathbb{C}_x|\!) \stackrel{\text{def}}{=} x \vdash x$,
- the causal order $\leq_{\mathbb{C}_x}$ is the transitive closure of $\leq_{x \vdash x}$ augmented with:

$$\begin{array}{llll} (1, e) & \rightarrowtriangle_{\mathbb{C}_x} & (2, f) & \text{if } e \leq_x f \text{ and } \mathsf{pol}_\mathsf{A}(\partial_x(e)) = +,\\ (2, e) & \rightarrowtriangle_{\mathbb{C}_x} & (1, f) & \text{if } e \leq_x f \text{ and } \mathsf{pol}_\mathsf{A}(\partial_x(e)) = -. \end{array}$$

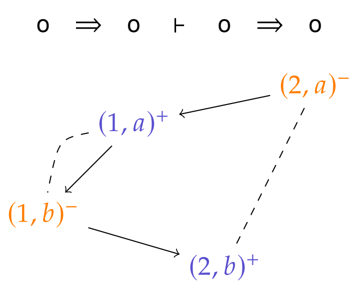


Figure 6.9: The augmentation $\mathbb{C}_x$.

In other words, $\mathbb{C}_x$ adds to $x \vdash x$ all immediate causal links of the form $(2, e) \rightarrowtriangle (1, e)$ for negative e, and $(1, e) \rightarrowtriangle (2, e)$ for positive e. Consider for instance the configuration x from Figure 6.8; the copycat augmentation on x is presented in Figure 6.9.

This lifts to isogmentations.

Definition 6.18 – Copycat isogmentation on x

Consider $\mathsf{x} \in \mathsf{Pos}(\mathsf{A})$. The **copycat isogmentation on** x, noted $\mathbb{c}_{\mathsf{x}} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{A})$, is defined by:

$$\mathbb{c}_{\mathsf{x}} \stackrel{\text{def}}{=} \overline{\mathbb{c}_{\underline{\mathsf{x}}}}\,.$$

Again, for this definition to make sense, we want the copycat isogmentation not to depend on the choice of representatives.

Lemma 6.19 – Copycat preserves isomorphisms

Consider $x, y \in \mathsf{Conf}(\mathsf{A})$. Then,

$$x \cong_{\mathsf{A}} y \quad \text{iff} \quad \mathbb{c}_x \cong \mathbb{c}_y\,.$$

Proof. ***Only if.*** Consider $\theta : x \cong_{\mathsf{A}} y$, then

$$\begin{array}{rcrcl} \varphi_\theta & : & \mathbb{c}_x & \cong & \mathbb{c}_y \\ & & (i, e) & \mapsto & (i, \theta(e)) \end{array}$$

is an augmentation isomorphism.

If. Likewise, if $\varphi : \mathbb{c}_x \cong \mathbb{c}_y$, then

$$\begin{array}{rcrcl} \theta_\varphi & : & x & \cong_{\mathsf{A}} & y \\ & & e & \mapsto & f \text{ such that } \varphi((1, e)) = (1, f) \end{array}$$

is a configuration isomorphism. □

Now, we can define our *identity strategies*: sums of copycat isogmentation on all positions of an arena. But with which coefficients? Since we want to obtain identities, we need to choose coefficients which exactly cancel the sum over all symmetries in the composition (Definition 6.16).

Definition 6.20 – Copycat strategy

The **copycat strategy** on the arena A, noted $\mathsf{id}_{\mathsf{A}} : \mathsf{A} \vdash \mathsf{A}$, is

$$\mathsf{id}_{\mathsf{A}} \stackrel{\text{def}}{=} \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{A})} \frac{1}{\sharp \mathsf{Sym}\,(\mathsf{x})} \cdot \mathbb{c}_{\mathsf{x}}\,,$$

where $\mathsf{Sym}\,(\mathsf{x})$ is the group of **endosymmetries** of x, *i.e.* of all symmetries $\theta : \underline{\mathsf{x}} \cong_{\mathsf{A}} \underline{\mathsf{x}}$ – remark that $\sharp\mathsf{Sym}\,(\mathsf{x})$ the cardinality of $\mathsf{Sym}\,(\mathsf{x})$ does not depend on the choice of the representative $\underline{\mathsf{x}}$

This use of such a coefficient to compensate for future sums over sets of permutations is reminiscent of the Taylor expansion of λ-terms.

6.3 The categorical structure of PCG

We finally have all the ingredients needed to build a category:

- negative arenas (objects),
- strategies between them (morphisms),
- composition,
- and identities.

Categorical laws will be proven in several stages. First, we establish isomorphisms corresponding to them, working concretely on augmentations – this means that these laws will refer to certain isomorphisms explicitly. Then, we use the compatibility of composition of augmentations with isomorphisms to transport these laws to isogmentations.

6.3.1 Associativity of the composition

To prove that the composition is associative, we define a ternary composition and prove that the composition of three morphisms (using the binary composition twice) is equal to their ternary composition, no matter the order of the compositions.

First, we define ternary interactions.

Definition 6.21 – Ternary interaction

Consider $q_1 \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $q_2 \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$ and $q_3 \in \mathsf{Aug}(\mathsf{C} \vdash \mathsf{D})$, with configuration isomorphisms:

$$\varphi : x_{q_1} \upharpoonright_{\mathsf{B}} \cong_{\mathsf{B}} x_{q_2} \upharpoonright_{\mathsf{B}} \qquad \text{and} \qquad \psi : x_{q_2} \upharpoonright_{\mathsf{C}} \cong_{\mathsf{C}} x_{q_3} \upharpoonright_{\mathsf{C}} .$$

We define the **ternary interaction** $q_3 \circledast^3_\psi q_2 \circledast^3_\varphi q_1$ with:

$$\begin{array}{rcll}
|q_3 \circledast^3_\psi q_2 \circledast^3_\varphi q_1| & \stackrel{\text{def}}{=} & |q_1| + |q_2| + |q_3| \,, & \\
\lhd^3_{q_i} & \stackrel{\text{def}}{=} & \{((i,e),(i,e')) \mid e <_{q_i} e'\} & \text{for } i = 1,2,3\,, \\
\lhd^3_\varphi & \stackrel{\text{def}}{=} & \{((1,e),(2,\varphi(e))) \mid \mathsf{pol}_{\mathsf{A}\vdash\mathsf{B}}(\partial_{q_1}((1,e))) = +\} & \\
 & \cup & \{((2,\varphi(e)),(1,e)) \mid \mathsf{pol}_{\mathsf{A}\vdash\mathsf{B}}(\partial_{q_1}((1,e))) = -\}\,, & \\
\lhd^3_\psi & \stackrel{\text{def}}{=} & \{((2,e),(3,\psi(e))) \mid \mathsf{pol}_{\mathsf{B}\vdash\mathsf{C}}(\partial_{q_2}((2,e))) = +\} & \\
 & \cup & \{((3,\psi(e)),(2,e)) \mid \mathsf{pol}_{\mathsf{B}\vdash\mathsf{C}}(\partial_{q_2}((2,e))) = -\}\,, & \\
\lhd_{q_3 \circledast^3_\psi q_2 \circledast^3_\varphi q_1} & \stackrel{\text{def}}{=} & \lhd^3_{q_1} \cup \lhd^3_{q_2} \cup \lhd^3_{q_3} \cup \lhd^3_\varphi \cup \lhd^3_\psi \,. &
\end{array}$$

This allows us to define ternary compositions.

Definition 6.22 – Ternary composition

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, $r \in \mathsf{Aug}(\mathsf{C} \vdash \mathsf{D})$, with configuration isomorphisms $\varphi \colon x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$ and $\psi \colon x_{p \upharpoonright \mathsf{C}} \cong_{\mathsf{C}} x_{r \upharpoonright \mathsf{C}}$.

We define their **ternary composition** $r \odot^3_\psi p \odot^3_\varphi q$ with

$$\begin{array}{rcl}
|r \odot^3_\psi p \odot^3_\varphi q| & \overset{\text{def}}{=} & |x_{q \upharpoonright \mathsf{A}}| + \emptyset + |x_{r \upharpoonright \mathsf{D}}| \\
(1,e) \leq_{(r \odot^3_\psi p \odot^3_\varphi q)} (1,f) & \text{iff} & e \leq_{x_{q \upharpoonright \mathsf{A}}} f \\
(3,e) \leq_{(r \odot^3_\psi p \odot^3_\varphi q)} (3,f) & \text{iff} & e \leq_{x_{p \upharpoonright \mathsf{C}}} f \\
(i,e) \leq_{r \odot^3_\psi p \odot^3_\varphi q} (j,f) & \text{iff} & (i,e) \left(\lhd_{r \circledast^3_\psi p \circledast^3_\varphi q} \right)^* (j,f) \\
\partial_{r \odot^3_\psi p \odot^3_\varphi q} & : & (1,e) \mapsto (1, \partial_{x_{q \upharpoonright \mathsf{A}}}(e)) \\
 & & (3,e) \mapsto (2, \partial_{x_{r \upharpoonright \mathsf{D}}}(e)) .
\end{array}$$

Remark: For now we do not claim that $r \odot^3_\psi p \odot^3_\varphi q$ is an augmentation. In particular we did not prove the acyclicity of $\lhd_{r \circledast^3_\psi p \circledast^3_\varphi q}$, thus we do not know if $\leq_{r \odot^3_\psi p \odot^3_\varphi q}$ is antisymmetric. However the proof of Proposition 6.25 does not rely on this; the antisymmetry of $\leq_{r \odot^3_\psi p \odot^3_\varphi q}$ will actually be a consequence of the isomorphism between $r \odot^3_\psi p \odot^3_\varphi q$ and $(r \odot_\psi p) \odot_{\ell \circ \varphi} q$.

We want to prove the following claim:

$$r \odot_{\psi \circ r^{-1}} (p \odot_\varphi q) \;\cong\; r \odot^3_\psi p \odot^3_\varphi q \;\cong\; (r \odot_\psi p) \odot_{\ell \circ \varphi} q \tag{6.3}$$

for q, p, r, φ, ψ as in the definition above, and the bijections

$$\ell \colon e \mapsto (1,e) \qquad \text{and} \qquad r \colon e \mapsto (2,e) .$$

First, we need some lemmas on relations:

Lemma 6.23 – Relations on disjoint sets

If $\blacktriangleleft$ is a relation on $\mathsf{A} \uplus \mathsf{B}$ two disjoint sets, then the following are equivalent:

(1) $\forall \mathsf{b}_1, \mathsf{b}_2 \in \mathsf{B}$ and $\mathsf{a}_1, \ldots \mathsf{a}_n \in \mathsf{A}$,
if $\mathsf{b}_1 \blacktriangleleft \mathsf{a}_1 \blacktriangleleft \ldots \blacktriangleleft \mathsf{a}_n \blacktriangleleft \mathsf{b}_2$,
then $\exists\, \mathsf{b}'_1, \ldots, \mathsf{b}'_k \in \mathsf{B}$ such that $\mathsf{b}_1 \blacktriangleleft \mathsf{b}'_1 \blacktriangleleft \ldots \blacktriangleleft \mathsf{b}'_k \blacktriangleleft \mathsf{b}_2$;

(2) $\blacktriangleleft^* \upharpoonright \mathsf{B} = (\blacktriangleleft \upharpoonright \mathsf{B})^*$.

Lemma 6.24 – Star of two relations

If $\blacktriangleleft_1, \blacktriangleleft_2$ are relations on A, then

$$(\blacktriangleleft_1 \uplus \blacktriangleleft_2^*)^* = (\blacktriangleleft_1 \uplus \blacktriangleleft_2)^* .$$

Now, we can go back to Equation (6.3).

Proposition 6.25 – Composition is associative

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, $r \in \mathsf{Aug}(\mathsf{C} \vdash \mathsf{D})$, with configuration isomorphisms $\varphi \colon x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$ and $\psi \colon x_{p \upharpoonright \mathsf{C}} \cong_{\mathsf{C}} x_{r \upharpoonright \mathsf{C}}$.

Then we have:

$$(r \odot_\psi p) \odot_{\ell \circ \varphi} q \;\cong\; r \odot_{\psi \circ r^{-1}} (p \odot_\varphi q) .$$

Reminder: r and ℓ are the bijections $\ell \colon e \mapsto (1,e)$ and $r \colon e \mapsto (2,e)$.

Proof. By definition of the composition, it is clear that

$$\ell \circ \varphi \colon x_{q\upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{(r \odot_\psi p)\upharpoonright \mathsf{B}} \qquad \text{and} \qquad \psi \circ \varkappa^{-1} \colon x_{(p \circ_\varphi q)\upharpoonright \mathsf{C}} \cong_{\mathsf{C}} x_{r\upharpoonright \mathsf{C}}\,;$$

indeed, composition preserves the underlying configuration structure and only adds tags to events.

To prove that the two augmentations are isomorphic, we actually prove that each is isomorphic to the ternary composition $r \odot^3_\psi p \odot^3_\varphi q$.

We start by proving that

$$\left(r \odot_\psi p\right) \odot_{\ell\circ\varphi} q \;\cong\; r \odot^3_\psi p \odot^3_\varphi q\,.$$

The two sets of events of both augmentations[1] differ only by the tags, so we set χ with

1: we have not proved yet that the object $r \odot^3_\psi p \odot^3_\varphi q$ *is* an augmentation, but it will be a consequence of the isomorphism.

$$\forall e \in \left|\, \left(r \odot_\psi p\right) \odot_{\ell\circ\varphi} q \right|,\; \chi(e) = \begin{cases} (1, e') & \text{if } e = (1, e') \\ (3, e') & \text{if } e = (2, (2, e')) \end{cases}$$

and we have $\chi \colon \left|\, \left(r \odot_\psi p\right) \odot_{\ell\circ\varphi} q\right| \cong \left|r \odot^3_\psi p \odot^3_\varphi q\right|$.

We now prove that χ is an isomorphism of augmentations.

Arena-preserving. We easily check that

$$\partial_{(r\odot_\psi p)\,\odot_{\ell\circ\varphi} q} = \partial_{r\odot^3_\psi p \odot^3_\varphi q} \circ \chi\,.$$

Configuration. By definition of the composition, we have:

$$\begin{aligned} \leq_{\llparenthesis (r\odot_\psi p)\odot_{\ell\circ\varphi} q\rrparenthesis} &= \leq_{x_{q\upharpoonright \mathsf{A}}} + \leq_{x_{r\odot_\psi p\upharpoonright \mathsf{D}}} \\ &= \leq_{x_{q\upharpoonright \mathsf{A}}} + \left(\emptyset + \leq_{x_{r\upharpoonright \mathsf{D}}}\right)\,. \end{aligned}$$

Writing $\chi(\blacktriangleleft)$ for $\{(\chi(a), \chi(b)) \mid a \blacktriangleleft b\}$ for any relation $\blacktriangleleft$, we obtain:

$$\begin{aligned} \chi\left(\leq_{\llparenthesis (r\odot_\psi p)\odot_{\ell\circ\varphi} q\rrparenthesis}\right) &= \chi\left(\leq_{x_{q\upharpoonright \mathsf{A}}} + \left(\emptyset + \leq_{x_{r\upharpoonright \mathsf{D}}}\right)\right) \\ &= \leq_{\llparenthesis r\odot^3_\psi p\odot^3_\varphi q\rrparenthesis}\,. \end{aligned}$$

Causality. By definition, we have:

$$\leq_{(r\odot_\psi p)\odot_{\ell\circ\varphi} q} = \left(\lhd_q \uplus \lhd_{r\odot_\psi p} \uplus \lhd_{\ell\circ\varphi}\right)^* \upharpoonright \left(|x_{q\upharpoonright \mathsf{A}}| + |x_{r\odot_\psi p\upharpoonright \mathsf{D}}|\right)\,.$$

For any $i \in \mathbb{N}$, set E and relation $\blacktriangleleft$, we note $(i, E) = \{(i, e) \mid e \in E\}$ and $(i, \blacktriangleleft) = \{((i, e_1), (i, e_2)) \mid e_1 \blacktriangleleft e_2\}$. Then, by definition again:

$$\begin{aligned} \lhd_{r\odot_\psi p} &= (2, \leq_{r\odot_\psi p}) \\ &= \left((2, \lhd_p) \uplus (2, \lhd_r) \uplus (2, \lhd_\psi)\right)^* \upharpoonright (2, (1, |x_{p\upharpoonright \mathsf{B}}|)) \uplus (2, (2, |x_{r\upharpoonright \mathsf{D}}|))\,. \end{aligned}$$

Now, since $\lhd_{r\odot_\psi p}$ is defined on $(2, |r \odot_\psi p|)$, *i.e.* on $(2, (1, |p|))$ and $(2, (2, |r|))$, we can add:

$$\begin{aligned} \lhd_{r\odot_\psi p} = &\left((2, \lhd_p) \uplus (2, \lhd_r) \uplus (2, \lhd_\psi)\right)^* \\ &\upharpoonright \left((2, (1, |x_{p\upharpoonright \mathsf{B}}|)) \uplus (2, (2, |x_{r\upharpoonright \mathsf{D}}|)) \uplus (1, |q|)\right)\,. \end{aligned}$$

Moreover, $\lhd_q$ is defined on $(1, |q|)$, and $\lhd_\varphi$ on $(1, |x_{q\restriction \mathsf{B}}|)$ and on $(2, |(r \odot_\psi p)_\mathsf{B}|) = (2, (1, |x_{p\restriction \mathsf{B}}|))$. So we can also write

$$\begin{aligned} &\lhd_q \uplus \lhd_{\ell\circ\varphi} \\ = &\left(\lhd_q \uplus \lhd_{\ell\circ\varphi}\right) \restriction \left((2,(1,|x_{p\restriction \mathsf{B}}|)) \uplus (2,(2,|x_{r\restriction \mathsf{D}}|)) \uplus (1,|q|)\right). \end{aligned}$$

Putting both equalities together, we obtain

$$\begin{aligned} \leq_{(r\odot_\psi p)\odot_{\ell\circ\varphi} q} = \Bigg(&\left(\lhd_q \uplus \left((2,\lhd_p) \uplus (2,\lhd_r) \uplus (2,\lhd_\psi)\right)^* \uplus \lhd_{\ell\circ\varphi}\right) \\ &\restriction \left((2,(1,|x_{p\restriction \mathsf{B}}|)) \uplus (2,(2,|x_{r\restriction \mathsf{D}}|)) \uplus (1,|q|)\right)\Bigg)^* \\ &\restriction (1,|x_{q\restriction \mathsf{A}}|) \uplus (2,(2,|x_{r\restriction \mathsf{D}}|)). \end{aligned}$$

For the sake of readability, we write:

$$\begin{aligned} E &= (2,(1,|x_{p\restriction \mathsf{C}}|)) \uplus (2,(2,|x_{r\restriction \mathsf{C}}|)), \\ F &= (2,(1,|x_{p\restriction \mathsf{B}}|)) \uplus (2,(2,|x_{r\restriction \mathsf{D}}|)) \uplus (1,|q|), \\ \lessdot &= \left(\lhd_q \uplus \left((2,\lhd_p) \uplus (2,\lhd_r) \uplus (2,\lhd_\psi)\right)^* \uplus \lhd_{\ell\circ\varphi}\right) \end{aligned}$$

Then $\lessdot$ is defined on $E \uplus F$, and we can apply Lemma 6.23. Indeed, for any chain

$$f_1 \lessdot e_1 \lessdot e_2 \lessdot \ldots \lessdot e_n \lessdot f_2$$

with $f_1, f_2 \in F$ and $\forall\, 1 \leq i \leq n, e_i \in E$, we know that all links must come from $\left((2,\lhd_p) \uplus (2,\lhd_r) \uplus (2,\lhd_\psi)\right)^*$ (because $\lhd_q$ and $\lhd_{\ell\circ\varphi}$ are only defined on F); so we also have $f_1 \lessdot f_2$. So, using Lemma 6.23, we write:

$$\begin{aligned} \leq_{(r\odot_\psi p)\odot_{\ell\circ\varphi} q} = &\left(\lhd_q \uplus \left((2,\lhd_p) \uplus (2,\lhd_r) \uplus (2,\lhd_\psi)\right)^* \uplus \lhd_{\ell\circ\varphi}\right)^* \\ &\restriction \left((1,|x_{q\restriction \mathsf{A}}|) \uplus (2,(2,|x_{r\restriction \mathsf{D}}|))\right). \end{aligned}$$

By Lemma 6.24, we obtain

$$\begin{aligned} \leq_{(r\odot_\psi p)\odot_{\ell\circ\varphi} q} = &\left(\lhd_q \uplus (2,\lhd_p) \uplus (2,\lhd_r) \uplus (2,\lhd_\psi) \uplus \lhd_{\ell\circ\varphi}\right)^* \\ &\restriction \left((1,|x_{q\restriction \mathsf{A}}|) \uplus (2,(2,|x_{r\restriction \mathsf{D}}|))\right). \end{aligned}$$

We can extend χ to an isomorphism $\chi' : |q| + (|p| + |r|) \cong |r \circledast^3_\psi p \circledast^3_\varphi q|$ with

$$\forall e \in |q| + (|p| + |r|),\ \chi'(e) = \begin{cases} (1,e') & \text{if } e = (1,e') \\ (2,e') & \text{if } e = (2,(1,e')) \\ (3,e') & \text{if } e = (2,(2,e')) \end{cases}.$$

Then it is clear that $\leq_{(r\odot_\psi p)\odot_{\ell\circ\varphi} q}$ and $\leq_{r\odot^3_\psi p\odot^3_\varphi q}$ are isomorphic *via* χ, concluding the first part of the proof.

Likewise, we prove $r \odot_{\psi\circ r^{-1}} \left(p \odot_\varphi q\right) \cong r \odot^3_\psi p \odot^3_\varphi q$.

Hence, $\left(r \odot_\psi p\right) \odot_{\ell\circ\varphi} q \;\cong\; r \odot_{\psi\circ r^{-1}} \left(p \odot_\varphi q\right)$. □

This means that the composition on *isogmentations* is associative.

Lemma 6.26 – Associativity for isogmentations

Consider $\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})$, $\mathsf{p} \in \mathsf{Isog}(\mathsf{B} \vdash \mathsf{C})$ and $\mathsf{r} \in \mathsf{Isog}(\mathsf{C} \vdash \mathsf{D})$.

Then,

$$(\mathsf{r} \odot \mathsf{p}) \odot \mathsf{q} \;=\; \mathsf{r} \odot (\mathsf{p} \odot \mathsf{q})\,.$$

Proof. We compute:

$$\begin{aligned}
(\mathsf{r} \odot \mathsf{p}) \odot \mathsf{q} &= \left(\sum_{\theta\colon x_{\underline{\mathsf{p}}\upharpoonright \mathsf{C}} \cong_{\mathsf{C}} x_{\underline{\mathsf{r}}\upharpoonright \mathsf{C}}} \overline{\underline{\mathsf{r}} \odot_\theta \underline{\mathsf{p}}} \right) \odot \mathsf{q} \\
&= \sum_{\theta\colon x^{\underline{\mathsf{p}}}_{\mathsf{C}} \cong_{\mathsf{C}} x^{\underline{\mathsf{r}}}_{\mathsf{C}}} \left((\overline{\underline{\mathsf{r}} \odot_\theta \underline{\mathsf{p}}}) \odot \mathsf{q} \right) \\
&= \sum_{\theta\colon x_{\underline{\mathsf{p}}\upharpoonright \mathsf{C}} \cong_{\mathsf{C}} x_{\underline{\mathsf{r}}\upharpoonright \mathsf{C}}} \left((\underline{\mathsf{r}} \odot_\theta \underline{\mathsf{p}}) \odot \underline{\mathsf{q}} \right) \\
&= \sum_{\theta\colon x_{\underline{\mathsf{p}}\upharpoonright \mathsf{C}} \cong_{\mathsf{C}} x_{\underline{\mathsf{r}}\upharpoonright \mathsf{C}}} \left(\sum_{\varphi\colon x_{\underline{\mathsf{q}}\upharpoonright \mathsf{B}} \cong_{\mathsf{B}} \ell(x_{\underline{\mathsf{p}}\upharpoonright \mathsf{B}})} \left((\underline{\mathsf{r}} \odot_\theta \underline{\mathsf{p}}) \odot_\varphi \underline{\mathsf{q}} \right) \right) \\
&= \sum_{\theta\colon x_{\underline{\mathsf{p}}\upharpoonright \mathsf{C}} \cong_{C} x_{\underline{\mathsf{r}}\upharpoonright \mathsf{C}}} \left(\sum_{\varphi\colon x_{\underline{\mathsf{q}}\upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}}\upharpoonright \mathsf{B}}} \left((\underline{\mathsf{r}} \odot_\theta \underline{\mathsf{p}}) \odot_{\ell \circ \varphi} \underline{\mathsf{q}} \right) \right) \\
&= \sum_{\theta\colon x_{\underline{\mathsf{p}}\upharpoonright \mathsf{C}} \cong_{C} x_{\underline{\mathsf{r}}\upharpoonright \mathsf{C}}} \left(\sum_{\varphi\colon x_{\underline{\mathsf{q}}\upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}}\upharpoonright \mathsf{B}}} \left(\underline{\mathsf{r}} \odot_{\theta \circ \ell^{-1}} (\underline{\mathsf{p}} \odot_\varphi \underline{\mathsf{q}}) \right) \right)
\end{aligned}$$

using the definition of composition of isogmentations; the definition of composition of strategies; Lemma 3.12 with Lemma 6.14; the definition of composition of augmentations; a direct substitution; and Proposition 6.25.

Lemmas and proposition used:
– 6.14: composition does not depend on the choice of representatives;
– 3.12: for any isogmentation q, $(\underline{\mathsf{q}}) \cong \mathsf{q}$;
– 6.25: associativity of composition of augmentations.

We then perfom the same steps in reverse order to obtain:

$$\sum_{\theta\colon x_{\underline{\mathsf{p}}\upharpoonright \mathsf{C}} \cong_{C} x_{\underline{\mathsf{r}}\upharpoonright \mathsf{C}}} \left(\sum_{\varphi\colon x_{\underline{\mathsf{q}}\upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{\underline{\mathsf{p}}\upharpoonright \mathsf{B}}} \left(\underline{\mathsf{r}} \odot_{\theta \circ \ell^{-1}} (\underline{\mathsf{p}} \odot_\varphi \underline{\mathsf{q}}) \right) \right) \;=\; \mathsf{r} \odot (\mathsf{p} \odot \mathsf{q})\;. \qquad \square$$

Thus, we have the associativity of composition for strategies.

Proposition 6.27 – Associativity for strategies

Consider $\sigma\colon \mathsf{A} \vdash \mathsf{B}$, $\tau\colon \mathsf{B} \vdash \mathsf{C}$ and $\delta\colon \mathsf{C} \vdash \mathsf{D}$ strategies. Then,

$$\delta \odot (\tau \odot \sigma) = (\delta \odot \tau) \odot \sigma\,.$$

Proof. Direct by bilinearity of composition and Lemma 6.26. $\square$

6.3.2 Neutrality of copycat

Again, we start with results on augmentations, before moving on to isogmentations and strategies.

First, we characterize causal links from negative events to positive events in copycat augmentations.

Lemma 6.28 – Causal links in copycat

Consider $x \in \mathsf{Conf}(\mathsf{A})$, and $e^-, f^+ \in |\mathfrak{c}_x|$ such that $e^- \rightarrowtriangle_{\mathfrak{c}_x} f^+$.

Then there are two cases:

$$\begin{array}{ll} (1) & e = (1,a),\ f = (2,a), \text{ and } \mathsf{pol}_x(a) = +, \\ (2) & e = (2,a),\ f = (1,a), \text{ and } \mathsf{pol}_x(a) = -. \end{array}$$

Proof. If $e = (1,a)$ then for any $a^+ \rightarrowtriangle_x b^-$ we have

$$(1,a) \leq_{\mathfrak{c}_x} (2,a) \leq_{\mathfrak{c}_x} (2,b) \leq_{\mathfrak{c}_x} (1,b)\,.$$

Likewise, if $e = (2,a)$ then for any $a^- \rightarrowtriangle_x b^+$ we have

$$(2,a) \leq_{\mathfrak{c}_x} (1,a) \leq_{\mathfrak{c}_x} (1,b) \leq_{\mathfrak{c}_x} (2,b)\,. \qquad \square$$

Reminder: Configurations *preserves causality*, meaning the order on x follows the order on A; and arenas are *alternating*, meaning that an event and its imediate successor cannot have the same polarity.

These are the only causal links that will concern us in the proof of neutrality; indeed, the following lemma expresses the fact that in any proof of isomorphism between two augmentations, if we already know their underlying configurations are isomorphic, then we only have to check the links from negative to positive events.

Lemma 6.29 – Isomorphic underlying configurations

Consider two augmentations q, p in any arena D.

If there exists a configuration isomorphism $\varphi : (\!|q|\!) \cong_{\mathsf{D}} (\!|p|\!)$, then for any $e^+, f^- \in |q|$,

$$e^+ \rightarrowtriangle_q f^- \Leftrightarrow \varphi(e)^+ \rightarrowtriangle_p \varphi(f)^-\,.$$

Proof. By courtesy and rule-abidingness, $e^+ \rightarrowtriangle_q f^-$ if and only if $e^+ \rightarrowtriangle_{(\!|q|\!)} f^-$. Likewise, $\varphi(e)^+ \rightarrowtriangle_p \varphi(f)^-$ if and only if $\varphi(e)^+ \rightarrowtriangle_{(\!|p|\!)} \varphi(f)^-$. Finally, since φ is a configuration isomorphism, $e^+ \rightarrowtriangle_{(\!|q|\!)} f^-$ if and only if $\varphi(e)^+ \rightarrowtriangle_{(\!|p|\!)} \varphi(f)^-$. $\square$

We can now prove the neutrality of copycat.

Lemma 6.30 – Neutrality of copycat

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $x \in \mathsf{Conf}(\mathsf{B})$ and $\varphi : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x$. Then,

$$\mathfrak{c}_x \odot_{\ell \circ \varphi} q \cong q\,.$$

Likewise, for $y \in \mathsf{Conf}(\mathsf{A})$ and $\psi : y \cong_{\mathsf{A}} x_{q \upharpoonright \mathsf{A}}$, we have:

$$q \odot_{\psi \circ r^{-1}} \mathfrak{c}_y \cong q$$

Proof. We have $|\mathbb{c}_x \odot_\varphi q| = |x_{q\restriction \mathsf{A}}| + (\emptyset + |x|)$. Consider:

$$\begin{array}{rccl} \chi: & |q| & \to & |\mathbb{c}_x \odot_{\ell\circ\varphi} q| \\ & e & \mapsto & \begin{cases} (1,e) & \text{if } e \in |x_{q\restriction \mathsf{A}}| \\ (2,(2,\varphi(e))) & \text{if } e \in |x_{q\restriction \mathsf{B}}|. \end{cases} \end{array}$$

We prove that χ is an isomorphism of augmentions.

Arena-prevering. Clear by definition.

Configuration-preserving. Since φ preserves the configuration order, we have:

$$\chi\left(\leq_{(\!|q|\!)}\right) = \leq_{x_{q\restriction \mathsf{A}}} + (\emptyset + \leq_x) = \leq_{(\!|\mathbb{c}_x \odot_{\ell\circ\varphi} q|\!)} .$$

Causality-preserving. For the sake of brevity, we write $\multimap_{\circledast}$ for $\multimap_{\mathbb{c}_x \circledast_{\ell\circ\varphi} q}$ and $\multimap_{\odot}$ for $\multimap_{\mathbb{c}_x \odot_{\ell\circ\varphi} q}$. By Lemma 6.29, we only need to look at links from negative to positive moves.

Consider $e^- \multimap_q f^+$. Then we have four possibilities:

(1) If $e, f \in |x_{q\restriction \mathsf{A}}|$.
Then $\chi(e) = (1,e)^-$ and $\chi(f) = (1,f)^+$. By Lemma 6.9,

$$(1,e)^- \multimap_{\circledast} (1,f)^+ .$$

Hence $\chi(e) \multimap_{\odot} \chi(f)$.

(2) If $e \in |x_{q\restriction \mathsf{A}}|$ and $f \in |x_{q\restriction \mathsf{B}}|$.
Then $\chi(e) = (1,e)^-$ and $\chi(f) = (2,(2,\varphi(f)))^+$. We have:

$$(1,e)^- \multimap_{\circledast} (1,f) \multimap_{\circledast} (2,(1,\varphi(f))) \multimap_{\circledast} (2,(2,\varphi(f)))^+$$

by Lemmas 6.9 and 6.28. Moreover, $(1,f)$ and $(2,(1,\varphi(f)))$ occur in B, so after the hiding we have $\chi(e) \multimap_{\odot} \chi(f)$.

(3) If $e \in |x_{q\restriction \mathsf{B}}|$ and $f \in |x_{q\restriction \mathsf{A}}|$.
Then $\chi(e) = (2,(2,\varphi(e)))^-$ and $\chi(f) = (1,f)^+$. We have:

$$(2,(2,\varphi(e)))^- \multimap_{\circledast} (2,(1,\varphi(e))) \multimap_{\circledast} (1,e) \multimap_{\circledast} (1,f)^+$$

by Lemmas 6.9 and 6.28. Moreover, $(2,(1,\varphi(e)))$ and $(1,e)$ occur in B, so after the hiding we have $\chi(e) \multimap_{\odot} \chi(f)$.

(4) If $e, f \in |x_{q\restriction \mathsf{B}}|$.
Then $\chi(e) = (2,(2,\varphi(e)))^-$ and $\chi(f) = (2,(2,\varphi(f)))^+$. By Lemmas 6.9 and 6.28, we have:

$$\begin{aligned}&(2,(2,\varphi(e)))^- \multimap_{\circledast} (2,(1,\varphi(e))) \multimap_{\circledast} (1,e)\\ &\multimap_{\circledast} (1,f) \multimap_{\circledast} (2,(1,\varphi(f))) \multimap_{\circledast} (2,(2,\varphi(f)))^+ .\end{aligned}$$

Again, $(2,(1,\varphi(e)))$, $(1,e)$, $(1,f)$ and $(2,(1,\varphi(f)))$ occur in B, so after the hiding we obtain $\chi(e) \multimap_{\odot} \chi(f)$.

Symmetrically, if $e^- \multimap_{\odot} f^+$, then $\chi^{-1}(e) \multimap_q \chi^{-1}(f)$.

The proof for the other isomorphism is similar. □

It may be surprising that $cc_x \odot_{\ell \circ \varphi} q \cong q$ regardless of φ: the choice of the symmetry is reflected in the isomorphism $\chi_\varphi : cc_x \odot_{\ell \circ \varphi} q \cong q$ obtained, which the statement of this lemma ignores.

Again, this result on augmentations extends nicely to the composition with the copycat strategy.

Proposition 6.31 – Neutrality of id

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$. Then, $\mathsf{id}_\mathsf{B} \odot \sigma = \sigma \odot \mathsf{id}_\mathsf{A} = \sigma$.

Proof. We focus on $\mathsf{id}_\mathsf{B} \odot \sigma$. First, we have:

$$
\begin{aligned}
&\mathsf{id}_\mathsf{B} \odot \sigma \\
&= \left(\sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \frac{1}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \mathsf{cc}_\mathsf{x} \right) \odot \left(\sum_{\mathsf{q} \in \mathsf{Isog}^+(A \vdash B)} \sigma(\mathsf{q}) \cdot \mathsf{q} \right) \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\mathsf{q} \in \mathsf{Isog}^+(A \vdash B)} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot (\mathsf{cc}_\mathsf{x} \odot \mathsf{q}) \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot (\overline{\mathsf{cc}_{\underline{\mathsf{x}}}} \odot \mathsf{q}) \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \left(\underline{(\overline{\mathsf{cc}_{\underline{\mathsf{x}}}})} \odot \underline{\mathsf{q}} \right) \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \left(\mathsf{cc}_{\underline{\mathsf{x}}} \odot \underline{\mathsf{q}} \right) \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \sum_{\theta : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_\mathsf{B} \underline{\mathsf{x}}} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \overline{\left(\mathsf{cc}_{\underline{\mathsf{x}}} \odot_{\ell \circ \theta} \underline{\mathsf{q}} \right)} \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \sum_{\theta : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_\mathsf{B} \underline{\mathsf{x}}} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \overline{(\underline{\mathsf{q}})} \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \sum_{\theta : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_\mathsf{B} \underline{\mathsf{x}}} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \mathsf{q} \\
&= \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{B})} \sum_{\substack{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B}) \\ \text{s.t. } x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_\mathsf{B} \underline{\mathsf{x}}}} \sum_{\theta : x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}} \cong_\mathsf{B} \underline{\mathsf{x}}} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\left(\overline{x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}}}\right)} \cdot \mathsf{q} \\
&= \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \sum_{\theta \in \mathsf{Sym}\left(\overline{x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}}}\right)} \frac{\sigma(\mathsf{q})}{\sharp\mathsf{Sym}\left(\overline{x_{\underline{\mathsf{q}} \upharpoonright \mathsf{B}}}\right)} \cdot \mathsf{q} \\
&= \sum_{\mathsf{q} \in \mathsf{Isog}^+(\mathsf{A} \vdash \mathsf{B})} \sigma(\mathsf{q}) \cdot \mathsf{q} \\
&= \sigma
\end{aligned}
$$

by unfolding the definition of the identity strategy (6.20); unfolding the definition of the composition of strategies (6.16); definition of the copycat isogmentation (6.18); Equation (6.2) and Lemma 3.12; Equation (6.2) and Lemma 3.12 again; definition of the composition of augmentations (Eq (6.2)) and of the copycat augmentation (6.17); Lemma 6.30; Lemma 3.12; Lemma 3.12 again; and a direct reasoning on symmetries.

Lemmas used:
– 3.12: representatives and isomorphism classes;
– 6.30: neutrality of copycat for augmentations.

The proof of the identity $\sigma \odot \mathsf{id}_\mathsf{A} = \sigma$ is symmetric. □

Notice how the sum over all symmetries exactly compensates for the coefficient in Definition 6.20!

All in all, we obtain a category PCG.

Theorem 6.32 – PCG is a category

PCG is a category defined with:

- the objects are negative arenas,
- for any arenas A, B, the morphisms from A to B are the strategies on $A \vdash B$,
- the composition is given by Definition 6.16,
- for any arena A, the identity is id_A.

Proof. Composition is associative by Proposition 6.27 and identities are neutral by Proposition 6.31. □

6.4 PCG is a SMCC

Now, we prove that PCG has a symmetric monoidal structure (see Definitions 1.1 and 1.2).

6.4.1 Tensor

We already defined the tensor of arenas (Definition 2.4) and of configurations (Definition 5.6). We now extend this construction to augmentations, then isogmentations, and finally strategies.

Definition 6.33 – Tensor of augmentations

Consider $q_1 \in \mathsf{Aug}(\mathsf{A}_1 \vdash \mathsf{B}_1)$ and $q_2 \in \mathsf{Aug}(\mathsf{A}_2 \vdash \mathsf{B}_2)$.

Their **tensor** is the augmentation $q_1 \otimes q_2 \in \mathsf{Aug}((\mathsf{A}_1 \otimes \mathsf{A}_2) \vdash (\mathsf{B}_1 \otimes \mathsf{B}_2))$ defined with:

$$(\!|q_1 \otimes q_2|\!) = \left(x_{q_1 \upharpoonright \mathsf{A}_1} \otimes x_{q_2 \upharpoonright \mathsf{A}_2}\right) \vdash \left(x_{q_1 \upharpoonright \mathsf{B}_1} \otimes x_{q_2 \upharpoonright \mathsf{B}_2}\right)$$

and

$$(k, (i, e)) \leq_{q_1 \otimes q_2} (l, (i, f)) \Leftrightarrow e \leq_{q_i} f\,.$$

This construction clearly preserves isomorphisms, hence it lifts to isogmentations using any representative. For definition, we take:

$$\mathsf{q}_1 \otimes \mathsf{q}_2 \stackrel{\text{def}}{=} \overline{\underline{\mathsf{q}_1} \otimes \underline{\mathsf{q}_2}} \qquad \in \mathsf{Isog}(\mathsf{A}_1 \otimes \mathsf{A}_2 \vdash \mathsf{B}_1 \otimes \mathsf{B}_2)$$

for any isogmentations $\mathsf{q}_1 \in \mathsf{Isog}(\mathsf{A}_1 \vdash \mathsf{B}_1)$ and $\mathsf{q}_2 \in \mathsf{Isog}(\mathsf{A}_2 \vdash \mathsf{B}_2)$.

Finally, we lift the definition to strategies.

Definition 6.34 – Tensor of strategies

Consider $\sigma_1 : \mathsf{A}_1 \vdash \mathsf{B}_1$ and $\sigma_2 : \mathsf{A}_2 \vdash \mathsf{B}_2$.

Their **tensor**, noted $\sigma_1 \otimes \sigma_2 : (\mathsf{A}_1 \otimes \mathsf{A}_2) \vdash (\mathsf{B}_1 \otimes \mathsf{B}_2)$, is the strategy defined with:

$$\sigma_1 \otimes \sigma_2 \stackrel{\text{def}}{=} \sum_{\mathsf{q}_1 \in \mathsf{Isog}(\mathsf{A}_1 \vdash \mathsf{B}_1)} \sum_{\mathsf{q}_2 \in \mathsf{Isog}(\mathsf{A}_2 \vdash \mathsf{B}_2)} \sigma_1(\mathsf{q}_1)\, \sigma_2(\mathsf{q}_2) \cdot (\mathsf{q}_1 \otimes \mathsf{q}_2) .$$

We now prove that this construction is a functor $\otimes : \mathsf{PCG} \times \mathsf{PCG} \longrightarrow \mathsf{PCG}$.

Indeed, we have the following lemmas regarding tensor:

Lemma 6.35 – Tensor and positions

Consider two arenas A_1 and A_2. Then:

$$\mathsf{Pos}(\mathsf{A}_1 \otimes \mathsf{A}_2) \cong \mathsf{Pos}(\mathsf{A}_1) \times \mathsf{Pos}(\mathsf{A}_2) .$$

Moreover, for any $\mathsf{x}_1 \in \mathsf{Pos}(\mathsf{A}_1)$ and $\mathsf{x}_2 \in \mathsf{Pos}(\mathsf{A}_2)$, we have:

$$\mathsf{Sym}(\mathsf{x}_1 \otimes \mathsf{x}_2) \cong \mathsf{Sym}(\mathsf{x}_1) \times \mathsf{Sym}(\mathsf{x}_2) .$$

Proof idea: Consider $x_1, y_1 \in \mathsf{Conf}(\mathsf{A}_1)$ and $x_2, y_2 \in \mathsf{Conf}(\mathsf{A}_2)$, with a bijection

$$\varphi : x_1 \otimes x_2 \cong y_1 \otimes y_2 .$$

We decompose φ in two isos φ_1, φ_2 with:

$$\begin{array}{rccl} \varphi_i : |x_i| & \to & |y_i| \\ e & \mapsto & f \text{ s.t. } \varphi((i,e)) = ((i,f)) \end{array} .$$

Lemma 6.36 – Tensor and copycat augmentation

Consider $x_1 \in \mathsf{Conf}(\mathsf{A}_1)$, $x_2 \in \mathsf{Conf}(\mathsf{A}_2)$. Then:

$$Œ_{x_1 \otimes x_2} \cong Œ_{x_1} \otimes Œ_{x_2} .$$

Copycat forces both side of the augmentation to be copies of the same configuration, meaning that any copycat isogmentation (on $\mathsf{A} \vdash \mathsf{A}$) can be characterized simply by a position on A.

Lemma 6.37 – Support of identity

Consider an arena A, then

$$\mathsf{supp}(\mathsf{id}_\mathsf{A}) \cong \mathsf{Pos}(\mathsf{A}) .$$

Proof idea: We set a bijection

$$\kappa : \mathsf{x} \in \mathsf{Pos}(\mathsf{A}) \mapsto \overline{Œ_{\underline{\mathsf{x}}}} \in \mathsf{supp}(\mathsf{id}_\mathsf{A}) .$$

These lemmas allow us to prove that the tensor respects identities.

Lemma 6.38 – Tensor and identities

Consider A, B arenas. Then,

$$\mathsf{id}_\mathsf{A} \otimes \mathsf{id}_\mathsf{B} = \mathsf{id}_{\mathsf{A} \otimes \mathsf{B}} .$$

Proof idea: Similar to the proof of Proposition 6.31, using the previous lemmas.

Proof. We have the following equalities:

$$\mathsf{id}_\mathsf{A} \otimes \mathsf{id}_\mathsf{B} = \sum_{\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{A})} \sum_{\mathsf{p} \in \mathsf{Isog}(\mathsf{B} \vdash \mathsf{B})} (\mathsf{id}_\mathsf{A}(\mathsf{q}) \times \mathsf{id}_\mathsf{B}(\mathsf{p})) \cdot (\mathsf{q} \otimes \mathsf{p})$$

$$
\begin{aligned}
&= \sum_{\mathsf{x}\in\mathsf{Pos}(\mathsf{A})}\sum_{\mathsf{y}\in\mathsf{Pos}(\mathsf{B})} \frac{1}{\sharp\mathsf{Sym}(\mathsf{x})\times\sharp\mathsf{Sym}(\mathsf{y})}\cdot\left(\overline{\mathfrak{C}_{\underline{\mathsf{x}}}}\otimes\overline{\mathfrak{C}_{\underline{\mathsf{y}}}}\right)\\
&= \sum_{\mathsf{x}\in\mathsf{Pos}(\mathsf{A})}\sum_{\mathsf{y}\in\mathsf{Pos}(\mathsf{B})} \frac{1}{\sharp\mathsf{Sym}(\mathsf{x})\times\sharp\mathsf{Sym}(\mathsf{y})}\cdot\left(\overline{\mathfrak{C}_{\underline{\mathsf{x}}}\otimes\mathfrak{C}_{\underline{\mathsf{y}}}}\right)\\
&= \sum_{\mathsf{x}\in\mathsf{Pos}(\mathsf{A})}\sum_{\mathsf{y}\in\mathsf{Pos}(\mathsf{B})} \frac{1}{\sharp\mathsf{Sym}(\mathsf{x})\times\sharp\mathsf{Sym}(\mathsf{y})}\cdot\left(\overline{\mathfrak{C}_{\underline{\mathsf{x}}\otimes\underline{\mathsf{y}}}}\right)\\
&= \sum_{\mathsf{x}\in\mathsf{Pos}(\mathsf{A})}\sum_{\mathsf{y}\in\mathsf{Pos}(\mathsf{B})} \frac{1}{\sharp\mathsf{Sym}(\mathsf{x})\times\sharp\mathsf{Sym}(\mathsf{y})}\cdot\left(\overline{\mathfrak{C}_{\underline{\mathsf{x}\otimes\mathsf{y}}}}\right)\\
&= \sum_{\mathsf{x}\in\mathsf{Pos}(\mathsf{A})}\sum_{\mathsf{y}\in\mathsf{Pos}(\mathsf{B})} \frac{1}{\sharp\mathsf{Sym}(\mathsf{x}\otimes\mathsf{y})}\cdot\left(\overline{\mathfrak{C}_{\underline{\mathsf{x}\otimes\mathsf{y}}}}\right)\\
&= \sum_{\mathsf{z}\in\mathsf{Pos}(\mathsf{A}\otimes\mathsf{B})} \frac{1}{\sharp\mathsf{Sym}(\mathsf{z})}\cdot\overline{\mathfrak{C}_{\underline{\mathsf{z}}}}\\
&= \mathsf{id}_{\mathsf{A}\otimes\mathsf{B}}
\end{aligned}
$$

by Definition 6.34 (tensor of strategies); Definition 6.20 (copycat strategy) and Lemma 6.37 (support of the copycat strategy); preservation of isomorphisms by tensor; Lemma 6.36 (tensor of copycat augmentations); Lemma 5.8 (tensor of configurations); Lemma 6.35 (tensor and symmetries); Lemma 6.35 (tensor of positions); and finally the definition of copycat again. □

Likewise, we have the following property for tensor and composition of augmentations:

Proof idea: For $i = 1, 2$, we set

$$\theta_i : b \in |x^{q_i}_{\mathsf{B}_i}| \mapsto b' \in |x^{p_i}_{\mathsf{B}_i}|$$

with b' such that $\theta(i, b) = (i, b')$. Then

$$
\begin{aligned}
\chi:\quad (1,(1,a)) &\mapsto (1,(1,a))\\
(1,(2,a)) &\mapsto (2,(1,a))\\
(2,(1,c)) &\mapsto (1,(2,c))\\
(2,(2,c)) &\mapsto (2,(2,c))
\end{aligned}
$$

is an isomorphism between $(p_1 \otimes p_2) \odot_\theta (q_1 \otimes q_2)$ and $(p_1 \odot_{\theta_1} q_1) \otimes (p_2 \odot_{\theta_2} q_2)$.

Lemma 6.39 – Tensor and comp., augmentations

Consider $q_i : \mathsf{A}_i \vdash \mathsf{B}_i$, $p_i : \mathsf{A}_i \vdash \mathsf{B}_i$ for $i = 1, 2$, along with the isomorphism:

$$\theta : x^{q_1\otimes q_2}_{\mathsf{B}_1\otimes\mathsf{B}_2} \cong_{\mathsf{B}_1\otimes\mathsf{B}_2} x^{p_1\otimes p_2}_{\mathsf{B}_1\otimes\mathsf{B}_2} .$$

Then θ can be decomposed into two isomorphisms

$$\theta_1 : x^{q_1}_{\mathsf{B}_1} \cong_{\mathsf{B}_1} x^{p_1}_{\mathsf{B}_1} \quad\text{and}\quad \theta_2 : x^{q_2}_{\mathsf{B}_2} \cong_{\mathsf{B}_1} x^{p_2}_{\mathsf{B}_2}$$

such that:

$$(p_1 \otimes p_2) \odot_\theta (q_1 \otimes q_2) \;\cong\; (p_1 \odot_{\theta_1} q_1) \otimes (p_2 \odot_{\theta_2} q_2) .$$

This translates to a property for strategies:

Proof idea: Similar to the proof of Proposition 6.31, using the previous lemma.

Lemma 6.40 – Tensor and comp., strategies

Consider $\sigma_i : \mathsf{A}_i \vdash \mathsf{B}_i$, $\tau_i : \mathsf{A}_i \vdash \mathsf{B}_i$ for $i = 1, 2$. Then,

$$(\tau_1 \odot \sigma_1) \otimes (\tau_2 \odot \sigma_2) \;=\; (\tau_1 \otimes \tau_2) \odot (\sigma_1 \otimes \sigma_2) .$$

Using Lemmas 6.38 and 6.40, we obtain that $\otimes$ is a bifunctor.

Proposition 6.41 – Functoriality of ⊗

The tensor $\otimes$ is a bifunctor on $\mathsf{PCG} \times \mathsf{PCG} \to \mathsf{PCG}$.

6.4.2 Structural morphisms – intuitively

Structural morphisms are all variations of copycat. As we did for copycat itself, we start with concrete representatives. Consider A, B, C arenas, and $x \in \mathsf{Conf}(\mathsf{A})$, $y \in \mathsf{Conf}(\mathsf{B})$, $z \in \mathsf{Conf}(\mathsf{C})$. Recall that we write I for the empty arena. Denoting the empty configuration on I with $\emptyset$, we set:

$$\begin{array}{rclrcl} (\!|\lambda^x_{\mathsf{A}}|\!) & = & \emptyset \otimes x \vdash x\,, & (\!|\alpha^{x,y,z}_{\mathsf{A,B,C}}|\!) & = & x \otimes (y \otimes z) \vdash (x \otimes y) \otimes z\,, \\ (\!|\rho^x_{\mathsf{A}}|\!) & = & x \otimes \emptyset \vdash x\,, & (\!|\gamma^{x,y}_{\mathsf{A,B}}|\!) & = & x \otimes y \vdash y \otimes x\,, \end{array}$$

and the corresponding augmentations are defined from these, augmented with the obvious copycat behaviour.

We lift this to isogmentations: for $\mathsf{x} \in \mathsf{Pos}(\mathsf{A})$, $\boldsymbol{\lambda}^{\mathsf{x}}_{\mathsf{A}}$ is the isomorphism class of $\lambda^{\mathsf{x}}_{\mathsf{A}}$; and likewise for the others. Then the strategy λ_{A} is defined as for id_{A} by setting:

$$\lambda_{\mathsf{A}} \stackrel{\text{def}}{=} \sum_{\mathsf{x} \in \mathsf{Pos}(\mathsf{A})} \frac{1}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \boldsymbol{\lambda}^{\mathsf{x}}_{\mathsf{A}}$$

and likewise for ρ_{A}, $\alpha_{\mathsf{A,B,C}}$ and $\gamma_{\mathsf{A,B}}$.

Before proving that these structural morphisms satisfy the conditions of Definitions 1.1 (monoidal category) and 1.2 (symmetric monoidal category), we need a few more technical tools, which we introduce in the next subsection.

6.4.3 Renamings

In order to handle the structural morphisms more easily, we introduce *renamings*, which allow us to change the arena image of an augmentation without modifying its structure.

Definitions. We start with renamings of arenas.

Definition 6.42 – Renamings on arenas

Consider arenas A and B. A **renaming** $f \in \mathsf{Ren}(\mathsf{A}, \mathsf{B})$ is a function $f : |\mathsf{A}| \to |\mathsf{B}|$ such that:

minimality-preserving: a minimal for $\leq_{\mathsf{A}} \Leftrightarrow f(\mathsf{a})$ minimal for $\leq_{\mathsf{B}}$,
causality-preserving: if $\mathsf{a}_1 \leq_{\mathsf{A}} \mathsf{a}_2$ then $f(\mathsf{a}_1) \leq_{\mathsf{B}} f(\mathsf{a}_2)$.

We now define (co-)renamings on configurations.

Definition 6.43 – Configuration (co-)renamings

Consider $x \in \mathsf{Conf}(\mathsf{A} \vdash \mathsf{B})$, $f \in \mathsf{Ren}(\mathsf{B}, \mathsf{B}')$, $g \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$. We define the **renaming of** x **by** f, denoted $f \rtimes x$, as:

$$\begin{array}{rcl} |f \rtimes x| & := & |x| \\ \leq_{f \rtimes x} & := & \leq_x \\ \partial_{f \rtimes x} & : & (1, e) \mapsto \partial_x((1, e)) \\ & & (2, e) \mapsto (2, f(\mathsf{b})) \text{ s.t. } (2, \mathsf{b}) = \partial_x((2, e))\,. \end{array}$$

Likewise, we define the **co-renaming of** x **by** g, denoted $x \ltimes g$, as:

$$\begin{array}{rcl} |x \ltimes g| & := & |x| \\ \leq_{x \ltimes g} & := & \leq_x \\ \partial_{x \ltimes g} & : & (1, e) \mapsto (1, g(\mathsf{a})) \text{ s.t. } (1, \mathsf{a}) = \partial_x((1', e)) \\ & & (2, e) \mapsto (2, e)\,. \end{array}$$

By definition of configurations and (co-)renamings, we have:

Proposition 6.44

With x, f, g as above, we obtain:

$$f \rtimes x \in \mathsf{Conf}(\mathsf{A} \vdash \mathsf{B}') \qquad x \ltimes g \in \mathsf{Conf}(\mathsf{A}' \vdash \mathsf{B})\,.$$

This allows us to define renamings on augmentations and isogmentations.

Definition 6.45 – Augmentation (co-)renamings

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $f \in \mathsf{Ren}(\mathsf{B}, \mathsf{B}')$, $g \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$.

We define the **renaming of** q **by** f, denoted $f \rtimes q$, as:

$$\begin{array}{rcl} (\!| f \rtimes q |\!) & := & f \rtimes (\!| q |\!) \\ \leq_{f \rtimes q} & := & \leq_q \quad . \end{array}$$

Likewise, we define the **co-renaming of** q **by** g, denoted $q \ltimes g$, as:

$$\begin{array}{rcl} (\!| q \ltimes g |\!) & := & (\!| q |\!) \ltimes g \\ \leq_{q \ltimes g} & := & \leq_p \quad . \end{array}$$

Again, by definition, we have:

Proposition 6.46

With q, f, g as above, we obtain:

$$f \rtimes q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B}') \qquad q \ltimes g \in \mathsf{Aug}(\mathsf{A}' \vdash \mathsf{B})\,.$$

It is clear that (co-)renamings are invariant under isomorphism.

Lemma 6.47

Consider the augmentations $q, p \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ such that $q \cong p$, along with the renamings $f \in \mathsf{Ren}(\mathsf{B}, \mathsf{B}')$ and $g \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$.

Then:

$$f \rtimes q \;\cong\; f \rtimes p \qquad \text{and} \qquad q \ltimes g \;\cong\; p \ltimes g\,.$$

Thus we unambiguously define isogmentation (co-)renamings.

Definition 6.48 – Isogmentation (co-)renamings

Consider $\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})$, $f \in \mathsf{Ren}(\mathsf{B}, \mathsf{B}')$, $g \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$.

We define the **renaming of** q **by** f, denoted $f \rtimes \mathsf{q}$, as:

$$f \rtimes \mathsf{q} := \overline{f \rtimes \underline{\mathsf{q}}}\,.$$

Likewise, we define the **co-renaming of** q **by** g, denoted $\mathsf{q} \ltimes g$, as:

$$\mathsf{q} \ltimes g := \overline{\underline{\mathsf{q}} \ltimes g}\,.$$

Finally, we can define renamings of strategies.

Definition 6.49 – Strategy (co-)renamings

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$, $f \in \mathsf{Ren}(\mathsf{B}, \mathsf{B}')$, $g \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$.

We define the **renaming of** σ **by** f, denoted $f \rtimes \sigma$, as:

$$f \rtimes \sigma := \sum_{\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})} \sigma(\mathsf{q}) \cdot (f \rtimes \mathsf{q}) \ .$$

Likewise, we define the **co-renaming of** σ **by** g, denoted $\sigma \ltimes g$, as:

$$\sigma \ltimes g := \sum_{\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})} \sigma(\mathsf{q}) \cdot (\mathsf{q} \ltimes g) \ .$$

Again, (co-)renamings of strategies are strategies:

Proposition 6.50

Consider σ, f, g as above. Then:

$$f \rtimes \sigma : \mathsf{A} \vdash \mathsf{B}' \qquad\qquad \sigma \ltimes g : \mathsf{A}' \vdash \mathsf{B}\,.$$

Now we can define structural morphisms using renamings – but before that, we state a few technical lemmas.

Technical lemmas. Renamings have properties that will be useful for proving that the structual morphisms indeed satisfy the definition of a resource category. We state some of these properties here; most of the proofs are immediate by definition.

Lemma 6.51 – Identity renaming

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$. Then:

$$\mathsf{id}_{\mathsf{B}} \rtimes \sigma \ = \ \sigma \ = \ \sigma \ltimes \mathsf{id}_{\mathsf{A}}\,.$$

Lemma 6.52 – Composition of renamings

Consider $\sigma : \mathsf{A}_0 \vdash \mathsf{B}_0$, $f_1 \in \mathsf{Ren}(\mathsf{B}_0, \mathsf{B}_1)$, $f_2 \in \mathsf{Ren}(\mathsf{B}_1, \mathsf{B}_2)$. Then:

$$f_2 \rtimes (f_1 \rtimes \sigma) \;=\; (f_2 \circ f_1) \rtimes \sigma\,.$$

Likewise, for any $g_1 \in \mathsf{Ren}(\mathsf{A}_0, \mathsf{A}_1)$, $g_2 \in \mathsf{Ren}(\mathsf{A}_1, \mathsf{A}_2)$, we have:

$$(\sigma \ltimes g_1) \ltimes g_2 \;=\; \sigma \ltimes (g_2 \circ g_1)\;.$$

Lemma 6.53 – Renaming of a composition

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$, $\tau : \mathsf{B} \vdash \mathsf{C}$, $f \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$, $g \in \mathsf{Ren}(\mathsf{C}, \mathsf{C}')$.

Then:

$$(\tau \odot \sigma) \ltimes f \;=\; \tau \odot (\sigma \ltimes f)$$

and

$$g \rtimes (\tau \odot \sigma) \;=\; (g \rtimes \tau) \odot \sigma\,.$$

Lemma 6.54 – Inverse renaming

Consider $f \in \mathsf{Ren}(\mathsf{A}, \mathsf{B})$. If f is invertible, then:

$$f \rtimes \mathsf{id}_{\mathsf{A}} \;=\; \mathsf{id}_{\mathsf{B}} \ltimes f^{-1}\,.$$

Lemma 6.55 – Composition with a renaming

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$, $f \in \mathsf{Ren}(\mathsf{B}, \mathsf{C})$ invertible, and $\tau : \mathsf{C} \vdash \mathsf{D}$. Then:

$$\tau \odot (f \rtimes \sigma) \;=\; (\tau \ltimes f^{-1}) \odot \sigma\,.$$

Lemma 6.56 – Renamings and tensors

Consider two strategies $\sigma_1 : \mathsf{A} \vdash \mathsf{C}$, $\sigma_2 : \mathsf{B} \vdash \mathsf{D}$ and two renamings $f_1 \in \mathsf{Ren}(\mathsf{C}, \mathsf{C}')$, $f_2 \in \mathsf{Ren}(\mathsf{D}, \mathsf{D}')$.

We define the product $f_1 \times f_2$ as:

$$\begin{array}{rccc} f_1 \times f_2 : & (\mathsf{C} \otimes \mathsf{D}) & \to & (\mathsf{C}' \otimes \mathsf{D}') \\ & (i, \mathsf{e}) & \mapsto & (i, f_i(\mathsf{e}))\,. \end{array}$$

Then:

$$(f_1 \rtimes \sigma_1) \otimes (f_2 \rtimes \sigma_2) \;=\; (f_1 \times f_2) \rtimes (\sigma_1 \otimes \sigma_2)\;.$$

Likewise, for any $g_1 \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$, $g_2 \in \mathsf{Ren}(\mathsf{B}, \mathsf{B}')$, we have:

$$(\sigma_1 \ltimes g_1) \otimes (\sigma_2 \ltimes g_2) = (\sigma_1 \otimes \sigma_2) \ltimes (g_1 \times g_2)\;.$$

Lemma 6.57 – Identity with an invertible renaming

Consider $f \in \mathsf{Ren}(\mathsf{A}, \mathsf{A}')$ a bijection.

Then $f \rtimes \mathsf{id}_\mathsf{A}$ and $\mathsf{id}_\mathsf{A} \ltimes f$ are isomorphisms.

Proof. The inverses are $f^{-1} \rtimes \mathsf{id}_{\mathsf{A}'}$ and $\mathsf{id}_{\mathsf{A}'} \ltimes f^{-1}$. We can check that:

$$
\begin{aligned}
(f^{-1} \rtimes \mathsf{id}_{\mathsf{A}'}) \odot (f \rtimes \mathsf{id}_\mathsf{A}) &= (\mathsf{id}_\mathsf{A} \ltimes f) \odot (f \rtimes \mathsf{id}_\mathsf{A}) && \text{(Lemma 6.54)}\\
&= \mathsf{id}_\mathsf{A} \odot \left(f^{-1} \rtimes (f \rtimes \mathsf{id}_\mathsf{A})\right) && \text{(Lemma 6.55)}\\
&= f^{-1} \rtimes (f \rtimes \mathsf{id}_\mathsf{A}) && \text{(Lemma 6.30)}\\
&= (f^{-1} \circ f) \rtimes \mathsf{id}_\mathsf{A} && \text{(Lemma 6.53)}\\
&= \mathsf{id}_\mathsf{A} && \text{(Lemma 6.51)}
\end{aligned}
$$

The other equalities are similar. □

6.4.4 Structural morphisms – formally

We now give alternate definitions of the structural morphisms.

Associator. For any arenas $\mathsf{A}, \mathsf{B}, \mathsf{C}$, we set the following renaming:

$$
\begin{array}{rlcl}
\mathbf{a}_{\mathsf{A},\mathsf{B},\mathsf{C}}: & (\mathsf{A} \otimes \mathsf{B}) \otimes \mathsf{C} & \to & \mathsf{A} \otimes (\mathsf{B} \otimes \mathsf{C})\\
& (1, (1, \mathsf{a})) & \mapsto & (1, \mathsf{a})\\
& (1, (2, \mathsf{b})) & \mapsto & (2, (1, \mathsf{b}))\\
& (2, \mathsf{c}) & \mapsto & (2, (2, \mathsf{c}))\,.
\end{array}
$$

We define the **associator** $\alpha_{\mathsf{A},\mathsf{B},\mathsf{C}}$ as:

$$\alpha_{\mathsf{A},\mathsf{B},\mathsf{C}} := \mathbf{a}_{\mathsf{A},\mathsf{B},\mathsf{C}} \rtimes \mathsf{id}_{(\mathsf{A}\otimes\mathsf{B})\otimes\mathsf{C}}\,.$$

Left-unitor. For any arena B, we set the following renaming:

$$
\begin{array}{rlcl}
\mathbf{l}_\mathsf{B}: & \mathsf{I} \otimes \mathsf{B} & \to & \mathsf{B}\\
& (2, \mathsf{b}) & \mapsto & \mathsf{b}\,.
\end{array}
$$

We define the **left-unitor** λ_B as:

$$\lambda_\mathsf{B} := \mathbf{l}_\mathsf{B} \rtimes \mathsf{id}_{\mathsf{I}\otimes\mathsf{B}}\,.$$

Right-unitor. For any arena A, we set the following renaming:

$$
\begin{array}{rlcl}
\mathbf{r}_\mathsf{A}: & \mathsf{A} \otimes \mathsf{I} & \to & \mathsf{A}\\
& (1, \mathsf{a}) & \mapsto & \mathsf{a}\,.
\end{array}
$$

We define the **right-unitor** ρ_A as:

$$\rho_\mathsf{A} := \mathbf{r}_\mathsf{A} \rtimes \mathsf{id}_{\mathsf{A}\otimes\mathsf{I}}\,.$$

Symmetry. For any arenas A, B, we set the following renaming:

$$\begin{array}{rlcl} \mathsf{s}_{\mathsf{A},\mathsf{B}}: & (\mathsf{A}\otimes\mathsf{B}) & \to & (\mathsf{B}\otimes\mathsf{A}) \\ & (1,\mathsf{a}) & \mapsto & (2,\mathsf{a}) \\ & (2,\mathsf{b}) & \mapsto & (1,\mathsf{b})\,. \end{array}$$

We define the **symmetry** $\gamma_{\mathsf{A},\mathsf{B}}$ as:

$$\gamma_{\mathsf{A},\mathsf{B}} := \mathsf{s}_{\mathsf{A},\mathsf{B}} \rtimes \mathsf{id}_{\mathsf{A}\otimes\mathsf{B}}\,.$$

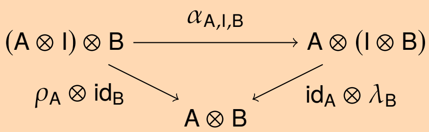
These morphisms behave like the intuitive definitions given in 6.4.2. We can now formally check they satisfy the conditions of Definitions 1.1 (monoidal category) and 1.2 (symmetric monoidal category).

Remark that all renamings are bijective, which implies that the structural morphisms are isomorphisms by Lemma 6.57.

We detail the triangle identity:

Lemma 6.58 – Triangle identity

For any arenas A, B, the following diagram commutes.



Proof. For any arenas A, B, we have:

$$\begin{array}{rlr}
& (\mathsf{id}_{\mathsf{A}}\otimes\lambda_{\mathsf{B}})\odot\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}} & \\
= & (\mathsf{id}_{\mathsf{A}}\otimes(\mathbf{l}_{\mathsf{B}}\rtimes\mathsf{id}_{\mathsf{I}\otimes\mathsf{B}}))\odot\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}} & (\text{Definition of } \lambda) \\
= & \left(\mathsf{id}_{\mathsf{A}}\otimes\left(\mathsf{id}_{\mathsf{B}}\ltimes\mathbf{l}_{\mathsf{B}}^{-1}\right)\right)\odot\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}} & (\text{Lemma 6.54}) \\
= & \left((\mathsf{id}_{\mathsf{A}}\ltimes\mathsf{id}_{\mathsf{A}})\otimes\left(\mathsf{id}_{\mathsf{B}}\ltimes\mathbf{l}_{\mathsf{B}}^{-1}\right)\right)\odot\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}} & (\text{Lemma 6.51}) \\
= & \left((\mathsf{id}_{\mathsf{A}}\otimes\mathsf{id}_{\mathsf{B}})\ltimes\left(\mathsf{id}_{\mathsf{A}}\times\mathbf{l}_{\mathsf{B}}^{-1}\right)\right)\odot\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}} & (\text{Lemma 6.56}) \\
= & (\mathsf{id}_{\mathsf{A}}\otimes\mathsf{id}_{\mathsf{B}})\odot((\mathsf{id}_{\mathsf{A}}\times\mathbf{l}_{\mathsf{B}})\rtimes\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}}) & (\text{Lemma 6.55}) \\
= & \mathsf{id}_{\mathsf{A}\otimes\mathsf{B}}\odot((\mathsf{id}_{\mathsf{A}}\times\mathbf{l}_{\mathsf{B}})\rtimes\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}}) & (\text{Lemma 6.38}) \\
= & (\mathsf{id}_{\mathsf{A}}\times\mathbf{l}_{\mathsf{B}})\rtimes\alpha_{\mathsf{A},\mathsf{I},\mathsf{B}} & (\text{Proposition 6.31}) \\
= & (\mathsf{id}_{\mathsf{A}}\times\mathbf{l}_{\mathsf{B}})\rtimes\left(\mathbf{a}_{\mathsf{A},\mathsf{I},\mathsf{B}}\rtimes\mathsf{id}_{(\mathsf{A}\otimes\mathsf{I})\otimes\mathsf{B}}\right) & (\text{Definition of } \alpha) \\
= & ((\mathsf{id}_{\mathsf{A}}\times\mathbf{l}_{\mathsf{B}})\circ\mathbf{a}_{\mathsf{A},\mathsf{I},\mathsf{B}})\rtimes\mathsf{id}_{(\mathsf{A}\otimes\mathsf{I})\otimes\mathsf{B}} & (\text{Lemma 6.52}) \\
= & (\mathbf{r}_{\mathsf{A}}\times\mathsf{id}_{B})\rtimes\mathsf{id}_{(\mathsf{A}\otimes\mathsf{I})\otimes\mathsf{B}} & (\star) \\
= & (\mathbf{r}_{\mathsf{A}}\times\mathsf{id}_{B})\rtimes(\mathsf{id}_{\mathsf{A}\otimes\mathsf{I}}\otimes\mathsf{id}_{\mathsf{B}}) & (\text{Lemma 6.38}) \\
= & (\mathbf{r}_{\mathsf{A}}\rtimes\mathsf{id}_{\mathsf{A}\otimes\mathsf{I}})\otimes(\mathsf{id}_{\mathsf{B}}\rtimes\mathsf{id}_{\mathsf{B}}) & (\text{Lemma 6.56}) \\
= & (\mathbf{r}_{\mathsf{A}}\rtimes\mathsf{id}_{\mathsf{A}\otimes\mathsf{I}})\otimes\mathsf{id}_{\mathsf{B}} & (\text{Lemma 6.54}) \\
= & \rho_{\mathsf{A}}\otimes\mathsf{id}_{\mathsf{B}} & (\text{Definition of } \rho)
\end{array}$$

where $(\star)$ is a direct computation of both functions. □

The other identities are very similar.

More generally, this construction

$$f \in \mathsf{Ren}(\mathsf{A},\mathsf{B}) \;\mapsto\; f \rtimes \mathsf{id}_\mathsf{A} \in \mathsf{PCG}(\mathsf{A}, f(\mathsf{A}))$$

is a strict monoidal functor between Ren and PCG, which means that the structural morphisms of PCG are simply obtained by transport from Ren. Hence all the coherence diagrams commute.

We still need to check all the morphisms are natural. We show the detailed proof for one diagram.

Lemma 6.59 – Naturality of λ

Consider arenas A, B. For any strategy $\sigma : \mathsf{A} \vdash \mathsf{B}$, we have

$$\sigma \odot \lambda_\mathsf{A} \;=\; \lambda_\mathsf{B} \odot (\mathsf{id}_\mathsf{I} \otimes \sigma)\ .$$

Proof. On the one hand, we have:

$$\begin{aligned} & \sigma \odot \lambda_\mathsf{A} \\ &= \sigma \odot (\mathbf{l}_\mathsf{A} \rtimes \mathsf{id}_{\mathsf{I}\otimes\mathsf{A}}) && \text{(Definition of } \lambda) \\ &= \left(\sigma \ltimes \mathbf{l}_\mathsf{A}^{-1}\right) \odot \mathsf{id}_{\mathsf{I}\otimes\mathsf{A}} && \text{(Lemma 6.55)} \\ &= \sigma \ltimes \mathbf{l}_\mathsf{A}^{-1} && \text{(Proposition 6.31)} \end{aligned}$$

and on the other hand:

$$\begin{aligned} & \lambda_\mathsf{B} \odot (\mathsf{id}_\mathsf{I} \otimes \sigma) \\ &= (\mathbf{l}_\mathsf{B} \rtimes \mathsf{id}_{\mathsf{I}\otimes\mathsf{B}}) \odot (\mathsf{id}_\mathsf{I} \otimes \sigma) && \text{(Definition of } \lambda) \\ &= \mathbf{l}_\mathsf{B} \rtimes (\mathsf{id}_{\mathsf{I}\otimes\mathsf{B}} \odot (\mathsf{id}_\mathsf{I} \otimes \sigma)) && \text{(Lemma 6.53)} \\ &= \mathbf{l}_\mathsf{B} \rtimes (\mathsf{id}_\mathsf{I} \otimes \sigma) && \text{(Proposition 6.31)} \end{aligned}$$

Consider an isogmentation $\mathsf{q} \in \mathsf{supp}(\sigma \ltimes \mathbf{l}_\mathsf{A}^{-1})$, then it is of the form $\mathsf{q} = \mathsf{q}' \ltimes \mathbf{l}_\mathsf{A}^{-1}$ with $\mathsf{q}' \in \mathsf{supp}(\sigma)$ and it appears in $\sigma \ltimes \mathbf{l}_\mathsf{A}^{-1}$ with the coefficient $\sigma(\mathsf{q}')$. But a direct computation yields:

$$\mathsf{q}' \ltimes \mathbf{l}_\mathsf{A}^{-1} \;=\; \mathbf{l}_\mathsf{B} \rtimes (0 \otimes \mathsf{q}')\ .$$

Hence, recalling that $\mathsf{id}_\mathsf{I} = 1 \cdot 0$ (with 0 the empty isogmentation), we obtain:

$$\begin{aligned} \mathbf{l}_\mathsf{B} \rtimes (\mathsf{id}_\mathsf{I} \otimes \sigma) &= \sum_{\mathsf{q}\in\mathsf{supp}(\sigma)} \sigma(\mathsf{q}) \cdot (\mathbf{l}_\mathsf{B} \rtimes (0 \otimes \mathsf{q}')) \\ &= \sum_{\mathsf{q}\in\mathsf{supp}(\sigma)} \sigma(\mathsf{q}) \cdot \left(\mathsf{q} \ltimes \mathbf{l}_\mathsf{A}^{-1}\right) \\ &= \sigma \ltimes \mathbf{l}_\mathsf{A}^{-1}\ . \end{aligned}$$ □

Again, the other naturality diagrams are similar.

We can now state our first theorem specifying the categorical structure of PCG.

Theorem 6.60 – PCG is a SMC

$(\mathsf{PCG}, \otimes, \mathsf{I})$ is a symmetric monoidal category.

6.4.5 Closed structure

Recall the currying bijection Λ introduced in Subsection 5.3.4.

Definition 6.61 – Currying strategies

Consider arenas G, A and B. For any $\sigma : (\mathsf{G} \otimes \mathsf{A}) \vdash \mathsf{B}$, we set

$$\Lambda_{\mathsf{G},\mathsf{A},\mathsf{B}}(\sigma) \overset{\text{def}}{=} \sum_{\mathsf{q} \in \mathsf{Isog}(\mathsf{G}\otimes\mathsf{A}\vdash\mathsf{B})} \sigma(\mathsf{q}) \cdot \Lambda^{\mathsf{Isog}}_{\mathsf{G},\mathsf{A},\mathsf{B}}(\mathsf{q}) \,.$$

This definition directly yields:

$$\Lambda_{\mathsf{G},\mathsf{A},\mathsf{B}} : \mathsf{PCG}(\mathsf{G} \otimes \mathsf{A}, \mathsf{B}) \;\cong\; \mathsf{PCG}(\mathsf{G}, \mathsf{A} \Rightarrow \mathsf{B}) \,.$$

Moreover, we have the following property:

Lemma 6.62 – Λ preserves composition

Consider arenas $\mathsf{G}, \mathsf{G}', \mathsf{A}, \mathsf{B}$ and strategies $\sigma : \mathsf{G}\otimes\mathsf{A} \vdash \mathsf{B}$ and $\tau : \mathsf{G}' \vdash \mathsf{G}$. Then, we have:

$$\Lambda_{\mathsf{G},\mathsf{A},\mathsf{B}}(\sigma) \odot \tau \;=\; \Lambda_{\mathsf{G}',\mathsf{A},\mathsf{B}}\,(\sigma \odot (\tau \otimes \mathsf{id}_{\mathsf{A}})) \,.$$

Proof. Computation of both sides of the equation, following the definitions. □

From this, we obtain the **evaluation morphism** with:

$$\mathbf{ev}_{\mathsf{A},\mathsf{B}} \overset{\text{def}}{=} \Lambda^{-1}_{\mathsf{A}\Rightarrow\mathsf{B},\mathsf{A},\mathsf{B}}\,(\mathsf{id}_{\mathsf{A}\Rightarrow\mathsf{B}}) \,.$$

Altogether, these give us the closed structure of PCG.

Theorem 6.63 – PCG is a SMCC

$(\mathsf{PCG}, \otimes, \mathsf{I})$ is a symmetric monoidal closed category.

Proof. PCG is a SMC by Theorem 6.60. For any arenas $\mathsf{G}, \mathsf{A}, \mathsf{B}$ and morphisms $\sigma : \mathsf{G} \otimes \mathsf{A} \vdash \mathsf{B}$ and $\tau : \mathsf{G} \vdash \mathsf{A} \Rightarrow \mathsf{B}$, we have the following equalities:

$$\mathsf{ev}_{\mathsf{A},\mathsf{B}} \odot (\Lambda_{\mathsf{G},\mathsf{A},\mathsf{B}}(\sigma) \otimes \mathsf{id}_{\mathsf{A}}) \;=\; \sigma \qquad (6.4)$$

$$\Lambda_{\mathsf{G},\mathsf{A},\mathsf{B}}\,(\mathsf{ev}_{\mathsf{A},\mathsf{B}} \odot (\tau \otimes \mathsf{id}_{\mathsf{A}})) \;=\; \tau \qquad (6.5)$$

which follow from a direct computation. □

6.5 From qualitative PCG to HO

We show that the earlier isomorphisms between isogmentations and quotiented plays extend to strategies and composition. **For this section, we consider the qualitative version of PCG, where strategies are *sets* of isogmentations without coefficients.**

Recall that we have the following isomorphism (Figure 3.17):

$$\mathsf{VisPlays}^+(\mathsf{A})_{/\sim_E} \underset{\mathbf{Plays}(-)}{\overset{\mathbf{isog}(-)}{\rightleftarrows}} \mathsf{Isog}(\mathsf{A}) \qquad (\cong)$$

for plays quotiented by Mellies' homotopy equivalence and isogmentations. Furthermore, we defined *Meagre Innocent Isogmentations* as −-linear isogmentations (Definition 3.29), and we saw that these were in bijection with innocent strategies in HO (Figure 3.23):

$$\mathsf{HO}_f^{\mathsf{Inn}}(\mathsf{A}) \underset{\mathsf{HOstrat}(-)}{\overset{\mathsf{MII}(-)}{\rightleftarrows}} \mathbf{MII}(\mathsf{A}) \qquad (\cong)$$

Finally, we defined *Fat Innocent Isogmentations* (Definition 3.45), which are the isoexpansions of a mii, corresponding to the set of plays of an innocent strategy in HO (Figure 3.25):

$$\begin{array}{ccc} \mathsf{HO}_f^{\mathsf{Inn}}(\mathsf{A}) & \xrightarrow{\mathsf{MII}(-)} & \mathbf{MII}(\mathsf{A}) \\ & \searrow^{\mathbf{isog}(-)} & \downarrow \mathsf{iexp}(-) \\ & & \mathbf{FII}(\mathsf{A}) \end{array}$$

Remark: We present the constructions for *finite* innocent strategies in HO; one could consider ∞-*isogmentations* for the general case.

Now, we want all these isomorphisms to still preserve the categorical structure of PCG: in particular, that the identities coincide, and that the composition is compatible with the isomorphism **Plays**(−).

6.5.1 Arrowing

In Chapter 3, we studied the link between plays and isogmentations for a *fixed* −-arena A. But what happens when we want to consider a *strategy* $\sigma\colon \mathsf{A} \vdash \mathsf{B}$, for A, B −-arenas? Since $\mathsf{A} \vdash \mathsf{B}$ is not negative, it is not an HO arena, hence we need first to turn $\sigma\colon \mathsf{A} \vdash \mathsf{B}$ into a strategy $\Lambda^{\Rightarrow}(\sigma)\colon \mathsf{A} \Rightarrow \mathsf{B}$.

This is a particular case of the curryfication.

Definition 6.64 – *Arrowing* of augmentations

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$. We define $\Lambda^{\Rightarrow}(q)$ with:

$$\begin{array}{rcl} |\Lambda^{\Rightarrow}(q)| & = & |q| \\ a \leq_{(\!|\Lambda^{\Rightarrow}(q)|\!)} b & \text{iff} & \left(a \leq_{(\!|q|\!)} b\right) \text{ or } (a = \mathsf{init}(b)) \\ a \leq_{\Lambda^{\Rightarrow}(a)} b & \text{iff} & a \leq_q b \\ \partial_{\Lambda^{\Rightarrow}(q)}(a) & = & \begin{cases} (2, \mathsf{b}) & \text{if } \partial_q(a) = (2, \mathsf{b})\,, \\ (1, (\mathsf{b}, \mathsf{c})) & \text{if } \partial_q(a) = (1, \mathsf{c}) \\ & \text{and } \partial_q(\mathsf{init}(a)) = (2, \mathsf{b})\,. \end{cases} \end{array}$$

Then $\Lambda^{\Rightarrow}(q) \in \mathsf{Aug}(\mathsf{A} \Rightarrow \mathsf{B})$.

Proof. Clear by definition and Lemma 5.13. □

Proposition 6.65 – Arrowing isomorphism

We have an isomorphism

$$\Lambda^{\Rightarrow} : \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B}) \cong \mathsf{Aug}(\mathsf{A} \Rightarrow \mathsf{B})\,.$$

We write $\Lambda^{\vdash}$ for the reverse isomorphism.

This construction clearly preserves isomorphism, hence we define

$$\Lambda^{\Rightarrow}(\mathsf{q}) \stackrel{\text{def}}{=} \overline{\Lambda^{\Rightarrow}(\underline{\mathsf{q}})}$$

for any $\mathsf{q} \in \mathsf{Isog}(\mathsf{A} \vdash \mathsf{B})$.

Finally we extend $\Lambda^{\Rightarrow}(-)$ to strategies[2] with, for any $\sigma : \mathsf{A} \vdash \mathsf{B}$,

$$\Lambda^{\Rightarrow}(\sigma) \stackrel{\text{def}}{=} \{\Lambda^{\Rightarrow}(\mathsf{q}) \mid \mathsf{q} \in \sigma\}\,.$$

2: Recall that we consider the qualitative version of PCG here, hence why we define $\Lambda^{\Rightarrow}(\sigma)$ as a set of isogmentations.

6.5.2 $\mathsf{Plays}^{\Rightarrow}(-)$ and innocent strategies

Now we can define the plays of strategies on $\mathsf{A} \vdash \mathsf{B}$.

Definition 6.66 – $\mathsf{Plays}^{\Rightarrow}(-)$

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$, then we define

$$\mathsf{Plays}^{\Rightarrow}(\sigma) \stackrel{\text{def}}{=} \mathsf{Plays}(\Lambda^{\Rightarrow}(\sigma))\,.$$

Reminder: For $q \in \mathsf{Aug}(\mathsf{A})$, we have:

$$\mathsf{Plays}(q) \stackrel{\text{def}}{=} \{\partial_q(\mathsf{t}) \mid \mathsf{t} \in \mathsf{Alt}(q)\}$$

(see Definition 3.24). For $\mathsf{q} \in \mathsf{Isog}(\mathsf{A})$:

$$\mathsf{Plays}(\mathsf{q}) \stackrel{\text{def}}{=} \mathsf{Plays}(\underline{\mathsf{q}})$$

and finally for any $\sigma \subseteq \mathsf{Isog}(\mathsf{A})$:

$$\mathsf{Plays}(\sigma) \stackrel{\text{def}}{=} \bigcup_{\mathsf{q} \in \sigma} \mathsf{Plays}(\mathsf{q})\,.$$

Be careful: in general $\mathsf{Plays}^{\Rightarrow}(\sigma)$ **is not** a strategy in HO! Indeed, strategies in PCG are sets of isogmentations, and there is no condition of non-emptyness, prefix closure, or determinism.

As before, we need to consider *innocent* strategies. Recall that innocence in PCG is characterized by being a FII *i.e.* the set of isoexpansions of a −-linear isogmentation (Definition 3.45). Thankfully, $\Lambda^{\Rightarrow}(-)$ preserves this property.

Lemma 6.67 – $\Lambda^{\Rightarrow}(-)$ preserves FIIs

Consider $\sigma : \mathsf{A} \vdash \mathsf{B}$. Then,

$$\sigma \in \mathbf{FII}(\mathsf{A} \vdash \mathsf{B}) \quad \Leftrightarrow \quad \Lambda^{\Rightarrow}(\sigma) \in \mathbf{FII}(\mathsf{A} \Rightarrow \mathsf{B}) \,.$$

Proof. **Only if.** $\Lambda^{\Rightarrow}(-)$ preserves isomorphisms, so if $\sigma = \mathsf{iexp}(\mathsf{q})$ for some $-$-linear q, then $\Lambda^{\Rightarrow}(\mathsf{q})$ is $-$-linear and $\Lambda^{\Rightarrow}(\sigma) = \mathsf{iexp}(\Lambda^{\Rightarrow}(\mathsf{q}))$.

If. Likewise, if $\Lambda^{\Rightarrow}(\sigma) = \mathsf{iexp}(\mathsf{p})$ for some $-$-linear p, then $\Lambda^{\vdash}(\mathsf{p})$ is $-$-linear and $\sigma = \mathsf{iexp}(\Lambda^{\vdash}(\mathsf{p}))$. □

Hence all the isomorphisms between PCG and HO presented in Chapter 3 still stand.

6.5.3 Identities

We now prove that $\mathsf{Plays}^{\Rightarrow}(-)$ preserves identities.

Proposition 6.68 – $\mathsf{Plays}^{\Rightarrow}(-)$ preserves identities.

Consider an arena A. Then,

$$\mathsf{Plays}^{\Rightarrow}(\mathsf{id}_{\mathsf{A}}) = \mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}} \,.$$

Proof. We start with the inclusion $\mathsf{Plays}^{\Rightarrow}(\mathsf{id}_{\mathsf{A}}) \subseteq \mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}}$.

Consider $s \in \mathsf{Plays}^{\Rightarrow}(\mathsf{id}_{\mathsf{A}})$. Unfolding the definitions, we have:

$$\begin{aligned}
\mathsf{Plays}^{\Rightarrow}(\mathsf{id}_{\mathsf{A}}) &= \mathsf{Plays}(\Lambda^{\Rightarrow}(\mathsf{id}_{\mathsf{A}})) \\
&= \bigcup_{\mathsf{q} \in \Lambda^{\Rightarrow}(\mathsf{id}_{\mathsf{A}})} \mathsf{Plays}(\mathsf{q}) \\
&= \bigcup_{\mathsf{q} \in \mathsf{id}_{\mathsf{A}}} \mathsf{Plays}(\Lambda^{\Rightarrow}(\mathsf{q})) \\
&= \bigcup_{\mathsf{x} \in \mathsf{Pos}(\mathsf{A})} \mathsf{Plays}(\Lambda^{\Rightarrow}(\mathbb{c}_{\mathsf{x}})) \\
&= \bigcup_{\mathsf{x} \in \mathsf{Pos}(\mathsf{A})} \left\{ \partial_{\underline{\Lambda^{\Rightarrow}(\mathbb{c}_{\mathsf{x}})}}(\mathsf{t}) \mid \mathsf{t} \in \mathsf{Alt}(\underline{\Lambda^{\Rightarrow}(\mathbb{c}_{\mathsf{x}})}) \right\} .
\end{aligned}$$

Consider $\mathsf{x} \in \mathsf{Pos}(\mathsf{A})$, $q := \underline{\Lambda^{\Rightarrow}(\mathbb{c}_{\mathsf{x}})}$ and $\mathsf{t} := \mathsf{t}_1 \dots \mathsf{t}_n \in \mathsf{Alt}(q)$ such that $s = \partial_q(\mathsf{t})$. We know that $s \in \mathsf{Plays}(\mathsf{A} \Rightarrow \mathsf{A})$.

Let us first prove condition (1) of the definition of $\mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}}$. We prove by induction on s' that for any $s' \sqsubseteq^+ s$, we have $s' \upharpoonright \mathsf{A}_\ell = s' \upharpoonright \mathsf{A}_r$, where we use indices to distinguish between the two copies of the arena A.

Reminder: (Definition 2.22) $s \in \mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}}$ iff:

1. $\forall s' \sqsubseteq^+ s,\ s' \upharpoonright \mathsf{A}_\ell = s' \upharpoonright \mathsf{A}_r$,
2. if s_i^-, s_{i+1}^+ are minimal in A, then s_{i+1} points to s_i.

The equality is clear on the empty case, so consider $s' s_i^- s_{i+1}^+ \sqsubseteq^+ s$. By induction, $s' \upharpoonright \mathsf{A}_\ell = s' \upharpoonright \mathsf{A}_r$. Moreover, we know that $\mathsf{t}_i^- \rightarrowtriangle_{\mathsf{t}} \mathsf{t}_{i+1}^+$, so by Lemma 3.22 we have $\mathsf{t}_i \rightarrowtriangle_q \mathsf{t}_{i+1}$. But $\Lambda^{\Rightarrow}$ preserves the causal order, so we also have $\mathsf{t}_i \rightarrowtriangle_{\Lambda^{\vdash}(q)} \mathsf{t}_{i+1}$. Since $\Lambda^{\vdash}(q) \cong \mathbb{c}_{\underline{\mathsf{x}}}$, we use Lemma 6.28 to conclude that t_i and t_{i+1} correspond to the same event in A, but from both side of the arena $\mathsf{A} \Rightarrow \mathsf{A}$, thus proving that $s' s_i^- s_{i+1}^+ \upharpoonright \mathsf{A}_\ell = s' s_i^- s_{i+1}^+ \upharpoonright \mathsf{A}_r$ (where the pointers are also equal, by definition of $\leq_{(\!|\mathbb{c}_{\underline{\mathsf{x}}}|\!)}$).

Now, we prove the second condition of the definition of $\mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}}$. Consider s_i^-, s_{i+1}^+ minimum in A, by negativity of A we have

$$\partial_q(\mathsf{t}_i^-) = (2, \mathsf{a}) \quad \text{and} \quad \partial_q(\mathsf{t}_{i+1}^+) = (1, (\mathsf{a}, \mathsf{a})) \quad \text{with } \mathsf{a} \in \min(\mathsf{A}).$$

So t_i is minimal for $\leq_{(\!|q|\!)}$; and thus for $\leq_q$ by minimality-preservation. Moreover, by Lemma 3.22, we have $\mathsf{t}_i^- \rightarrowtriangle_q \mathsf{t}_{i+1}^+$. Since $\Lambda^{\Rightarrow}$ preserves the causal order, we actually have

$$\mathsf{t}_i \in \min(\leq_{\Lambda^{\vdash}(q)}) \text{ and } \mathsf{t}_i \rightarrowtriangle_{\Lambda^{\vdash}(q)} \mathsf{t}_{i+1} ,$$

so by Lemma 5.13 we obtain $\mathsf{t}_i = \mathsf{init}(\mathsf{t}_{i+1})$. Thus, by definition of $\Lambda^{\Rightarrow}$, we have

$$\mathsf{t}_i \rightarrowtriangle_{(\!|q|\!)} \mathsf{t}_{i+1} ,$$

and s_{i+1} points to s_i as needed.

Hence, $\mathsf{Plays}^{\Rightarrow}(\mathsf{id}_{\mathsf{A}}) \subseteq \mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}}$. We now prove the reverse inclusion.

Consider $s \in \mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}}$. We need to find $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{A})$ such that

$$\overline{q} \in \mathsf{id}_{\mathsf{A}} \quad \text{and} \quad s \in \mathsf{Plays}^{\Rightarrow}(q) .$$

Consider $q := \Lambda^{\vdash}(\mathsf{aug}(s))$. Then it is clear that:

$$q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{A}) \quad \text{and} \quad s \in \mathsf{Plays}^{\Rightarrow}(q) ,$$

so we only have to check that $\overline{q} \in \mathsf{id}_{\mathsf{A}}$, *i.e.* we want $\overline{q} = \mathbb{c}_{\mathsf{x}}$ for some $\mathsf{x} \in \mathsf{Pos}(\mathsf{A})$. Since $s \in \mathsf{Plays}(\mathsf{A} \Rightarrow \mathsf{A})$, the restriction $s \upharpoonright \mathsf{A}_{r}$ informs a position $\mathsf{x} \in \mathsf{Pos}(\mathsf{A})$. Indeed, consider the configuration x with:

$$\begin{array}{rcl} |x| & := & \{1, \ldots, p\} \text{ with } p \text{ the lenght of } s \upharpoonright \mathsf{A}_{r} , \\ i \rightarrowtriangle_x j & \text{iff} & (s \upharpoonright \mathsf{A}_{r})_j \text{ points to } (s \upharpoonright \mathsf{A}_{r})_i . \end{array}$$

Then $x \in \mathsf{Conf}(\mathsf{A})$, and we set $\mathsf{x} := \overline{x}$. We show that $q \cong \mathbb{c}_x$, with the isomorphism:

$$\begin{array}{rcl} \xi : |q| & \to & |\mathbb{c}_x| \\ i & \mapsto & (\jmath, j) \text{ s.t. } s_i \in (s \upharpoonright \mathsf{A}_{\jmath}) \text{ and } s_i \text{ has the index } j \text{ in } s \upharpoonright \mathsf{A}_{\jmath} . \end{array}$$

The isomorphism between event sets, static orders and display maps is clear from the fact that for any $s' \sqsubseteq^+ s$, $s' \upharpoonright \mathsf{A}_{\ell} = s' \upharpoonright \mathsf{A}_{r}$. For the causal order, we have:

(a) Let us prove that $i^- \rightarrowtriangle_q k^+$ iff $k = i + 1$. We know that s_i and s_{i+1} correspond to the same event in both sides of $\mathsf{A} \Rightarrow \mathsf{A}$. Hence, if $\xi(i) = (\jmath, j)$, then $\xi(i+1) = (\jmath', j)$ with $\jmath' \neq \jmath$, and $\xi(i) \rightarrowtriangle_{\mathbb{c}_x} \xi(i+1)$ by Lemma 6.28. Reciprocally, if $(\jmath, j)^- \rightarrowtriangle_{\mathbb{c}_x} (\jmath', j')^+$, then by Lemma 6.28 $\jmath \neq \jmath'$ and $j = j'$, and clearly $\xi^{-1}(\jmath', j) = \xi^{-1}(\jmath, j) + 1$.

(b) Let us prove that $i^+ \rightarrowtriangle_q k^-$ iff s_k points to s_i. Since both restrictions of s to A_{ℓ} and A_{r} are plays, s_k points to s_i implies that both moves are played in the same side $\mathsf{A}_{\jmath}$ of $\mathsf{A} \Rightarrow \mathsf{A}$. So $\xi(i) = (\jmath, j)$ and $\xi(k) = (\jmath, j')$; and the pointers are preserved, so $(\jmath, j) \rightarrowtriangle_{\mathbb{c}_x} (\jmath, j')$. Reciprocally, $(\jmath, j) \rightarrowtriangle_{\mathbb{c}_x} (\jmath', j')$ iff $\jmath = \jmath'$ and $j \rightarrowtriangle_x j'$, *i.e.* $\xi^{-1}((\jmath, j)) \rightarrowtriangle_q \xi^{-1}((\jmath, j'))$. □

6.5.4 Composition

We now prove that $\mathsf{Plays}^{\Rightarrow}(-)$ is compatible with the composition.

Proposition 6.69 – $\mathsf{Plays}^{\Rightarrow}(-)$ and composition

Consider A, B and C arenas; and two strategies $\sigma : \mathsf{A} \vdash \mathsf{B}$ and $\tau : \mathsf{B} \vdash \mathsf{C}$. Then,

$$\mathsf{Plays}^{\Rightarrow}(\tau \odot \sigma) = \mathsf{Plays}^{\Rightarrow}(\tau) \odot^{\mathsf{HO}} \mathsf{Plays}^{\Rightarrow}(\sigma) .$$

One of the inclusion is easy: given an interaction between two plays, we can construct an isomorphism between the corresponding augmentations.

Lemma 6.70 – $\mathsf{Plays}^{\Rightarrow}(-)$ and composition, part 1

Consider A, B and C arenas with $\sigma : \mathsf{A} \vdash \mathsf{B}$ and $\tau : \mathsf{B} \vdash \mathsf{C}$. Then:

$$\mathsf{Plays}^{\Rightarrow}(\tau) \odot^{\mathsf{HO}} \mathsf{Plays}^{\Rightarrow}(\sigma) \subseteq \mathsf{Plays}^{\Rightarrow}(\tau \odot \sigma) .$$

Proof. Consider $s \in \mathsf{Plays}^{\Rightarrow}(\tau) \odot^{\mathsf{HO}} \mathsf{Plays}^{\Rightarrow}(\sigma)$. Then $s \in s^{\tau} \odot^{\mathsf{HO}} s^{\sigma}$ for some $s^{\tau} \in \mathsf{Plays}^{\Rightarrow}(\tau)$ and $s^{\sigma} \in \mathsf{Plays}^{\Rightarrow}(\sigma)$. In other words, there exists an interaction $u \in \mathrm{I}(\mathsf{A}, \mathsf{B}, \mathsf{C})$ such that:

$$u \upharpoonright \mathsf{A}, \mathsf{B} = s^{\sigma} \qquad u \upharpoonright \mathsf{B}, \mathsf{C} = s^{\tau} \qquad u \upharpoonright \mathsf{A}, \mathsf{C} = s .$$

Hence there exists $q \in \mathsf{q} \in \Lambda^{\Rightarrow}(\sigma)$ (resp. $p \in \mathsf{p} \in \Lambda^{\Rightarrow}(\tau)$) with the linearisation t^{σ} (resp. t^{τ}), such that:

$$s^{\sigma} = \partial_q(\mathsf{t}^{\sigma}) \qquad \text{and} \qquad s^{\tau} = \partial_p(\mathsf{t}^{\tau}) .$$

Moreover, the events occuring in B in s^{σ} and in s^{τ} are compatible, and we can define an isomorphism $\varphi : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$ through u. It is clear that such a φ is a configuration isomorphism, since the constructions $\partial_q(\mathsf{t}^{\sigma})$ and $\partial_p(\mathsf{t}^{\tau})$ preserve pointers and arena image. So, we can consider $r = \Lambda^{\vdash}(p) \odot_{\varphi} \Lambda^{\vdash}(q)$, and we have $s \in \mathsf{Alt}(r)$. □

The other inclusion is not so easy, because we need to build a sequential interaction from the isomorphism. In other words, given an alternating play $s \in \mathsf{Plays}^{\Rightarrow}(\tau \odot \sigma)$, we must prove that s can only come from the composition of two alternating plays in $\mathsf{Plays}^{\Rightarrow}(\sigma)$ and $\mathsf{Plays}^{\Rightarrow}(\tau)$. The problem is that we only know that s is obtained from the composition of some augmentation in σ and some augmentation in τ, but it is not clear how we can linearise them and construct an interaction.

To do so, we need additional lemmas on polarity, and the notion of *states* of augmentations and interactions.

Definition 6.71 – State of an augmentation

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$. A **state** of q is $X \subseteq |q|$ which is down-closed for $\leq_q$.

Remark that a state of $q \in \mathsf{Aug}(\mathsf{A})$ is almost an augmentation on A (with orders and display map inherited from q), save for the +-coveredness condition which might not be respected. Thus the definition of alternating linearisations can easily be extended to states.

Definition 6.72 – Alternating linearisation of a state

Consider $q \in \mathsf{Aug}(\mathsf{A})$ with a state X. An **alternating linearisation** of X is a total order on the events of X, noted $\mathsf{t} = \mathsf{t}_1 \dots \mathsf{t}_n$ with $\{\mathsf{t}_i \mid 1 \leq i \leq n\} = X$, such that:

$$\begin{array}{ll} \textit{polarity-alternating:} & \forall i < n,\ \mathsf{pol}(\mathsf{t}_i) \neq \mathsf{pol}(\mathsf{t}_{i+1})\,. \\ \textit{causality-respecting:} & \forall i < n,\ \mathsf{t}_i \leq_q \mathsf{t}_{i+1}\,. \end{array}$$

We write $\mathsf{Alt}(X)$ for the set of alternating linearisations of X.

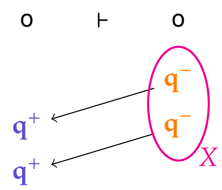


Figure 6.10: An augmentation q with a state X.

Notation: If $X \subseteq |q|$, we write:

$$|X|^- := \{a \in X \mid \mathsf{pol}_q(a) = -\}\,,$$
$$|X|^+ := \{a \in X \mid \mathsf{pol}_q(a) = +\}\,.$$

However, because states are not always +-covered, they might not have alternating linearisation! Take the augmentation on Figure 6.10 for instance, with the state X. The linearisation obviously fails because X has two negative events and no positive event.

In general, we can observe that an alternative linearisation must either have the same number of positive and negative events, or just one more negative event (because the arenas are negative, so the first event is always negative). This leads us to define O-states and P-states.

Definition 6.73 – O-states and P-state

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$.

An **O-state** of q is a state X such that $\sharp|X|^- = \sharp|X|^+$.

A **P-state** of q is a state X such that $\sharp|X|^- = \sharp|X|^+ + 1$

Not only is it clear that a state accepting an alternating linearisation must be an O-state or a P-state, depending on its size; but we can actually prove that *all* O-states and P-states have alternating linearisations.

Lemma 6.74 – Linearisation of states

Consider an augmentation $q \in \mathsf{Aug}(\mathsf{A})$ with a state X. Then:

- X is an **O-state** if and only if it has an even-length alternating linearisation.
- X is a **P-state** if and only if it has an odd-length alternating linearisation.

Proof. We construct an alternating linearisation inductively on the size of the state, proving that all O/P-states have alternating linearisations.

If $\sharp X = 0$. Immediate.

If $\sharp X = 2n + 1$. Then X is a P-state, and $\sharp|X|^- = \sharp|X|^+ + 1$. For any event $a \in X$, we call **successors in** X of a the events of X immediately following a in $\leq_q$, noted $\mathsf{succ}_X(a)$. We know that every negative event of X has at most one successor by determinism, so since $\sharp|X|^- = \sharp|X|^+ + 1$, every negative event has exactly one successor *except one*, which we will call a. Since a has no successor, it is maximal in X. Thus we can remove it, and $X\backslash\{a\}$ is still down-closed; so it is an O-state of q of size $2n$. By induction hypothesis, we construct an even-length alternating linearisation of $X\backslash\{a\}$, and then add a at the end of the linearisation.

If $\sharp X = 2n + 2$. Then X is an O-state, and $\sharp|X|^- = \sharp|X|^+ \leq 1$. Hence any negative event has *exactly one* successor. Take any event b^+ positive maximal in X. Then $X\backslash\{b\}$ is still down-closed, so by induction hypothesis it is a P-state of q of size $2n + 1$. Thus it has an alternating linearization, which must end with a negative event a^-. Moreover, this last event is the only negative event of $X\backslash\{b\}$ without a successor. Hence $b \in \mathsf{succ}_X(a)$, and we can add b at the end of the linearisation. □

We now define states for the interaction of two augmentations. Consider two augmentations $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$ and $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, with the isomorphism $\varphi\colon x_{q\restriction\mathsf{B}} \cong_{\mathsf{B}} x_{p\restriction\mathsf{B}}$. For any $X \subseteq |p \circledast_\varphi q|$, we write:

$$
\begin{aligned}
X \restriction \mathsf{A},\mathsf{C} &= \{(1,e) \in X \mid \partial_q(e) = (1,\mathsf{a})\} \cup \{(2,e) \in X \mid \partial_p(e) = (2,\mathsf{c})\},\\
X \restriction \mathsf{A},\mathsf{B}_q &= \{e \mid (1,e) \in X\},\\
X \restriction \mathsf{B}_q,\mathsf{B}_p &= \{(1,e) \in X \mid \partial_q(e) = (2,\mathsf{b})\} \cup \{(2,e) \in X \mid \partial_p(e) = (1,\mathsf{b})\},\\
X \restriction \mathsf{B}_p,\mathsf{C} &= \{e \mid (2,e) \in X\}.
\end{aligned}
$$

Then, we have:

$$
X \restriction \mathsf{A},\mathsf{C} \subseteq |p \odot_\varphi q| \qquad X \restriction \mathsf{A},\mathsf{B}_q \subseteq |q| \qquad X \restriction \mathsf{B}_p,\mathsf{C} \subseteq |p|\,.
$$

We say that $X \restriction \mathsf{B}_q,\mathsf{B}_p$ **is a state of** φ if it is down-closed for $\lhd^*$ restricted to events occuring in B; in that case we define O-states and P-states as in Definition 6.73, following the polarities of $\mathsf{B}_q \vdash \mathsf{B}_p$.

A subset of an interaction $X \subseteq |p \circledast_\varphi q|$ can thus yield up to four different states: a (potential) state of $p \odot_\varphi q$, one of q, one of φ and one of p – and each of these states can be an O/P-state.

Definition 6.75 – KLMN-states of an interaction

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, and $\varphi\colon x_{q\restriction\mathsf{B}} \cong_{\mathsf{B}} x_{p\restriction\mathsf{B}}$.

A **KLMN-state** of $p \circledast_\varphi q$, for K,L,M,N ∈ {O,P}, is $X \subseteq |p \circledast_\varphi q|$ down-closed for $\lhd^*$, such that:

1. $X \restriction \mathsf{A},\mathsf{C}$ is a K-state of $p \odot_\varphi q$,
2. $X \restriction \mathsf{A},\mathsf{B}_q$ is a L-state of q,
3. $X \restriction \mathsf{B}_q,\mathsf{B}_p$ is a M-state of φ,
4. $X \restriction \mathsf{B}_p,\mathsf{C}$ is a N-state of p,

Thanks to these polarities we can finally describe the linearisation needed for the proof of Proposition 6.69.

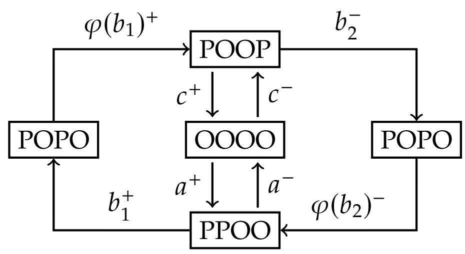


Figure 6.11: Polarities of an interaction

Lemma 6.76 – Polarities of an interaction

Consider $q \in \mathsf{Aug}(\mathsf{A} \vdash \mathsf{B})$, $p \in \mathsf{Aug}(\mathsf{B} \vdash \mathsf{C})$, $\varphi : x_{q \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{p \upharpoonright \mathsf{B}}$, and X a KLMN-state of $p \circledast_\varphi q$, with t an alternating linearisation of $X \upharpoonright \mathsf{A}, \mathsf{C}$.

Then X has a linearisation t following the diagram of Figure 6.11 (where a^+ is an event occuring in A and with polarity $+$ in A), such that $t \upharpoonright \mathsf{A}, \mathsf{C} = \mathsf{t}$, and we are in one of the following cases:

1. $X \upharpoonright \mathsf{A}, \mathsf{C}$ is an O-state. Then X is an OOOO-state.
2. $X \upharpoonright \mathsf{A}, \mathsf{C}$ is a P-state. Then we have three cases:
   a) X is a PPOO-state.
   b) X is a POPO-state.
   c) X is a POOP-state.

Proof. We prove the lemma by induction on the size of X. If X is empty, it is an OOOO-state. Otherwise:

1. If $X \upharpoonright \mathsf{A}, \mathsf{C}$ is an O-state, consider $e = \max(\mathsf{t})$. Since $X \upharpoonright \mathsf{A}, \mathsf{C}$ is an O-state, e must be positive in $\mathsf{A} \vdash \mathsf{C}$. By courtesy, e is maximal in X. Indeed, assume e is not maximal in X, then there exists $b \in X$ such that $e \rightarrowtriangle_{\lhd^*} b$ with e^+ occuring in A or C and b^- occuring in B, contradiction. So $X \backslash \{e\} \upharpoonright \mathsf{A}, \mathsf{C}$ is a P-state, and by induction we are in the second case of the lemma.
   a) If e occurs in A, then $X \backslash \{e\} \upharpoonright \mathsf{A}, \mathsf{B}_q$ is a P-state, and by I.H. $X \backslash \{e\}$ is a PPOO-state. Hence, X is an OOOO-state.
   b) If e occurs in C, then $X \backslash \{e\} \upharpoonright \mathsf{B}_p, \mathsf{C}$ is a P-state, and by H.I. $X \backslash \{e\}$ is a POOP-state. Hence, X is an OOOO-state.
2. If $X \upharpoonright \mathsf{A}, \mathsf{C}$ is a P-state, consider $e = \max(\mathsf{t})$.
   a) If e is also maximal in X and occurs in A, then e is negative in $\mathsf{A} \vdash \mathsf{C}$, *i.e.* positive in A, and $X \backslash \{e\} \upharpoonright \mathsf{A}, \mathsf{C}$ is an O-state. By induction hypothesis, $X \backslash \{e\}$ is an OOOO-state. So X is a PPOO-state.
   b) If e is also maximal in X and occurs in C, then e is negative in $\mathsf{A} \vdash \mathsf{C}$, *i.e.* negative in C, and $X \backslash \{e\} \upharpoonright \mathsf{A}, \mathsf{C}$ is an O-state. By induction hypothesis, $X \backslash \{e\}$ is an OOOO-state. So X is a POOP-state.
   c) If e isn't maximal in X, then there exists a b maximal in X, such that $e \lhd^* b$ and b occurs in X.
      i. If b is negative in B and occurs in B_q, then it is

negative in $\mathsf{A} \vdash \mathsf{B}$ and positive in $\mathsf{B}_q \vdash \mathsf{B}_p$. So, by induction hypothesis, $X\backslash\{b\} \upharpoonright \mathsf{A}, \mathsf{B}_q$ is an O-state and $X\backslash\{b\} \upharpoonright \mathsf{B}_q, \mathsf{B}_p$ is a P-state, which means $X\backslash\{b\}$ is a POPO-state. So X is a PPOO-state.

ii. If b is negative in B and occurs in B_p, then it is negative in $\mathsf{B}_q \vdash \mathsf{B}_p$ and positive in $\mathsf{B}_p \vdash \mathsf{C}$. So, by induction hypothesis, $X\backslash\{b\} \upharpoonright \mathsf{B}_q, \mathsf{B}_p$ is an O-state and $X\backslash\{b\} \upharpoonright \mathsf{B}_p, \mathsf{C}$ is a P-state, which means $X\backslash\{b\}$ is a POOP-state. So X is a POPO-state.

iii. If b is positive in B and occurs in B_q, then it is positive in $\mathsf{A} \vdash \mathsf{B}$ and negative in $\mathsf{B}_q \vdash \mathsf{B}_p$. So, by induction hypothesis, $X\backslash\{b\} \upharpoonright \mathsf{A}, \mathsf{B}_q$ is a P-state and $X\backslash\{b\} \upharpoonright \mathsf{B}_q, \mathsf{B}_p$ is an O-state, which means $X\backslash\{b\}$ is a PPOO-state. So X is a POPO-state.

iv. If b is positive in B and occurs in B_p, then it is positive in $\mathsf{B}_q \vdash \mathsf{B}_p$ and negative in $\mathsf{B}_p \vdash \mathsf{C}$. So, by induction hypothesis, $X\backslash\{b\} \upharpoonright \mathsf{B}_q, \mathsf{B}_p$ is a P-state and $X\backslash\{b\} \upharpoonright \mathsf{B}_p, \mathsf{C}$ is an O-state, which means $X\backslash\{b\}$ is a POPO-state. So X is a POOP-state.

In each case, we can check that the inductively constructed linearisation t follows Figure 6.11 and that $t \upharpoonright \mathsf{A}, \mathsf{C} = \mathsf{t}$. □

Now we prove the reverse inclusion of Proposition 6.69.

Lemma 6.77 – $\mathsf{Plays}^{\Rightarrow}(-)$ and composition, part 2

Consider A, B and C arenas with $\sigma : \mathsf{A} \vdash \mathsf{B}$ and $\tau : \mathsf{B} \vdash \mathsf{C}$. Then:

$$\mathsf{Plays}^{\Rightarrow}(\tau \odot \sigma) \subseteq \mathsf{Plays}^{\Rightarrow}(\tau) \odot^{\mathsf{HO}} \mathsf{Plays}^{\Rightarrow}(\sigma)\,.$$

Proof. Consider $s \in \mathsf{Plays}^{\Rightarrow}(\tau \odot \sigma)$. By definition there exist

$$\mathsf{p} \in \sigma, \qquad p = \underline{\mathsf{p}}, \qquad \mathsf{r} \in \tau, \qquad r = \underline{\mathsf{r}}, \qquad \varphi : x_{p \upharpoonright \mathsf{B}} \cong_{\mathsf{B}} x_{r \upharpoonright \mathsf{B}},$$

with $q = \Lambda^{\Rightarrow}(r \odot_{\varphi} p)$ and $\mathsf{t} \in \mathsf{Alt}(q)$, such that $s = \partial_q(\mathsf{t})$. We want to prove that

$$s \in \mathsf{Plays}^{\Rightarrow}(\tau) \odot^{\mathsf{HO}} \mathsf{Plays}^{\Rightarrow}(\sigma).$$

More precisely we need plays $s^{\sigma} \in \mathsf{Plays}(\Lambda^{\Rightarrow}(\mathsf{p}))$ and $s^{\tau} \in \mathsf{Plays}(\Lambda^{\Rightarrow}(\mathsf{r}))$, along with an interaction $u \in \mathrm{I}(\mathsf{A}, \mathsf{B}, \mathsf{C})$ such that

$$u \upharpoonright \mathsf{A}, \mathsf{B} = s^{\sigma}, \qquad u \upharpoonright \mathsf{B}, \mathsf{C} = s^{\tau}, \qquad u \upharpoonright \mathsf{A}, \mathsf{C} = s.$$

This interaction will be constructed thanks to Lemma 6.76. Consider

$$X = |\Lambda^{\Rightarrow}(r \circledast_{\varphi} p)| = |r \circledast_{\varphi} p|$$

Then X is an OOOO-state of $r \circledast_{\varphi} p$, and t is an alternating linearisation of $X \upharpoonright \mathsf{A}, \mathsf{C}$ (since $\Lambda^{\Rightarrow}(-)$ does not change polarities). By Lemma 6.76, there exists a linearisation t, following the diagram of Figure 6.11, such that $t \upharpoonright \mathsf{A}, \mathsf{C} = \mathsf{t}$. Since t follows the state diagram (and $\emptyset$, X are OOOO-states), events in B must occur in pairs: any event $(1, b)$ occuring in B_p^+ is followed by $(2, \varphi(b))$ occuring in B_r, and likewise any event $(2, b)$ occuring in B_r^- is followed by $(1, \varphi^{-1}(b))$

occuring in B_p. Hence, we construct t' where we consider pairs of events occuring in B, and the corresponding interaction u, where pointers follow $\multimap_{\Lambda^{\Rightarrow}(p)}$ and $\multimap_{\Lambda^{\Rightarrow}r}$. By construction, we have:

$$u \upharpoonright \mathsf{A}, \mathsf{B} \in \mathsf{Plays}(\Lambda^{\Rightarrow}(\mathsf{p})) \qquad u \upharpoonright \mathsf{B}, \mathsf{C} \in \mathsf{Plays}(\Lambda^{\Rightarrow}(\mathsf{r})) \qquad u \upharpoonright \mathsf{A}, \mathsf{C} = s$$

as required. □

Hence $\mathsf{Plays}^{\Rightarrow}(-)$ is compatible with the compositions. Since the composition in HO games preserves innocence, we can deduce that the composition in PCG also preserves innocence (*i.e.* being a FII).

6.5.5 Functor between PCG and HO

We are now able to properly state the correspondance between the two categories FII the category of FIIs in PCG and $\mathsf{HO}_f^{\mathsf{Inn}}$ the category of finite innocent strategies in HO.

Proposition 6.78 – Functor between PCG and HO

There is a functor:

$$\mathsf{Plays}^{\Rightarrow}(-) \colon \mathsf{FII} \longrightarrow \mathsf{HO}_f^{\mathsf{Inn}}\,.$$

Proof. For any $\sigma \in \mathsf{FII}(\mathsf{A} \vdash \mathsf{B})$, we have $\mathsf{Plays}^{\Rightarrow}(\sigma) \in \mathsf{HO}_f^{\mathsf{Inn}}(\mathsf{A} \Rightarrow \mathsf{B})$ by Lemma 6.67 and Proposition 3.47. We conclude with Propositions 6.68 and 6.69 for identity and composition respectively. □

Remark: Proposition 3.47 actually gives us the isomorphism for plays quotiented by homotopy, but since $\mathsf{Plays}^{\Rightarrow}(-)$ returns plays we drop the quotient here.

The construction for the general functor between PCG and possibly *infinite* innocent strategies is not detailed here. The idea is to define infinite augmentations and isogmentations (again, one can think of them as infinite "trees of P-views") and then work mostly with finite prefixes of infinite augmentations; hence the actual proofs are not so different.

Cartesian Structure. Recall that in PCG, for any arenas A and B, the projections π_{A} and π_{B} are strategies with a copycat-like behavior on $\mathsf{A} \otimes \mathsf{B} \vdash \mathsf{A}$ and $\mathsf{A} \otimes \mathsf{B} \vdash \mathsf{B}$. They correspond exactly to $\pi_{\mathsf{A}}^{\mathsf{HO}}$ and $\pi_{\mathsf{B}}^{\mathsf{HO}}$.

Lemma 6.79 – Preservation of projections

For any arenas A, B, we have:

$$\mathsf{Plays}^{\Rightarrow}(\pi_{\mathsf{A}}) = \pi_{\mathsf{A}}^{\mathsf{HO}} \qquad \text{and} \qquad \mathsf{Plays}^{\Rightarrow}(\pi_{\mathsf{B}}) = \pi_{\mathsf{B}}^{\mathsf{HO}}\,.$$

The proof is very similar to the one for Proposition 6.68.

Closed Structure. Recall the isomorphism in PCG:

$$\Lambda_{\mathsf{A},\mathsf{B},\mathsf{C}} \colon \mathsf{PCG}(\mathsf{A} \otimes \mathsf{B}, \mathsf{C}) \cong \mathsf{PCG}(\mathsf{A}, \mathsf{B} \Rightarrow \mathsf{C})\,.$$

Likewise, in HO the currying isomorphism is:

$$\Lambda^{\mathsf{HO}}_{\mathsf{A,B,C}} : \mathsf{HO}(\mathsf{A} \otimes \mathsf{B}, \mathsf{C}) \cong \mathsf{HO}(\mathsf{A}, \mathsf{B} \Rightarrow \mathsf{C}),$$

We show that $\mathsf{Plays}^{\Rightarrow}(-)$ is compatible with the curryfication.

Lemma 6.80 – Preservation of curryfication

For any $\sigma : \mathsf{A} \otimes \mathsf{B} \vdash \mathsf{C}$, we have:

$$\mathsf{Plays}^{\Rightarrow}(\Lambda_{\mathsf{A,B,C}}(\sigma)) = \Lambda^{\mathsf{HO}}_{\mathsf{A,B,C}}(\mathsf{Plays}^{\Rightarrow}(\sigma)).$$

Proof. By computation; both curryfication morphisms behave in the same way. □

Recall that the evaluation in PCG is defined by:

$$\mathrm{ev}_{\mathsf{A,B}} \stackrel{\text{def}}{=} \Lambda^{-1}_{\mathsf{A}\Rightarrow\mathsf{B},\mathsf{A},\mathsf{B}}(\mathsf{id}_{\mathsf{A}\Rightarrow\mathsf{B}}) \in \mathsf{PCG}((\mathsf{A} \Rightarrow \mathsf{B}) \otimes \mathsf{A}, \mathsf{B}).$$

Likewise, the evaluation in HO is:

$$\mathsf{ev}^{\mathsf{HO}}_{\mathsf{A,B}} \stackrel{\text{def}}{=} \left(\Lambda^{\mathsf{HO}}_{\mathsf{A}\Rightarrow\mathsf{B},\mathsf{A},\mathsf{B}}\right)^{-1} (\mathsf{cc}^{\mathsf{HO}}_{\mathsf{A}\Rightarrow\mathsf{B}}) \in \mathsf{HO}((\mathsf{A} \Rightarrow \mathsf{B}) \otimes \mathsf{A}, \mathsf{B}).$$

From the previous lemmas, we directly obtain:

$$\mathsf{Plays}^{\Rightarrow}(\mathrm{ev}_{\mathsf{A,B}}) = \mathsf{ev}^{\mathsf{HO}}_{\mathsf{A,B}}.$$

All in all, we have a strict cartesian closed functor.

Theorem 6.81 – Strict cartesian closed functor

$\mathsf{Plays}^{\Rightarrow}(-)$ is a strict cartesian closed functor between FII and $\mathsf{HO}^{\mathsf{Inn}}_{f}$.

6.6 Conclusion and perspectives

We enriched the game model PCG with *composition*, allowing us to study its categorical structure. For now, we showed that PCG is a SMCC – in the next chapter we define resource categories, which are better suited to express what interest us in the structure of PCG.

We also established a strict cartesian closed functor between PCG and HO, building upon the isomorphisms from Chapter 3. This correspondance only focuses on the qualitative aspect of PCG; the natural follow-up question would be about the significance of the coefficients in HO.

Resource Categories | 7

Resource categories intend to capture the categorical structure of *pointer concurrent games*. The aim is to obtain a categorical framework enabling the characterization of morphisms behaving "linearly", to show that these morphisms in pointer concurrent games are in bijection with normal terms of the *resource λ-calculus*; and also to structure the interpretation of resource terms as strategies, to prove invariance under reduction.

We start by giving the definition in Section 7.1, as well as some useful properties in Section 7.2. We focus on the interpretation and its soundness in Section 7.3 – in Chapter 8 we will prove that PCG is indeed a resource category. Finally we give an example of the construction of a resource category from a differential category (more exactly a monoidal storage category) in Section 7.4.

7.1 Definition

Before the actual definition, we give some of the intuitions behind the main components of a resource category:

- Composition in games generates sums of isogmentations; likewise, substitution in the resource calculus generates sums of terms. Hence, resource categories have an *additive* structure.
- Resource terms are built using multisets of terms; we would like a way to "flatten" multisets of morphisms into one morphism. This operation is constructed via a *bialgebra* structure.
- Resource categories are *not* linear, because strategies, the morphisms in pointer concurrent games, do not have a linear behavior in general. However, we want to characterize the morphisms that do behave linearly – because they correspond to resource terms. This is achieved using the *pointed identities* morphisms.

7.1.1 Additivity

We call *additive* categories that are enriched over commutative monoids[1].

1: We follow the definition of [7, Section 2], which differs from the one given in [32].

[7]: Blute, Cockett, and Seely (2006), 'Differential categories'

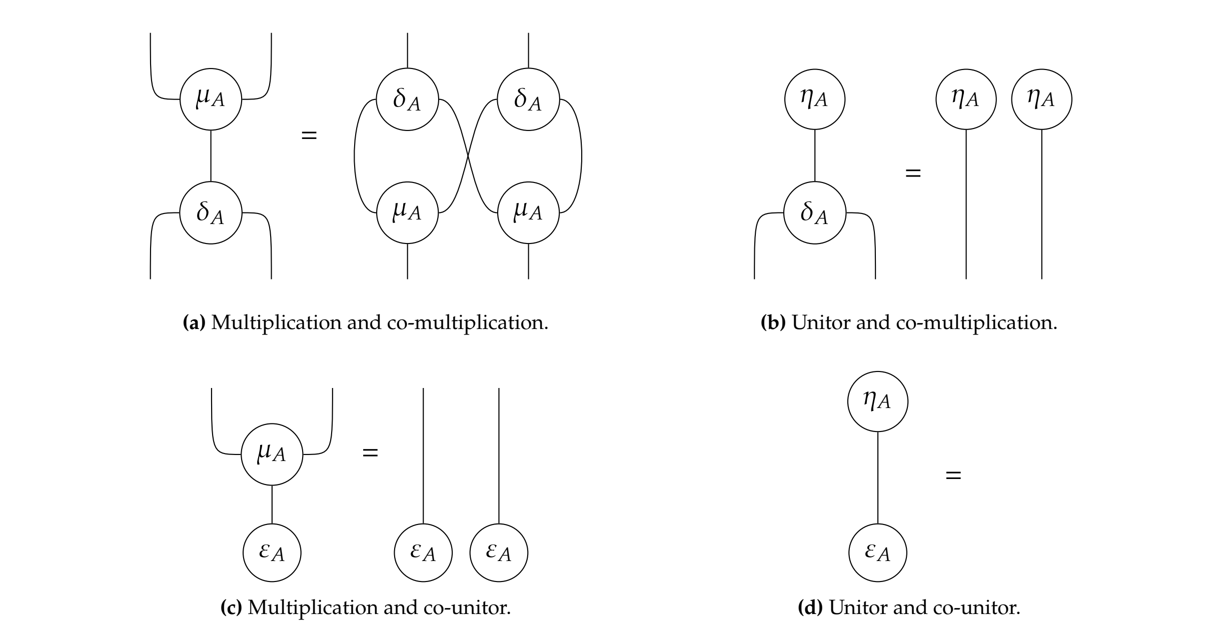
[32]: Mac Lane (1971), *Categories for the Working Mathematician*

Definition 7.1 – Additive SMC (ASMC)

An **additive symmetric monoidal category** (asmc) is a symmetric monoidal category (see Definition 1.2) where each hom-set is a commutative monoid, with an addition + and a zero 0, such that composition and tensor distribute over the additive structure:

$$h \circ (f+g) \circ k = h \circ f \circ k + h \circ g \circ k \qquad h \circ 0 \circ k = 0$$
$$h \otimes (f+g) \otimes k = h \otimes f \otimes k + h \otimes g \otimes k \qquad h \otimes 0 \otimes k = 0$$

for any morphisms k, f, g, h.



(a) Multiplication and co-multiplication. **(b)** Unitor and co-multiplication. **(c)** Multiplication and co-unitor. **(d)** Unitor and co-unitor.

Figure 7.1: Bialgebra laws.

7.1.2 Bialgebras.

Resource categories are equipped with *bialgebras*, which are a monoid and a comonoid with coherence laws between the two structures.

Definition 7.2 – Bialgebra

Consider C an additive symmetric monoidal category.

A **bialgebra** on C is $(A, \delta_A, \varepsilon_A, \mu_A, \eta_A)$ with

- (A, μ_A, η_A) a commutative monoid (see Definition 1.5),
- $(A, \delta_A, \varepsilon_A)$ a commutative comonoid (see Definition 1.7),
- and additional bialgebra laws presented in Figure 7.1.

In resource categories, every object has a bialgebra structure. Intuitively, comonoids (A, δ_A, η_A) are a way to represent *duplications* and duplicable objects: if a request is made on the output of δ_A on either side of the tensor, the request is forwarded to its input. Monoids $(A, \mu_A, \varepsilon_A)$ reflect the sums coming from compositions of strategies: requests made on the output of μ_A are forwarded non-deterministically to its input on either side of the tensor.

7.1.3 Pointed Identity

Finally, we wish to characterize morphisms that "behave linearly" (in pointer concurrent game, they correspond to singleton multisets of tree-like augmentations, using their argument exactly once). To do so, we introduce a morphism called *pointed identity*, which acts as an identity *only for "linear morphisms"*[2].

2: The name *pointed* identity comes from the particular case of pointed identities in the resource category of pointer concurrent games: tree-like augmentations corresponding to linear morphisms in games are called *pointed*, because their forestial structure has a unique root.

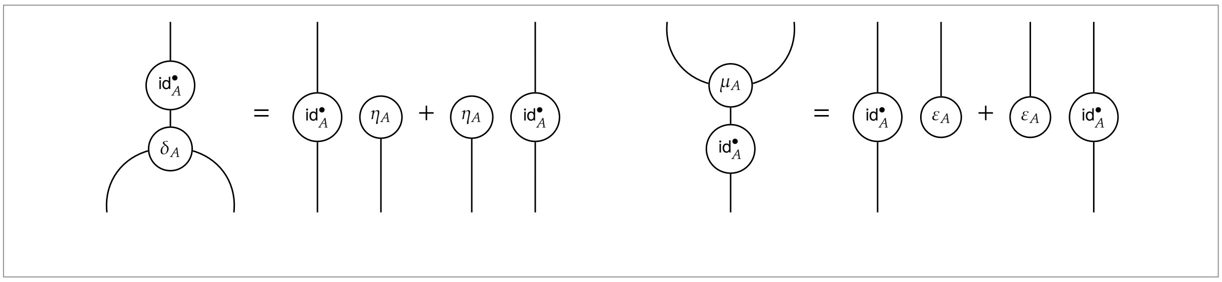


Figure 7.2: Laws for (co)multiplication and pointed identity

Definition 7.3 – Pointed identity

Consider $\mathscr{C}$ an asmc where each object has a bialgebra structure. For any A, a **pointed identity** is $\mathsf{id}^\bullet_A \in \mathscr{C}(A, A)$ satisfying:

idempotent:	$\mathsf{id}^\bullet_A \circ \mathsf{id}^\bullet_A = \mathsf{id}^\bullet_A$
non-erasable:	$\varepsilon_A \circ \mathsf{id}^\bullet_A = 0$
non-erasing:	$\mathsf{id}^\bullet_A \circ \eta_A = 0$

and the equations of Figure 7.2.

The equations of Figure 7.2 express the following properties of $\mathsf{id}^\bullet_A$:

- *non-duplicable:* the post-composition with the co-multiplication δ_A is the sum of "$\mathsf{id}^\bullet_A$ takes a request from the left-hand side of the tensor" and "$\mathsf{id}^\bullet_A$ takes a request from the right-hand side", but no situation in which $\mathsf{id}^\bullet_A$ takes requests from both sides simultaneously;
- *non-duplicative:* the pre-composition with the multiplication μ_A is the sum of "$\mathsf{id}^\bullet_A$ forwards a request to the left-hand side of the tensor" and "$\mathsf{id}^\bullet_A$ forwards a request to the right hand side" but no "$\mathsf{id}^\bullet_A$ forwards the request to both sides".

This "strongly linear" behavior of $\mathsf{id}^\bullet$ will allow us to characterize linear morphisms: those which are invariant by composition with the pointed identity.

Definition 7.4 – (Co-)Pointed Morphisms

Consider A, B in an asmc $\mathscr{C}$ equipped with bialgebras, and the pointed identities $\mathsf{id}^\bullet_A$ and $\mathsf{id}^\bullet_B$.

Then $f \in \mathscr{C}(A, B)$ is **pointed** if $\mathsf{id}^\bullet_B \circ f = f$. We write $f \in \mathscr{C}_\bullet(A, B)$.

Dually, f is **co-pointed** if $f \circ \mathsf{id}^\bullet_A = f$. We write $f \in \mathscr{C}^\bullet(A, B)$.

Intuitively, pointed morphisms are morphisms behaving linearly for the substitution: they can only be used exactly once. Dually, co-pointed morphisms are morphisms behaving linearly with their arguments: they require exactly one resource.

7.1.4 Resource Categories

We can now define resource categories.

Figure 7.3: Compatibility of (co)monoids with the monoidal structure

Definition 7.5 – Resource Category

Consider an asmc $\mathscr{C}$. It is a **resource category** if each object A has a bialgebra structure $(A, \delta_A, \varepsilon_A, \mu_A, \eta_A)$ with a pointed identity $\mathsf{id}^\bullet_A$, and bialgebras are compatible with the monoidal structure of $\mathscr{C}$ in the sense that the morphisms satisfy:

co-unitor with tensor:	$\varepsilon_{A\otimes B} = \lambda_I \circ (\varepsilon_A \otimes \varepsilon_B)$
unitor with tensor:	$\eta_{A\otimes B} = (\eta_A \otimes \eta_B) \circ \lambda_I$
(co-)unitors with unit:	$\varepsilon_I = \eta_I = \mathsf{id}_I$

and the equations of Figure 7.3.

Resource categories offer an interpretation of the resource calculus, in which (singleton multisets of) terms are pointed morphisms. Linearity here is characterized using pointed identities; but linearity can also be linked to differential categories. Pointed identity laws are very similar to the dereliction and codereliction morphisms laws which occur in differential categories, which will guide us in our construction of a resource category in Section 7.4.

7.1.5 Closeness

Since we are interested in interpreting typed λ-terms, we want some kind of currying isomorphism. Hence we consider *closed* resource categories.

Definition 7.6 – Closed resource category

A resource category $\mathscr{C}$ is **closed** if for all $A \in \mathscr{C}$, the endofunctor $- \otimes A$ has a right adjoint $A \Rightarrow -$.

The **currying** is the natural (in A, B, C) isomorphism:

$$\Lambda_{A,B,C} : \mathscr{C}(A \otimes B, C) \;\cong\; \mathscr{C}(A, B \Rightarrow C)\,.$$

For all A, B, the **evalutation** morphism is:

$$\mathrm{ev}_{A,B} \stackrel{\text{def}}{=} \Lambda^{-1}_{A\Rightarrow B,A,B}(\mathsf{id}_{A\Rightarrow B})\,.$$

We ask that the currying isomorphism is compatible with pointed identity in the following sense, for all A, B:

$$\mathsf{id}^\bullet_{A\Rightarrow B} \;=\; \Lambda_{A\Rightarrow B,A,B}\left(\mathsf{id}^\bullet_B \circ \mathrm{ev}_{A,B}\right)\,.$$

7.2 Properties of resource categories

7.2.1 Constructions

We first give a few additionnal constructions in resource categories, which will be useful both for the interpretation of resource calculus and for describing the categorical structure of PCG.

Union. Strategies in pointer concurrent games are sums of augmentations, and augmentations have a forestial structure: they are, in a way, finite multisets of tree-like sub-augmentations. This matches the fact that in resource calculus, terms are applied to multisets of terms instead of terms. Bialgebra morphisms allow us to formalize this intuition and to flatten any multiset of morphisms into a single morphism.

For any morphisms $f, g \colon A \to B$, we define their **union** as:

$$f * g \stackrel{\text{def}}{=} \mu_B \circ (f \otimes g) \circ \delta_A \in \mathscr{C}(A, B)\,,$$

capturing the idea of the union of the multisets f and g.

Moreover, we define the union of the empty multiset as:

$$1_{A,B} \stackrel{\text{def}}{=} \eta_B \circ \varepsilon_A \in \mathscr{C}(A, B)\,.$$

With these definitions, $(\mathscr{C}(A, B), *, 1_{A,B})$ is a commutative monoid (and $\mathscr{C}(A, B)$ is a commutative semiring, where the composition and the tensor only preserve the additive monoid).

Since $*$ is associative, we unambiguously define the n**-ary union**: given a multiset of morphisms $\bar{f} = [f_1, \dots, f_n]$ in $\mathscr{M}_f(\mathscr{C}(A, B))$, we set:

$$\Pi\, \bar{f} \;=\; f_1 * \dots * f_n \;\in \mathscr{C}(A, B)\,.$$

Hence, we send multisets of morphisms to single morphisms *via* Π.

Remark that this construction matches $\Pi_{\mathsf{Isog}}[-]$, the "flattening" of a multiset of isogmentations on the same arena into a single isogmentation.

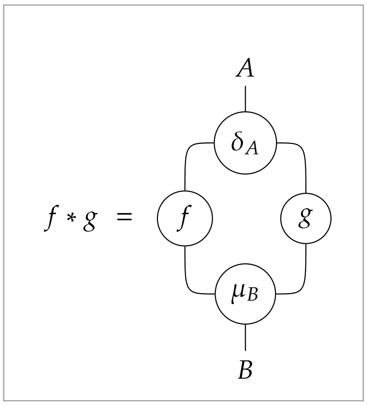


Figure 7.4: Union.

Tupling. Likewise, we would like a construction matching $\langle - \rangle_{\mathsf{Isog}}$, the tupling of a sequence of isogmentations on some arenas $\Gamma \vdash \mathsf{A}_i$'s into an isogmentation on the arena $\Gamma \vdash \vec{\mathsf{A}}^{\otimes}$.

For any objects A, B, C with morphisms $f \colon A \to B$ and $g \colon A \to C$, we define their **tupling**:

$$\langle f, g \rangle \stackrel{\text{def}}{=} (f \otimes g) \circ \delta_A \in \mathscr{C}(A, B \otimes C)\,.$$

This ressembles the product in cartesian categories; in the same way, we define the **tupling projections**:

$$\pi_\ell \stackrel{\text{def}}{=} \rho_A \circ (\mathsf{id}_A \otimes \varepsilon_B) \qquad \in \mathscr{C}(A \otimes B, A)$$
$$\pi_r \stackrel{\text{def}}{=} \lambda_B \circ (\varepsilon_A \otimes \mathsf{id}_B) \qquad \in \mathscr{C}(A \otimes B, B)$$

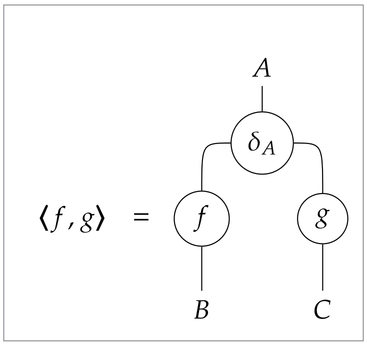


Figure 7.5: Tupling.

In case of ambiguity, we write $(\pi_\ell^{A,B}, \pi_r^{A,B})$. We might also occasionally use the notations (π_1, π_2) or (π_A, π_B).

However, *this is not a cartesian product!* **We do not have** $\pi_\ell \circ \langle f, g \rangle = f$ **in general** – indeed, this only holds if g is *erasable*, *i.e.* if $\varepsilon_C \circ g = \varepsilon_A$. Likewise, $\langle f, g \rangle \circ h = h$ only holds if $h \in \mathscr{C}(D, A)$ is *duplicable*, *i.e.* if $\delta_A \circ h = (h \otimes h) \circ \delta_D$ (see Subsection 7.2.3).

Again, we extend the definition to the n**-ary tupling**; given morphisms $(f_i \colon A \longrightarrow B_i)_{1 \leq i \leq n}$, we set:

$$\langle f_1, \dots, f_n \rangle \stackrel{\text{def}}{=} \langle f_1, \langle f_2, \dots, f_n \rangle \rangle \qquad \in \mathscr{C}(A, B_1 \otimes \dots \otimes B_n)$$

with the projections π_i's constructed in the obvious way.

Packing. In the next section, sequences and bags are interpreted as actual tuples and multisets rather than directly as morphisms in $\mathscr{C}$. To compose bags we "flatten" them via the union; likewise for sequences of morphisms we might use the tupling to see the sequence as a single morphism. Putting these two construction together, we define the **packing** of a sequence of bags of morphisms. Given the sequence $\vec{f} := \langle \bar{f}_1, \dots, \bar{f}_n \rangle$ with multiset $\bar{f}_i \in \mathcal{M}_f(\mathscr{C}(A, B_i))$ for any $1 \leq i \leq n$, we set:

$$\langle\!| \vec{f} |\!\rangle \stackrel{\text{def}}{=} \langle \Pi \bar{f}_1, \dots, \Pi \bar{f}_n \rangle \qquad \in \mathscr{C}(A, B_1 \otimes \dots \otimes B_n)\,.$$

7.2.2 Bags of pointed morphisms

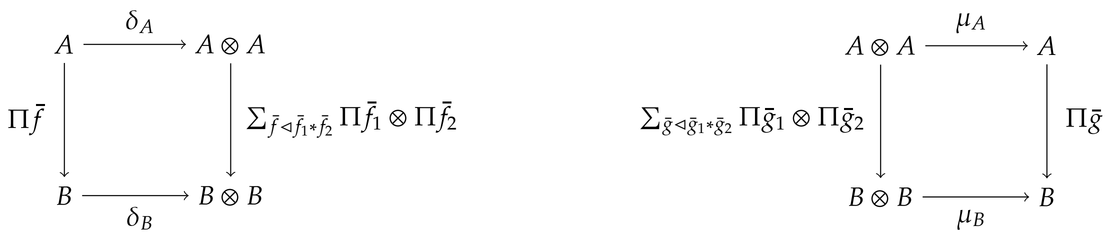
One of the key properties of resource calculus is the fact that the substitution creates *sums* of resource terms, following the multiple ways of splitting a bag of terms. In resource categories, terms are interpreted as pointed morphisms, and bags of terms as bags of pointed morphisms, flattened via the union operation when needed. Hence, we need to study the categorical equivalent of splitting a bag of terms: how does the union of a bag of pointed morphisms behave when we try to "split" it into two (or several) morphisms?

The key property derived from the definition of resource categories expresses how the product of a bag of pointed morphisms interacts with the comonoid structure – and dually for product of a bag of co-pointed morphisms and monoids.

Lemma 7.7 – Key Lemma

Consider $\mathscr{C}$ a resource category, then:

1. For any bag of pointed morphisms $\bar{f} \in \mathcal{M}_f(\mathscr{C}_\bullet(A, B))$,
   a) the diagram of Figure 7.6a commutes;
   b) we have $\varepsilon_B \circ \Pi \bar{f} = 1$ if $\bar{f}$ is empty, 0 otherwise.
2. For any bag of co-pointed morphisms $\bar{g} \in \mathcal{M}_f(\mathscr{C}^\bullet(A, B))$,
   a) the diagram of Figure 7.6b commutes;
   b) we have $\Pi \bar{g} \circ \eta_A = 1$ if $\bar{g}$ is empty, 0 otherwise.



(a) Pointed morphisms and comultiplication. **(b)** Copointed morphisms and multiplication.

Figure 7.6: Interaction of bags with the (co-)monoid structure.

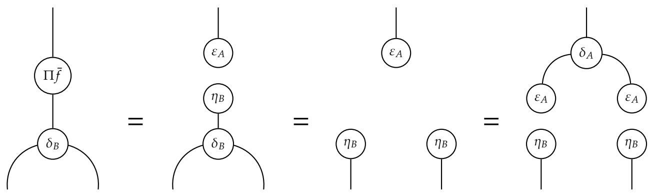
Proof. **1.** Consider a bag of pointed morphisms $\bar{f} \in \mathcal{M}_f(\mathscr{C}_\bullet(A, B))$.

a) We reason by induction on the size of $\bar{f}$.

If $\bar{f}$ is empty, then by definition $\Pi\bar{f} = \eta_B \circ \varepsilon_A$. Moreover, the only 2-partitioning of [] is $[] \triangleleft [] * []$. Hence, we have:

$$\sum_{\bar{f} \triangleleft \bar{f}_1 * \bar{f}_2} \Pi\bar{f}_1 \otimes \Pi\bar{f}_2 = \Pi[] \otimes \Pi[] = (\eta_B \circ \varepsilon_A) \otimes (\eta_B \circ \varepsilon_A) . \quad (7.1)$$

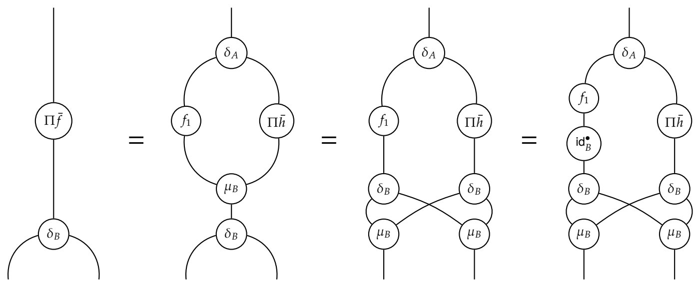
Using bialgebra laws (Figure 7.1), we compute:



And by (7.1), we obtain exactly the diagram of Figure 7.6a.

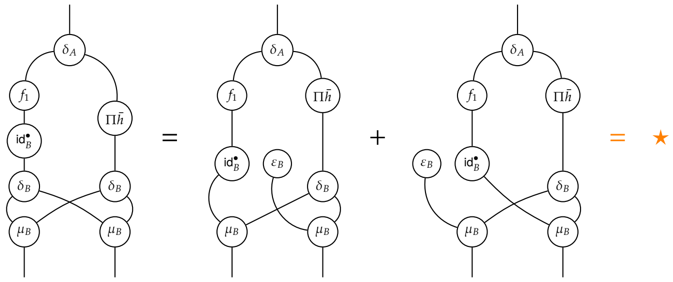
If $\bar{f} = [f_1, \ldots, f_n]$ with $n \geq 1$, let us write $\bar{h} = [f_2, \ldots, f_n]$.

Then $\Pi\bar{f} = f_1 * \Pi\bar{h}$, and we compute:

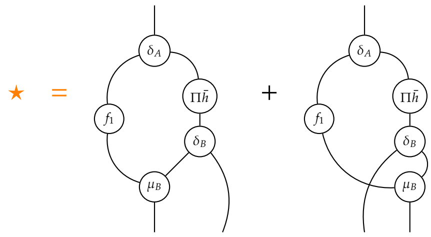


using the exchange rule of bialgebra (Figure 7.1) and the fact that f_1 is pointed.

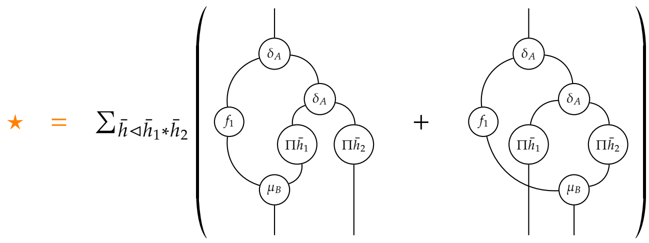
From the laws of pointed identities (Figure 7.2), we get:



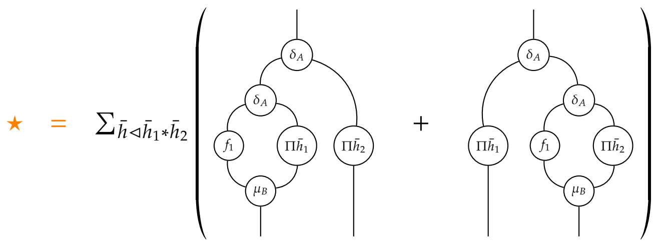
Using the fact that f_1 is pointed and a bialgebra law (Figure 7.1), we simplify this sum to:



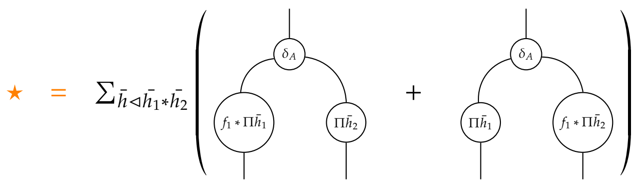
Now, by induction hypothesis, we can replace $\delta_B \circ \Pi\bar{h}$ in the diagram above, and by additivity we get:



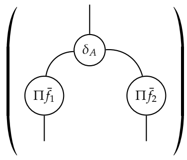
By associativity and symmetry of δ, this rewrites to:



By definition of the union, this is exactly:



Therefore, following the definition of a 2-partitioning, we have:

$\star \;=\; \Sigma_{\bar{f} \triangleleft \bar{f}_1 * \bar{f}_2}$ 

which is exactly the diagram from Figure 7.6a.

b) If $\bar{f}$ is empty, we have:

$$\begin{aligned} \varepsilon_B \circ \Pi\bar{f} &= \varepsilon_B \circ \eta_B \circ \varepsilon_A \\ &= \mathsf{id}_I \circ \varepsilon_A \\ &= 1_{A,I} \end{aligned}$$

by definition of the union on the empty multiset; a bialgebra law (Figure 7.1); and a coherence law of resource categories.

Otherwise, writing $\bar{f}$ as $[f_1, \ldots, f_n]$, we have $\Pi\bar{f} = f_1 * \Pi[f_2, \ldots, f_n]$. Writing $h = \Pi[f_2, \ldots, f_n]$, we obtain:

$$\begin{aligned} \varepsilon_B \circ \Pi\bar{f} &= \varepsilon_B \circ \big(f_1 * h\big) \\ &= \varepsilon_B \circ \mu_B \circ \big(f_1 \otimes h\big) \circ \delta_A \\ &= (\varepsilon_B \otimes \varepsilon_B) \circ \big(f_1 \otimes h\big) \circ \delta_A \\ &= \big(\varepsilon_B \circ f_1\big) \otimes (\varepsilon_B \circ h) \circ \delta_A \\ &= \big(\varepsilon_B \circ \mathsf{id}^\bullet_B \circ f_1\big) \otimes (\varepsilon_B \circ h) \circ \delta_A \\ &= 0. \end{aligned}$$

by definition of the union; a bialgebra law; bifunctoriality of $\otimes$; the fact that f_1 is pointed; and the fact that the pointed identity is non-erasable.

2. Completely symmetric to **1**. □

We also describe the interactions of (unions of) bags of pointed morphisms with pointed identities.

Lemma 7.8 – Interaction of bags with $\mathsf{id}^\bullet$

Consider $\bar{f} \in \mathscr{M}_f(\mathscr{C}_\bullet(A, B))$. Then:

$$\mathsf{id}^\bullet_B \circ \Pi\bar{f} \;=\; \begin{cases} g & \text{if } \bar{f} = [g], \\ 0 & \text{otherwise.} \end{cases}$$

Proof. If $\bar{f} = [g]$, we clearly have

$$\mathsf{id}^\bullet_B \circ \Pi[g] \;=\; \mathsf{id}^\bullet_B \circ g \;=\; g$$

since g is pointed.

Otherwise, if $\bar{f} = [\,]$, then $\Pi\bar{f} = 1_{A,B} = \eta_B \circ \varepsilon_A$, and we compute:

$$\mathsf{id}^\bullet_B \circ \Pi\bar{f} \;=\; \mathsf{id}^\bullet_B \circ \eta_B \circ \varepsilon_A \;=\; 0 \circ \varepsilon_A \;=\; 0$$

by non-erasing property of pointed identity.

Finally, if $\bar{f}$ has at least two elements, we can write $\Pi\bar{f} = g * \Pi\bar{h}$ with g a pointed morphism and $\bar{h}$ a non-empty bag of pointed morphisms. We compute:

$$\begin{aligned}
&\mathsf{id}^\bullet_B \circ (g * \Pi\bar{h}) \\
&= \mathsf{id}^\bullet_B \circ (\mu_B \circ (g \otimes \Pi\bar{h}) \circ \delta_A) \\
&= \{(\mathsf{id}^\bullet_B \otimes \varepsilon_B) \circ (g \otimes \Pi\bar{h}) \circ \delta_A\} + \{(\varepsilon_B \otimes \mathsf{id}^\bullet_B) \circ (g \otimes \Pi\bar{h}) \circ \delta_A\} \\
&= \{(\mathsf{id}^\bullet_B \circ g) \otimes (\varepsilon_B \circ \Pi\bar{h}) \circ \delta_A\} + \{(\varepsilon_B \circ g) \otimes (\mathsf{id}^\bullet_B \circ \Pi\bar{h}) \circ \delta_A\} \\
&= \{(\mathsf{id}^\bullet_B \circ g) \otimes 0 \circ \delta_A\} + \{0 \otimes (\mathsf{id}^\bullet_B \circ \Pi\bar{h}) \circ \delta_A\} \\
&\quad = \; 0
\end{aligned}$$

by definition of the union; non-duplicative property of $\mathsf{id}^\bullet$; bifunctoriality of $\otimes$; Lemma 7.7 for the first term and pointedness of g with non-erasive property of $\mathsf{id}^\bullet$ for the second one; and asmc laws. □

7.2.3 Comonoid morphisms

As observed in the previous pages, resource categories are *not* cartesian: although tupling shares some similarities with a cartesian product, it does not behave like one in general. However, some particular morphisms do behave as is usual in a cartesian category: *comonoid morphisms*.

Definition 7.9 – Comonoid morphism

A morphism $f \in \mathscr{C}(A, B)$ is a **comonoid morphism** if:

$$\delta_B \circ f = (f \otimes f) \circ \delta_A \qquad \text{and} \qquad \varepsilon_B \circ f = \varepsilon_A\,.$$

Of course, identities are comonoid morphisms. It follows from a simple diagram chasing that the projections also are, as well as 1.

Moreover, comonoid morphisms are closed under composition.

Remark: Morphisms obtained by the interpretation of resource terms are never comonoid morphisms, but structural morphisms used in the interpretation always are.

Lemma 7.10 – Comonoid morphism and tupling

Consider $f \in \mathscr{C}(A, B)$, $g \in \mathscr{C}(A, C)$ and $h \in \mathscr{C}(D, A)$.

1. If f is a comonoid morphism, then $\pi_r \circ \langle f, g\rangle = g$;
2. If g is a comonoid morphism, then $\pi_\ell \circ \langle f, g\rangle = f$;
3. If h is a comonoid morphism, then $\langle f, g\rangle \circ h = \langle f \circ h, g \circ h\rangle$.

Proof. Straightforward from the definitions. □

The analogous properties for n-ary tupling follow by induction – we shall also refer to Lemma 7.10 when using these generalizations.

A similar lemma holds for unions.

Lemma 7.11 – Comonoid morphism and union

Consider $\bar{f} \in \mathscr{M}_f(\mathscr{C}(A, B))$ and $h \in \mathscr{C}(C, A)$.

If h is a comonoid morphism, then $\left(\Pi\bar{f}\right) \circ h = \Pi\left(\bar{f} \circ h\right)$.

Finally, we state several distribution properties for the composition with the tupling of a comonoid morphism and a union of pointed morphisms; which again are direct consequences of the definitions.

Lemma 7.12 – Left-projection and $\langle h, \Pi\bar{f}\rangle$

Consider $\bar{f} \in \mathscr{M}_f(\mathscr{C}_\bullet(A, B))$ and $h \in \mathscr{C}(A, B)$ a comonoid morphism. Then:

$$\pi_\ell \circ \langle h, \Pi\bar{f}\rangle = \begin{cases} h & \text{if } \bar{f} \text{ is empty,} \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 7.13 – Tupling and $\langle h, \Pi\bar{f}\rangle$

Consider $\bar{f} \in \mathscr{M}_f(\mathscr{C}_\bullet(A, B))$ and $h \in \mathscr{C}(A, B)$ a comonoid morphism. If $g_1, \ldots, g_n \in \mathscr{C}(C, A)$, then:

$$\langle g_i \mid 1 \le i \le n\rangle \circ \langle h, \Pi\bar{f}_i\rangle = \sum_{\bar{f} \triangleleft \bar{f}_1 * \ldots * \bar{f}_n} \langle g_i \circ \langle h, \Pi\bar{f}_i\rangle \mid 1 \le i \le n\rangle .$$

Lemma 7.14 – Union and $\langle h, \Pi\bar{f}\rangle$

Consider $\bar{f} \in \mathscr{M}_f(\mathscr{C}_\bullet(A, B))$ and $h \in \mathscr{C}(A, B)$ a comonoid morphism. If $\bar{g} := [g_1, \ldots, g_n] \in \mathscr{M}_f(\mathscr{C}(C, A))$, then:

$$\Pi\bar{g} \circ \langle h, \Pi\bar{f}_i\rangle = \sum_{\bar{f} \triangleleft \bar{f}_1 * \ldots * \bar{f}_n} \prod_{1 \le i \le n} \left(g_i \circ \langle h, \Pi\bar{f}_i\rangle\right) .$$

7.3 Interpretation and Soundness

7.3.1 Interpretation

From now on, we fix a closed resource category $\mathscr{C}$ with a chosen object o.

Types and contexts. We first set:

$$
\begin{aligned}
[\![\alpha]\!] &\stackrel{\text{def}}{=} o \\
[\![\langle A_1, \ldots, A_n \rangle]\!] &\stackrel{\text{def}}{=} [\![A_1]\!] \otimes \cdots \otimes [\![A_n]\!] \\
[\![A \to B]\!] &\stackrel{\text{def}}{=} [\![A]\!] \Rightarrow [\![B]\!]
\end{aligned}
$$

Remark: Note that this interpretation is very similar to the one used for PCG, which will make sens in the following chapter – where we study PCG as a resource category, whose objects are arenas – since we choose the singleton arena o as the object o and all the other constructions are the same.

For contexts, we set $[\![\Gamma]\!] \stackrel{\text{def}}{=} \bigotimes_{(x:A)\in\Gamma}[\![A]\!]$.

Note that for any type $A := \vec{B} \to \alpha$, currying and associativity morphisms induce an isomorphism:

$$\zeta_A : [\![A]\!] \longrightarrow [\![\vec{B}]\!] \Rightarrow o\,.$$

If $(x : A) \in \Gamma$, we then write

$$\mathrm{var}_x^\Gamma : [\![\Gamma]\!] \longrightarrow [\![\vec{B}]\!] \Rightarrow o$$

for the projection morphism $[\![\Gamma]\!] \to [\![A]\!]$ followed by ζ_A.

For Γ and Δ disjoint we also use the following isomorphism:

$$⩕_{\Gamma,\Delta} : [\![\Gamma]\!] \otimes [\![\Delta]\!] \longrightarrow [\![\Gamma, \Delta]\!]\,,$$

defined from the symmetric monoidal structure in the obvious way. Remark that $⩕$ is a comonoid morphism.

Terms. The interpretation of terms (or, rather, of typing derivations) follows the three kinds of judgements from Chapter 5.

Consider $\Gamma, A \in \mathscr{C}$ and $\vec{A} := \langle A_1, \ldots, A_n \rangle$. We define:

- $\mathsf{Tm}_{\mathscr{C}}(\Gamma; A) \stackrel{\text{def}}{=} \mathscr{C}_\bullet(\Gamma, A)$,
- $\mathsf{Bg}_{\mathscr{C}}(\Gamma; A) \stackrel{\text{def}}{=} \mathscr{M}_f(\mathsf{Tm}_{\mathscr{C}}(\Gamma; A))$,
- $\mathsf{Sq}_{\mathscr{C}}(\Gamma; \vec{A}) \stackrel{\text{def}}{=} \Pi_{1\le i\le n}\mathsf{Bg}_{\mathscr{C}}(\Gamma; A_i)$.

Remark that sequences and bags are interpreted as *actual* sequences and bags at the "meta-level", rather than via the "internal" bags (*i.e.* products of pointed maps) or products (*i.e.* via the monoidal structure) in $\mathscr{C}$.

This apparent duplication of structure will be resolved when interpreting applications. For that purpose, in addition to the product $\Pi\bar{f} \in \mathscr{C}(\Gamma, A)$ of a bag of morphisms $\bar{f} \in \mathsf{Bg}_{\mathscr{C}}(\Gamma; A)$, we also define the **packing** of a sequence of morphisms $\vec{f} := \langle \bar{f}_1, \ldots, \bar{f}_n \rangle \in \mathsf{Sq}_{\mathscr{C}}(\Gamma; \vec{A})$ as:

$$\langle\!| \vec{f} \,|\!\rangle \stackrel{\text{def}}{=} \langle \Pi\bar{f}_1, \ldots, \Pi\bar{f}_n \rangle \quad \in \mathscr{C}(\Gamma, \vec{A}^{\otimes})\,.$$

$$
\begin{aligned}
\llbracket \Gamma \vdash_{\mathsf{Tm}} \lambda x.s : A \longrightarrow B \rrbracket &\overset{\text{def}}{=} \Lambda_{\llbracket\Gamma\rrbracket,\llbracket A\rrbracket,\llbracket B\rrbracket}(\llbracket \Gamma, x : A \vdash_{\mathsf{Tm}} s : B \rrbracket \circ \wedge\!\!\!\wedge_{\llbracket\Gamma\rrbracket,\llbracket x:A\rrbracket}) \\
\llbracket \Gamma \vdash_{\mathsf{Tm}} x\, \vec{t} : \alpha \rrbracket &\overset{\text{def}}{=} \mathrm{ev}_{\llbracket\vec{A}\rrbracket,\llbracket\alpha\rrbracket} \circ \langle \mathsf{id}^{\bullet}_{\llbracket\vec{A}\rrbracket\Rightarrow o} \circ \mathrm{var}^{\Gamma}_{x}, \langle\!| \llbracket \Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{A} \rrbracket |\!\rangle \rangle \\
\llbracket \Gamma \vdash_{\mathsf{Tm}} s\, \bar{t} : B \rrbracket &\overset{\text{def}}{=} \mathrm{ev}_{\llbracket A\rrbracket,\llbracket B\rrbracket} \circ \langle \llbracket \Gamma \vdash_{\mathsf{Tm}} s : A \longrightarrow B \rrbracket, \Pi\llbracket \Gamma \vdash_{\mathsf{Bg}} \bar{t} : A \rrbracket \rangle \\
\llbracket \Gamma \vdash_{\mathsf{Bg}} [s_1, \dots, s_n] : A \rrbracket &\overset{\text{def}}{=} [\ \llbracket \Gamma \vdash_{\mathsf{Tm}} s_i : A \rrbracket \mid 1 \le i \le n\ ] \\
\llbracket \Gamma \vdash_{\mathsf{Sq}} \langle \bar{s}_1, \dots, \bar{s}_n \rangle : \vec{A} \rrbracket &\overset{\text{def}}{=} \langle\ \llbracket \Gamma \vdash_{\mathsf{Bg}} \bar{s}_i : A_i \rrbracket \mid 1 \le i \le n\ \rangle
\end{aligned}
$$

Figure 7.7: Interpretation of the resource calculus

We now define the three interpretation functions:

- $\llbracket - \rrbracket : \mathsf{Tm}(\Gamma, A) \longrightarrow \mathsf{Tm}_{\mathscr{C}}(\llbracket\Gamma\rrbracket; \llbracket A\rrbracket)$,
- $\llbracket - \rrbracket : \mathsf{Bg}(\Gamma, A) \longrightarrow \mathsf{Bg}_{\mathscr{C}}(\llbracket\Gamma\rrbracket; \llbracket A\rrbracket)$,
- $\llbracket - \rrbracket : \mathsf{Sq}(\Gamma, \vec{A}) \longrightarrow \mathsf{Sq}_{\mathscr{C}}(\llbracket\Gamma\rrbracket; \llbracket \vec{A}\rrbracket)$,

all written $\llbracket - \rrbracket$, by mutual induction as in Figure 7.7.

Remark that by definition, the partitions of $\llbracket \bar{s} \rrbracket$ coincide with (the interpretations of the elements of) the partitions of $\bar{s}$.

The interpretation is extended to sums of terms:

- $\llbracket - \rrbracket : \Sigma\mathsf{Tm}(\Gamma; A) \longrightarrow \Sigma\mathsf{Tm}_{\mathscr{C}}(\llbracket\Gamma\rrbracket; \llbracket A\rrbracket)$

relying on the additive structure of $\mathscr{C}$. We give no interpretation to sums of bags or sequences.

Notation: For the sake of brevity, we might omit brackets when using the interpretation of types, *e.g.* we write id_A for $\mathsf{id}_{\llbracket A \rrbracket}$. Likewise, we might write for example $\llbracket s \rrbracket$ for $\llbracket \Gamma \vdash_{\mathsf{Tm}} s : A \rrbracket$.

7.3.2 Technical lemmas

Finally, we state some technical results needed for the substitution lemma.

Lemma 7.15 – Weakening

Consider $\Gamma \vdash_{\mathsf{Bg}} \bar{s} : A$. Then, we have:

$$\Pi\llbracket \Gamma, x : B \vdash_{\mathsf{Bg}} \bar{s} : A \rrbracket \circ \wedge\!\!\!\wedge_{\Gamma,(x:B)} = \Pi\llbracket \Gamma \vdash_{\mathsf{Bg}} \bar{s} : A \rrbracket \circ \pi_{\ell}$$

Proof. Structural induction of the generalised statement for terms, bags and sequences. □

Lemma 7.16 – Variable substitution

Consider a context Γ with $(y : B) \in \Gamma$. Let $\Delta = \Gamma, x : A$. Then:

$$
\begin{aligned}
&(1) \quad \mathrm{var}^{\Delta}_{x} \circ \wedge\!\!\!\wedge_{\Gamma,(x:A)} = \zeta_A \circ \pi_{r}\,, \\
&(2) \quad \mathrm{var}^{\Delta}_{y} \circ \wedge\!\!\!\wedge_{\Gamma,(x:A)} = \mathrm{var}^{\Gamma}_{y} \circ \pi_{\ell}\,.
\end{aligned}
$$

Proof. Direct from the definitions. □

Lemma 7.17 – Types isomorphism

Consider a type $\mathsf{A} := \vec{B} \to \alpha$. Then,

$$\zeta_{\mathsf{A}} \circ \mathsf{id}^{\bullet}_{[\![\mathsf{A}]\!]} \;=\; \mathsf{id}^{\bullet}_{[\![\vec{B}]\!]\Rightarrow o} \circ \zeta_{\mathsf{A}} \,.$$

Proof. From the properties of Λ and ev, and the compatibility of $\mathsf{id}^{\bullet}$ with Λ (see Definition 7.6). □

Lemma 7.18 – Interpretation of $s\vec{t}$

Consider $\Gamma \vdash_{\mathsf{Tm}} s\vec{t} : \alpha$, with $\Gamma \vdash_{\mathsf{Tm}} s : A$, and $A := \vec{B} \Rightarrow \alpha$. We have the following equality:

$$[\![\Gamma \vdash_{\mathsf{Tm}} s\vec{t} : \alpha]\!] \;=\; \mathrm{ev}_{\vec{B},\alpha} \circ \langle \zeta_A \circ [\![s]\!], \langle\!| [\![\vec{t}]\!] |\!\rangle \rangle$$

Proof. A tedious computation using properties of the structural morphisms. □

7.3.3 Substitution lemma

We show that the interpretation of a substitution *in the resource calculus* can be expressed as a substitution *in the semantics*.

Semantic substitution. The bulk of the proof consists in proving a suitable substitution lemma, for which we must first give a semantic account of substitution. We define three semantic substitution functions:

$$\begin{array}{rcll}
-\langle\!\langle -/x \rangle\!\rangle & : & \mathsf{Tm}_{\mathscr{C}}([\![\Gamma, x : A]\!]; [\![B]\!]) \times \mathsf{Bg}_{\mathscr{C}}([\![\Gamma]\!]; [\![A]\!]) & \to \mathsf{Tm}_{\mathscr{C}}([\![\Gamma]\!]; [\![B]\!]) \\
-\langle\!\langle -/x \rangle\!\rangle & : & \mathsf{Bg}_{\mathscr{C}}([\![\Gamma, x : A]\!]; [\![B]\!]) \times \mathsf{Bg}_{\mathscr{C}}([\![\Gamma]\!]; [\![A]\!]) & \to \mathscr{C}([\![\Gamma]\!], [\![B]\!]) \\
-\langle\!\langle -/x \rangle\!\rangle & : & \mathsf{Sq}_{\mathscr{C}}([\![\Gamma, x : A]\!]; [\![\vec{B}]\!]) \times \mathsf{Bg}_{\mathscr{C}}([\![\Gamma]\!]; [\![A]\!]) & \to \mathscr{C}([\![\Gamma]\!], [\![\vec{B}]\!])
\end{array}$$

using our cartesian-like notations:

$$\begin{array}{rcl}
f\langle\!\langle \bar{g}/x \rangle\!\rangle & \stackrel{\text{def}}{=} & f \circ \mathbb{M}_{[\![\Gamma]\!],[\![x:A]\!]} \circ \langle \mathsf{id}_{[\![\Gamma]\!]}, \Pi\bar{g} \rangle \\
\bar{f}\langle\!\langle \bar{g}/x \rangle\!\rangle & \stackrel{\text{def}}{=} & \Pi\bar{f} \circ \mathbb{M}_{[\![\Gamma]\!],[\![x:A]\!]} \circ \langle \mathsf{id}_{[\![\Gamma]\!]}, \Pi\bar{g} \rangle \\
\vec{f}\langle\!\langle \bar{g}/x \rangle\!\rangle & \stackrel{\text{def}}{=} & \langle\!| \vec{f} |\!\rangle \circ \mathbb{M}_{[\![\Gamma]\!],[\![x:A]\!]} \circ \langle \mathsf{id}_{[\![\Gamma]\!]}, \Pi\bar{g} \rangle \,.
\end{array}$$

Substitution lemma. We may now state the main lemma:

Lemma 7.19 – Substitution

Consider $\bar{t} \in \mathsf{Bg}(\Gamma; A)$, $\Delta = \Gamma, x : A$ and $s \in \mathsf{Tm}(\Delta; B)$. Then,

$$[\![s\langle \bar{t}/x \rangle]\!] = [\![s]\!]\langle\!\langle [\![\bar{t}]\!]/x \rangle\!\rangle \,.$$

Proof. We show the result by induction on typing derivation, proving the stronger statement that for all $\bar{t} \in \mathsf{Bg}(\Gamma; A)$ and $\Delta = \Gamma, x : A$, we have:

(1) if $s \in \mathsf{Tm}(\Delta; B)$, then $[\![s\langle \bar{t}/x\rangle]\!] = [\![s]\!]\langle\!\langle [\![\bar{t}]\!]/x\rangle\!\rangle$;
(2) if $\bar{s} \in \mathsf{Bg}(\Delta; B)$ and $\bar{s}\langle \bar{t}/x\rangle = \sum_{1\le i\le n} \bar{s}_i$, then $\sum_{1\le i\le n} \Pi[\![\bar{s}_i]\!] = [\![\bar{s}]\!]\langle\!\langle [\![\bar{t}]\!]/x\rangle\!\rangle$;
(3) if $\vec{s} \in \mathsf{Sq}(\Delta; \vec{B})$ and $\vec{s}\langle \bar{t}/x\rangle = \sum_{1\le i\le n} \vec{s}_i$, then $\sum_{1\le i\le n} \langle\!| [\![\vec{s}_i]\!] |\!\rangle = [\![\vec{s}]\!]\langle\!\langle [\![\bar{t}]\!]/x\rangle\!\rangle$;

Remark that the hypothesis for bags and sequences must be stated carefully: syntax substitution yields sums of bags and sequences whereas the semantic substitution is not stable under sums.

Case 1. Assume $s \in \mathsf{Tm}(\Delta; B)$. We have three possibilities.

▸ **If s is an abstraction:** We consider $\Delta \vdash_{\mathsf{Tm}} \lambda y.u : C \to D$.

By definition of the substitution, we have:

$$(\lambda y.u)\,\langle \bar{t}/x\rangle = \lambda y.\,(u\langle \bar{t}/x\rangle)\ .$$

Writing Ω for $\Gamma, y : C$, we compute:

$$\begin{aligned}
&[\![\lambda y.u]\!]\langle\!\langle [\![\bar{t}]\!]/x\rangle\!\rangle\\
&= [\![\lambda y.u]\!] \circ \mathbb{M}_{\Gamma,(x:A)} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle\\
&= \Lambda_{\Delta,C,D}\left([\![\Delta, y : C \vdash_{\mathsf{Tm}} u : D]\!] \circ \mathbb{M}_{\Delta,(y:C)}\right) \circ \mathbb{M}_{\Gamma,(x:A)} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle\\
&= \Lambda_{\Gamma,C,D}\left([\![u]\!] \circ \mathbb{M}_{\Delta,(y:C)} \circ \left(\left(\mathbb{M}_{\Gamma,(x:A)} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle\right) \otimes \mathsf{id}_{(y:C)}\right)\right)\\
&= \Lambda_{\Gamma,C,D}\left([\![u]\!] \circ \mathbb{M}_{\Omega,(x:A)} \circ \langle \mathbb{M}_{\Gamma,(y:C)}, \Pi[\![\bar{t}]\!] \circ \pi_\ell\rangle\right)\\
&= \Lambda_{\Gamma,C,D}\left([\![u]\!] \circ \mathbb{M}_{\Omega,(x:A)} \circ \langle \mathsf{id}_\Omega, \Pi[\![\bar{t}]\!] \circ \pi_\ell \circ \mathbb{M}^{-1}_{\Gamma,(y:C)}\rangle \circ \mathbb{M}_{\Gamma,(y:C)}\right)\\
&= \Lambda_{\Gamma,C,D}\left([\![u]\!] \circ \mathbb{M}_{\Omega,(x:A)} \circ \langle \mathsf{id}_\Omega, \Pi[\![\Omega \vdash_{\mathsf{Bg}} \bar{t} : A]\!]\rangle \circ \mathbb{M}_{\Gamma,(y:C)}\right)\\
&= \Lambda_{\Gamma,C,D}\left([\![u]\!]\langle\!\langle [\![\bar{t}]\!]/x\rangle\!\rangle \circ \mathbb{M}_{\Gamma,(y:C)}\right)\\
&= \Lambda_{\Gamma,C,D}\left([\![\Omega \vdash_{\mathsf{Tm}} u\langle \bar{t}/x\rangle : B]\!] \circ \mathbb{M}_{\Gamma,(y:C)}\right)\\
&= [\![\Gamma \vdash_{\mathsf{Tm}} \lambda y.\,(u\langle \bar{t}/x\rangle) : B]\!]
\end{aligned}$$

by definition of the substitution; definition of the interpretation; naturality of Λ; a (lengthy) diagram chasing; Lemma 7.10 and the fact that $\mathbb{M}$ is a comonoid morphism; Lemma 7.15; definition of the semantic substitution; induction hypothesis; and finally definition of the interpretation.

▸ **If s is an application $u\,\bar{v}$:** We consider $\Delta \vdash_{\mathsf{Tm}} u\,\bar{v} : B$.

By definition of the substitution, we have:

$$(u\,\bar{v})\,\langle \bar{t}/x\rangle = \sum_{\bar{t} \lhd \bar{t}_1 * \bar{t}_2} (u\langle \bar{t}_1/x\rangle)\ (\bar{v}\langle \bar{t}_2/x\rangle\ )\ .$$

Writing $\Lambda\!\!\!\Lambda$ for $\Lambda\!\!\!\Lambda_{[\![\Gamma]\!],[\![x:A]\!]}$, we now compute:

$$
\begin{aligned}
& [\![u\ \bar{v}]\!]\langle\!\langle[\![\bar{t}]\!]/x\rangle\!\rangle \\
&= [\![u\ \bar{v}]\!] \circ \Lambda\!\!\!\Lambda \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle \\
&= \mathrm{ev}_{C,B} \circ \langle [\![u]\!], \Pi[\![\bar{v}]\!]\rangle \circ \Lambda\!\!\!\Lambda \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle \\
&= \mathrm{ev}_{C,B} \circ \langle [\![u]\!] \circ \Lambda\!\!\!\Lambda, \Pi[\![\bar{v}]\!] \circ \Lambda\!\!\!\Lambda\rangle \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle \\
&= \sum_{[\![\bar{t}]\!] \triangleleft \bar{g}_1 * \bar{g}_2} \mathrm{ev}_{C,B} \circ \langle [\![u]\!] \circ \Lambda\!\!\!\Lambda \circ \langle \mathsf{id}_\Gamma, \Pi\bar{g}_1\rangle, \Pi[\![\bar{v}]\!] \circ \Lambda\!\!\!\Lambda \circ \langle \mathsf{id}_\Gamma, \Pi\bar{g}_2\rangle\rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} \mathrm{ev}_{C,B} \circ \langle [\![u]\!] \circ \Lambda\!\!\!\Lambda \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_1]\!]\rangle, \Pi[\![\bar{v}]\!] \circ \Lambda\!\!\!\Lambda \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_2]\!]\rangle\rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} \mathrm{ev}_{C,B} \circ \langle [\![u]\!]\langle\!\langle[\![\bar{t}_1]\!]/x\rangle\!\rangle, [\![\bar{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} \mathrm{ev}_{C,B} \circ \langle [\![u\langle\bar{t}_1/x\rangle]\!], \sum_{i\in I} \Pi[\![\bar{v}_{\bar{t}_2,i}]\!]\rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} \sum_{i\in I} \mathrm{ev}_{C,B} \circ \langle [\![u\langle\bar{t}_1/x\rangle]\!], \Pi[\![\bar{v}_{\bar{t}_2,i}]\!]\rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} \sum_{i\in I} [\![(u\langle\bar{t}_1/x\rangle)\ (\bar{v}_{\bar{t}_2,i})]\!] \\
&= \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} [\![(u\langle\bar{t}_1/x\rangle)\ (\bar{v}\langle\bar{t}_2/x\rangle)]\!] \\
&= [\![(u\ \bar{v})\langle\bar{t}/x\rangle]\!]
\end{aligned}
$$

by definition of the semantic substitution; definition of the interpretation, assuming we have $\Delta \vdash_{\mathsf{Tm}} u : C \multimap B$ and $\Delta \vdash_{\mathsf{Bg}} \bar{v} : B$; Lemma 7.10; Lemma 7.7; the observation that the partitions of $\bar{t}$ coincide with the partitions of $[\![\bar{t}]\!]$; definition of the semantic substitution again; induction hypothesis on u and $\bar{v}$, assuming

$$\bar{v}\langle\bar{t}_2/x\rangle = \sum_{i\in I} \bar{v}_{\bar{t}_2,i}\,; \tag{7.2}$$

compatibility of everything with the additive structure; definition of the interpretation again; definition of the interpretation of a sum and (7.2); and finally definition of the substition.

► **If s is an application $y\,\vec{v}$:** Again, we have subcases:

▷ **If the head variable is x:** We consider $\Delta \vdash_{\mathsf{Tm}} x\,\vec{v} : \alpha$.

By definition of the substitution, we have:

$$
\begin{aligned}
(x\,\vec{v})\langle\bar{t}/x\rangle &= \sum_{\bar{t} \triangleleft \bar{t}_1 * \bar{t}_2} (x\langle\bar{t}_1/x\rangle)\ (\vec{v}\langle\bar{t}_2/x\rangle\ ) \\
&= \sum_{\bar{t} \triangleleft [u] * \bar{t}'} u\ (\vec{v}\langle\bar{t}'/x\rangle\ )
\end{aligned}
$$

Writing $\wedge\!\!\!\wedge$ for $\wedge\!\!\!\wedge_{\Gamma,(x:A)}$, we compute:

$$
\begin{aligned}
& [\![x\,\vec{v}]\!]\langle\!\langle[\![\bar{t}]\!]/x\rangle\!\rangle \\
={}& [\![x\,\vec{v}]\!] \circ \wedge\!\!\!\wedge \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle \\
={}& \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \mathrm{var}^{\Delta}_{x}, \langle\!|[\![\vec{v}]\!]|\!\rangle\rangle \circ \wedge\!\!\!\wedge \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle \\
={}& \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \mathrm{var}^{\Delta}_{x} \circ \wedge\!\!\!\wedge, \langle\!|[\![\vec{v}]\!]|\!\rangle \circ \wedge\!\!\!\wedge\rangle \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!]\rangle \\
={}& \sum_{\bar{t}\lhd\bar{t}_1*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \mathrm{var}^{\Delta}_{x} \circ \wedge\!\!\!\wedge \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_1]\!]\rangle, \langle\!|[\![\vec{v}]\!]|\!\rangle \circ \wedge\!\!\!\wedge \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_2]\!]\rangle\rangle \\
={}& \sum_{\bar{t}\lhd\bar{t}_1*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \mathrm{var}^{\Delta}_{x} \circ \wedge\!\!\!\wedge \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_1]\!]\rangle, [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
={}& \sum_{\bar{t}\lhd\bar{t}_1*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \zeta_A \circ \pi_{r} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_1]\!]\rangle, [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
={}& \sum_{\bar{t}\lhd\bar{t}_1*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \zeta_A \circ \Pi[\![\bar{t}_1]\!], [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
={}& \sum_{\bar{t}\lhd\bar{t}_1*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \zeta_A \circ \mathsf{id}^{\bullet}_{A} \circ \Pi[\![\bar{t}_1]\!], [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
={}& \sum_{\bar{t}\lhd[u]*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \zeta_A \circ [\![u]\!], [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
={}& \sum_{\bar{t}\lhd[u]*\bar{t}_2} [\![u\ (\vec{v}\langle\bar{t}_2/x\rangle)]\!] \\
={}& [\![(x\,\vec{v})\,\langle\bar{t}/x\rangle]\!]
\end{aligned}
$$

by definition of the semantic substitution; definition of the interpretation; Lemma 7.10; Lemma 7.13; definition of the semantic substitution; Lemma 7.16 case (1); Lemma 7.10; Lemma 7.17; Lemma 7.8; and the last lines are as in the previous case using Lemma 7.18.

▷ **If the head variable is $y \neq x$:** We consider $\Delta \vdash_{\mathsf{Tm}} y\,\vec{v} : \alpha$.

By definition of the substitution, we have:

$$(y\,\vec{v})\,\langle\bar{t}/x\rangle = y\ (\vec{v}\langle\bar{t}/x\rangle)$$

As in the previous subcase, we compute:

$$
\begin{aligned}
& [\![x\,\vec{v}]\!]\langle\!\langle[\![\bar{t}]\!]/x\rangle\!\rangle \\
={}& \sum_{\bar{t}\lhd\bar{t}_1*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \mathrm{var}^{\Delta}_{y} \circ \wedge\!\!\!\wedge \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_1]\!]\rangle, [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
={}& \sum_{\bar{t}\lhd\bar{t}_1*\bar{t}_2} \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \mathrm{var}^{\Gamma}_{y} \circ \pi_{\ell} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_1]\!]\rangle, [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}_2]\!]/x\rangle\!\rangle\rangle \\
={}& \mathrm{ev}_{\vec{C},\alpha} \circ \langle \mathsf{id}^{\bullet}_{\vec{C}\to\alpha} \circ \mathrm{var}^{\Gamma}_{y}, [\![\vec{v}]\!]\langle\!\langle[\![\bar{t}]\!]/x\rangle\!\rangle\rangle \\
={}& [\![(y\,\vec{v})\,\langle\bar{t}/x\rangle]\!]
\end{aligned}
$$

with the same justifications as before; Lemma 7.16 case (2); Lemma 7.12; and the same justifications as before.

Case 2. Assume $\bar{s} := [s_i \mid 1 \le i \le n] \in \mathsf{Bg}(\Delta; B)$.

Writing $\mathbin{/\!\!\!\backslash}$ for $\mathbin{/\!\!\!\backslash}_{\Gamma,(x:A)}$, we compute:

$$\begin{aligned}
& [\![\bar{s}]\!]\langle\!\langle [\![\bar{t}]\!]/x \rangle\!\rangle \\
&= \Pi[\![\bar{s}]\!] \circ \mathbin{/\!\!\!\backslash} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!] \rangle \\
&= \left(\prod_{1 \le i \le n} [\![s_i]\!] \circ \mathbin{/\!\!\!\backslash} \right) \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!] \rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_i * \ldots \bar{t}_n} \prod_{1 \le i \le n} [\![s_i]\!] \circ \mathbin{/\!\!\!\backslash} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_i]\!] \rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_i * \ldots \bar{t}_n} \prod_{1 \le i \le n} [\![s_i]\!]\langle\!\langle [\![\bar{t}_i]\!]/x \rangle\!\rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_i * \ldots \bar{t}_n} \prod_{1 \le i \le n} [\![s_i\langle \bar{t}_i/x \rangle]\!]
\end{aligned}$$

by definition of the semantic substitution; Lemma 7.11; Lemma 7.11 again; definition of the semantic substitution; and finally induction hypothesis.

Case 3. Assume $\vec{s} := \langle \bar{s}_i \mid 1 \le i \le n \rangle \in \mathsf{Sq}(\Delta; \vec{B})$.

Writing $\mathbin{/\!\!\!\backslash}$ for $\mathbin{/\!\!\!\backslash}_{\Gamma,(x:A)}$, we compute:

$$\begin{aligned}
& [\![\vec{s}]\!]\langle\!\langle [\![\bar{t}]\!]/x \rangle\!\rangle \\
&= \langle\!| [\![\vec{s}]\!] |\!\rangle \circ \mathbin{/\!\!\!\backslash} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!] \rangle \\
&= \langle \Pi[\![\bar{s}_i]\!] \mid 1 \le i \le n \rangle \circ \mathbin{/\!\!\!\backslash} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!] \rangle \\
&= \langle \Pi[\![\bar{s}_i]\!] \circ \mathbin{/\!\!\!\backslash} \mid 1 \le i \le n \rangle \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!] \rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_i * \ldots \bar{t}_n} \langle \Pi[\![\bar{s}_i]\!] \circ \mathbin{/\!\!\!\backslash} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}_i]\!] \rangle \mid 1 \le i \le n \rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_i * \ldots \bar{t}_n} \langle [\![\bar{s}_i]\!]\langle\!\langle [\![\bar{t}_i]\!]/x \rangle\!\rangle \mid 1 \le i \le n \rangle \\
&= \sum_{\bar{t} \triangleleft \bar{t}_i * \ldots \bar{t}_n} \langle [\![\bar{s}_i\langle \bar{t}_i/x \rangle]\!] \mid 1 \le i \le n \rangle
\end{aligned}$$

by definition of the semantic substitution; definition of packing; Lemma 7.10; Lemma 7.13; definition of the semantic substitution; and finally induction hypothesis. □

7.3.4 Soundness

From the substitution lemma above, we deduce that the interpretation is invariant under reduction.

Theorem 7.20 – Soundness

Consider $S \in \Sigma\mathsf{Tm}(\Gamma; A)$. If $S \rightsquigarrow S'$ then $[\![S]\!] = [\![S']\!]$.

Proof. Preservation of β-reduction follows from Lemma 7.19. Invariance for bags and sequences must be stated carefully: reduction yields sums of bags and sequences whereas the sets $\mathsf{Bg}_{\mathscr{C}}(\Gamma, A)$ and $\mathsf{Sq}_{\mathscr{C}}(\Gamma, \vec{A})$ are not stable under sums.

Toplevel reduction. Consider a redex $\Gamma \vdash_{\mathsf{Tm}} (\lambda x.s)\ \bar{t} : A$. Then:

$$
\begin{aligned}
& [\![(\lambda x.s)\ \bar{t}]\!] \\
= {} & \mathrm{ev}_{B\to A} \circ \langle [\![\lambda x.s]\!], \Pi[\![\bar{t}]\!] \rangle \\
= {} & \mathrm{ev}_{B\to A} \circ \langle \Lambda_{\Gamma,B,A}\left([\![\Gamma, x : B \vdash_{\mathsf{Tm}} s : A]\!] \circ \mathbb{M}_{\Gamma,(x:B)}\right), \Pi[\![\bar{t}]\!] \rangle \\
= {} & \mathrm{ev}_{B\to A} \circ \left(\Lambda_{\Gamma,B,A}\left([\![s]\!] \circ \mathbb{M}_{\Gamma,(x:B)}\right) \otimes \mathsf{id}_A\right) \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!] \rangle \\
= {} & [\![s]\!] \circ \mathbb{M}_{\Gamma,(x:B)} \circ \langle \mathsf{id}_\Gamma, \Pi[\![\bar{t}]\!] \rangle \\
= {} & [\![s]\!]\langle\!\langle [\![\bar{t}]\!]/x \rangle\!\rangle \\
= {} & [\![s\langle \bar{t}/x \rangle]\!]
\end{aligned}
$$

by definition of the interpretation (with $\Gamma \vdash_{\mathsf{Tm}} \lambda x.s : B \to A$); definition of the interpretation again; smcc laws and the definition of tupling; equations of monoidal closure; the definition of semantic substitution; and finally Lemma 7.19.

Context closure. To show that invariance extends by context closure, we prove the three statements:

(1) if $s \in \mathsf{Tm}(\Gamma; A)$ and $s \rightsquigarrow S'$ then $[\![s]\!] = [\![S']\!]$;
(2) if $\bar{s} \in \mathsf{Bg}(\Gamma; A)$ and $\bar{s} \rightsquigarrow \sum_{i\in I} \bar{s}_i$ then $\Pi[\![\bar{s}]\!] = \sum_{i\in I} \Pi[\![\bar{s}_i]\!]$;
(3) if $\vec{s} \in \mathsf{Sq}(\Gamma; \vec{A})$ and $\vec{s} \rightsquigarrow \sum_{i\in I} \vec{s}_i$ then $\langle\!|[\![\vec{s}]\!]|\!\rangle = \sum_{i\in I} \langle\!|[\![\vec{s}_i]\!]|\!\rangle$

by mutual induction, following the inductive definition of context closure. Finally, this extends to sums as required. □

7.4 How to build your own resource category

Resource calculus is closely related to Erhrard and Regnier's differential lambda-calculus [20], which is usually interpreted using differential categories (introduced in [7] as a categorical framework for differential linear logic).

[20]: Ehrhard and Regnier (2003), 'The differential lambda-calculus'
[7]: Blute, Cockett, and Seely (2006), 'Differential categories'

However, here we study resource calculus in relation with games, and strategies of pointer concurrent games are not built from a model of linear logic: their categorical structure is not a differential category. Nevertheless, resource categories are built using similar constructions to some differential categories, more precisely *monoidal storage categories* as described in [8].

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

The intuition behind these similarities is that the exponential ! of differential categories allows us to go from linear morphisms from A to B, to morphisms from $!A$ to $!B$, which behave linearly with respect to $!A$ and $!B$, but not with respect to the original objects A and B. These intuitions will guide us in our construction of resource categories from additive monoidal storage categories – which are the categories we mostly refer to when mentioning "differential categories" in this section, although differential categories in general are a much wider notion. Our main focus here is not to give an exhaustive presentation of differential categories, but rather to present the particular categorical structure which we will use to build a resource category.

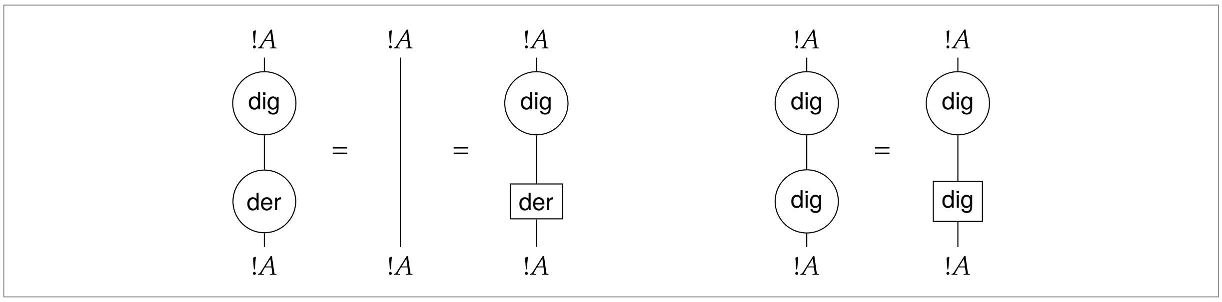
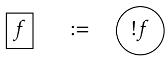


Figure 7.8: Comonad laws

7.4.1 Additive monoidal storage categories

Coalgebra modality. *Coalgebra modalities* are similar to comonoids (Definition 1.6), but they build over a comonad.

Notation: Recall that we use squared boxes for ! applied to morphisms:



Remark: We write dig and der for the natural transformations because they match the *digging* and *dereliction* rules of linear logic (introduced in [24]).

[24]: Girard (1987), 'Linear logic'

Definition 7.21 – Comonad

Consider a category $\mathscr{C}$. A **comonad** on $\mathscr{C}$ is $(!, \mathsf{dig}, \mathsf{der})$ with

$$\begin{aligned} !: \mathscr{C} &\to \mathscr{C} && \text{an endofunctor,} \\ \mathsf{dig}_A: !A &\to !!A && \text{a natural transformation,} \\ \mathsf{der}_A: !A &\to A && \text{a natural transformation,} \end{aligned}$$

satisfying the equations of Figure 7.8.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Definition 7.22 – Coalgebra modality [8, Definition 1]

A **coalgebra modality** on a symmetric monoidal category $\mathscr{C}$ is $(!, \mathsf{dig}, \mathsf{der}, \Delta, \mathsf{e})$ with $(!, \mathsf{dig}, \mathsf{der})$ a comonad and two natural transformations

$$\Delta_A: !A \to !A \otimes !A \qquad \mathsf{e}_A: !A \to I$$

such that for any A, $(!A, \Delta_A, \mathsf{e}_A)$ is a comutative comonoid (Definition 1.7) and dig preserves Δ in the sense of Figure 7.9.

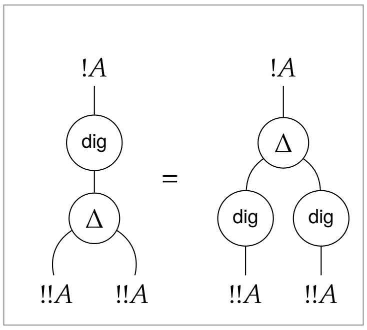


Figure 7.9: Coalgebra modality

Bialgebra modality. Next, we define *bialgebra modalities*, which again are reminiscent of bialgebras seen in previous sections.

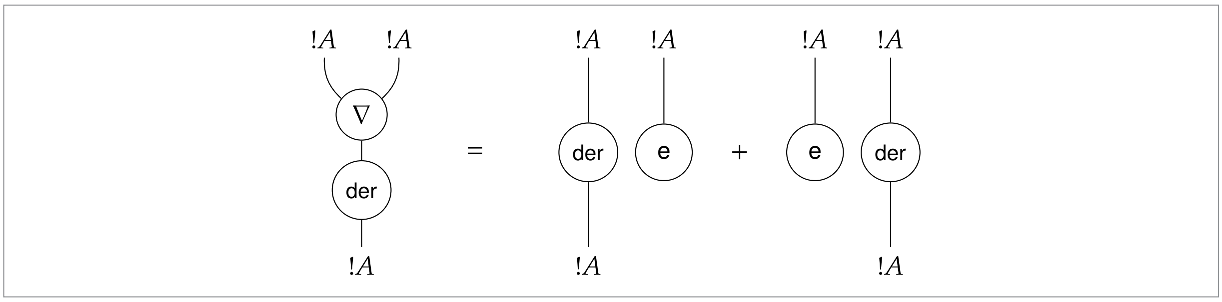


Figure 7.10: Bialgebra modality

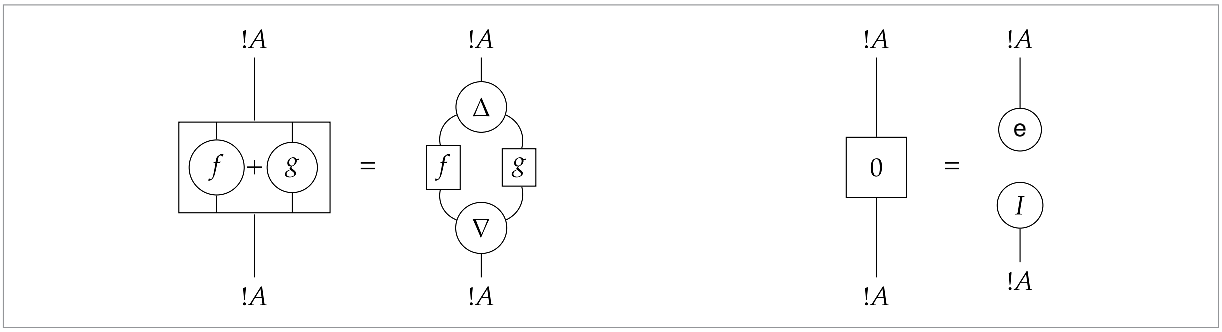


Figure 7.11: Addititve Bialgebra Modality Laws

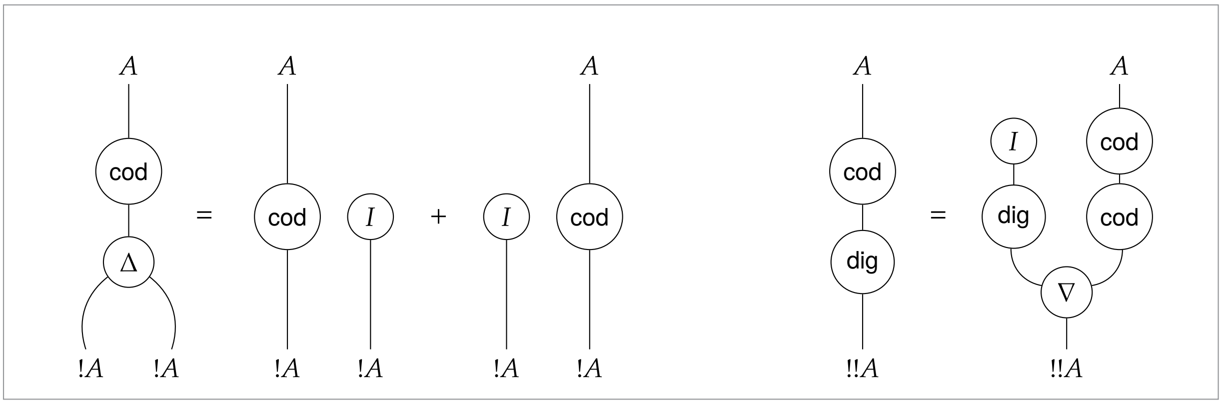


Figure 7.12: *Product rule* and *chain rule* of codereliction

Definition 7.23 – Bialgebra modality [8, Definition 4]

A **bialgebra modality** on an asmc $\mathscr{C}$ is $(!, \mathsf{dig}, \mathsf{der}, \Delta, \mathsf{e}, \nabla, I)$ with $(!, \mathsf{dig}, \mathsf{der}, \Delta, \mathsf{e})$ a coalgebra modality and for any A, a bialgebra $(!A, \Delta_A, \mathsf{e}_A, \nabla_A, I_A)$ following the equation of Figure 7.10.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Definition 7.24 – Additive bialgebra modality [8, Definition 5]

An **additive bialgebra modality** in an asmc $\mathscr{C}$ is a bialgebra modality $(!, \mathsf{dig}, \mathsf{der}, \Delta, \mathsf{e}, \nabla, I)$ compatible with the additive structure in the sense of Figure 7.11.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Additive bialgebra modalities can be equipped with a *codereliction*, a natural transformation $\mathsf{cod}_A : A \longrightarrow !A$ named codereliction because it has the inverse type to der_A, but which is *not* an inverse of der_A.

Definition 7.25 – Codereliction [8, Definition 9][3]

Consider an asmc $\mathscr{C}$. A **codereliction** for an additive bialgebra modality $(!, \mathsf{dig}, \mathsf{der}, \Delta, \mathsf{e}, \nabla, I)$ is a natural transformation $\mathsf{cod}_A : A \longrightarrow !A$ satisfying the following equations:

$$\mathsf{e}_A \circ \mathsf{cod}_A = 0 \qquad (\textit{constant rule})$$

$$\mathsf{der}_A \circ \mathsf{cod}_A = \mathsf{id}_A \qquad (\textit{linear rule})$$

as well as the equations of Figure 7.12.

3: The chain rule equation given here is the version presented in [23] and not the (slightly longer) version of [7, Definition 4.11]; however both are equivalent in monoidal storage categories ([8, Lemma 7 and Corollary 5]).

[23]: Fiore (2007), 'Differential Structure in Models of Multiplicative Biadditive Intuitionistic Linear Logic'

[7]: Blute, Cockett, and Seely (2006), 'Differential categories'

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Codereliction is a key notion of differential categories: in an asmc with a bialgebra modality, coderelictions *induce* deriving transformations[4] ([7, Theorem 4.12]). In an asmc with an *additive* bialgebra modality, codereliction are *in bijection* with deriving transformations ([8, Theorem 4]).

4: More precisely they are in bijection with deriving transformations satisfying the ∇- rule of [7].

[7]: Blute, Cockett, and Seely (2006), 'Differential categories'

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Storage Categories. Now, we focus on *storage categories*, which are smcs with a coalgebra modality and a cartesian product &, with the following isomorphism:

$$!(A\&B) \cong !A \otimes !B$$

called **Seely isomorphism** (introduced as "Δ iso" in [38]).

[38]: Seely (1989), 'Linear Logic, ∗-Autonomous Categories and Cofree Coalgebras'

Recall that in a category $\mathscr{C}$, a *terminal object* is an object T such that for any object $A \in \mathscr{C}$, there exists a unique morphism in $\mathscr{C}(A, \mathrm{T})$, noted $\top_A : A \longrightarrow \mathrm{T}$. A category $\mathscr{C}$ has *finite products* if it has a terminal object and for all objects $A, B \in \mathscr{C}$, there is a product $(A\&B, \pi_A, \pi_B)$ in $\mathscr{C}$ satisfying the universal property of products.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Definition 7.26 – Seely Isomorphism [8, Definition 10]

Consider an smc $\mathscr{C}$ with a binary product &, a terminal object T, and a coalgebra modality $(!, \mathsf{dig}, \mathsf{der}, \Delta, \mathsf{e})$. It has **Seely isomorphisms** if the map χ_{T}, defined as:

$$\chi_{\mathrm{T}} : !\mathrm{T} \xrightarrow{\mathsf{e}_{\mathrm{T}}} I \,,$$

and the natural transformation χ, defined as:

$$\chi_{A,B} : !(A\&B) \xrightarrow{\Delta_{A\&B}} !(A\&B) \otimes !(A\&B) \xrightarrow{!\pi_A \otimes !\pi_B} !A \otimes !B \,,$$

are isomorphisms.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Definition 7.27 – Monoidal Storage Category [8, Definition 10]

A **monoidal storage category** is a smc with finite products and a coalgebra modality with Seely isomorphisms.

We can consider storage categories with an additive structure.

Definition 7.28 – Additive Monoidal Storage Category

An **additive monoidal storage category** [8, Definition 11] is a category $\mathscr{C}$ that is a monoidal storage category and an additive symmetric monoidal category, with the same monoidal structure.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

Additive storage categories are actually related to asmcs with a bialgebra modality.

Proposition 7.29 – from [8, Theorem 6]

Consider an additive monoidal storage category $\mathscr{C}$.

Then we define $(!, \mathsf{dig}, \mathsf{der}, \Delta, \mathsf{e}, \nabla, I)$ with:

$$\begin{aligned}
\Delta_A &: !A \xrightarrow{!\langle \mathsf{id}_A, \mathsf{id}_A \rangle} !(A \& A) \xrightarrow{\chi_{A,A}} !A \otimes !A \\
\mathsf{e}_A &: !A \xrightarrow{!0} !\top \xrightarrow{\chi_\top} I \\
\nabla_A &: !A \otimes !A \xrightarrow{\chi^{-1}_{A,A}} !(A \& A) \xrightarrow{\pi_1 + \pi_2} !A \\
I_A &: I \xrightarrow{\chi^{-1}_\top} !\top \xrightarrow{!0} !A
\end{aligned}$$

and it is a bialgebra modality.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

In [8], the authors even prove that those additive storage categories are equivalent to asmcs with a bialgebra structure.

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

7.4.2 The construction

We start from an additive monoidal storage category.

Definition 7.30 – $\mathsf{Res}(-)$

Consider an additive monoidal storage category $\mathscr{C}$ with a codereliction cod. Using the notation of Proposition 7.29, we define the category $\mathsf{Res}(\mathscr{C})$ with same objects as $\mathscr{C}$ and morphisms as:

$$\mathsf{Res}(\mathscr{C})(A, B) = \mathscr{C}(!A, !B) .$$

Definition 7.31 – Tensor for $\mathsf{Res}(-)$

We define a bifunctor $\otimes_{\mathsf{Res}(\mathscr{C})}$ in the following way:

$$\begin{aligned}
A \otimes_{\mathsf{Res}(\mathscr{C})} B &= A \& B \\
f \otimes_{\mathsf{Res}(\mathscr{C})} g &= \chi^{-1}_{C,D} \circ \big(f \otimes g\big) \circ \chi_{A,B}
\end{aligned}$$

for any objects A, B, C, D and morphisms $f \in \mathsf{Res}(\mathscr{C})(A, C)$ and $g \in \mathsf{Res}(\mathscr{C})(B, D)$.

Indeed, morphisms of a resource category do not all behave linearly, which is why we define $\mathsf{Res}(\mathscr{C})(A, B)$ as $\mathscr{C}(!A, !B)$: these are morphisms that are not necessarily linear with respect to A and B. To obtain a monoidal structure in $\mathsf{Res}(\mathscr{C})$, we prove that $\otimes_{\mathsf{Res}(\mathscr{C})}$ is a tensor, using Seely isomorphisms to see $!(A \otimes_{\mathsf{Res}(\mathscr{C})} B)$ as $!A \otimes !B$. The additive bialgebra modality structure of $\mathscr{C}$ easily induces a bialgebra structure in $\mathsf{Res}(\mathscr{C})$ (which we will define precisely in the next proof). Finally, recall the parting remark of Section 7.1: pointed identity laws are very similar to the dereliction and codereliction laws of differential categories. We will thus construct $\mathsf{id}^\bullet$ from der and cod.

Theorem 7.32

Consider an additive monoidal storage category $\mathscr{C}$ with a codereliction cod.

Then $\mathsf{Res}(\mathscr{C})$ is a resource category.

Proof. We use notations of Definition 7.30. To make the equations less cluttered, we write $\mathscr{R}$ for $\mathsf{Res}(\mathscr{C})$ and A for id_A, and we omit indices for χ when they are clear from the context.

SMC. We prove that $(\mathscr{R}, \otimes_{\mathscr{R}}, \mathrm{T})$ is a smc (Definition 1.2). We set:

$$\begin{aligned}
\alpha^{\mathscr{R}}_{A,B,C} &: \ !((A\&B)\&C) \xrightarrow{\chi} !(A\&B)\otimes !C \xrightarrow{\chi\otimes !C} (!A\otimes !B)\otimes !C \\
&\quad \xrightarrow{\alpha^{\mathscr{C}}_{!A,!B,!C}} !A \otimes (!B\otimes !C) \xrightarrow{!A\otimes\chi^{-1}} !A\otimes !(B\&C) \xrightarrow{\chi^{-1}} !(A\&(B\&C)) \\
\lambda^{\mathscr{R}}_{A} &: \ !(\mathrm{T}\&A) \xrightarrow{\chi} !\mathrm{T}\otimes !A \xrightarrow{\chi_{\mathrm{T}}\otimes !A} I\otimes !A \xrightarrow{\lambda^{\mathscr{C}}_{!A}} !A \\
\rho^{\mathscr{R}}_{A} &: \ !(A\&\mathrm{T}) \xrightarrow{\chi} !A\otimes !\mathrm{T} \xrightarrow{!A\otimes\chi_{\mathrm{T}}} !A \otimes I \xrightarrow{\rho^{\mathscr{C}}_{!A}} !A \\
\sigma^{\mathscr{R}}_{A,B} &: \ !(A\&B) \xrightarrow{\chi} !A\otimes !B \xrightarrow{\sigma^{\mathscr{C}}_{!A,!B}} !B\otimes !A \xrightarrow{\chi^{-1}} !(B\&A)
\end{aligned}$$

and a direct diagram chasing, using smc properties of $\mathscr{C}$ and the fact that χ is an isomorphism, shows that $\mathscr{R}$ is a smc too.

Additivity. Direct from the additive structure of $\mathscr{C}$.

Bialgebra structure. For any object A, we define the morphisms:

$$\begin{aligned}
\delta^{\mathscr{R}}_{A} &: \ !A \xrightarrow{\Delta^{\mathscr{C}}_{A}} !A\otimes !A \xrightarrow{\chi^{-1}} !(A\&A) & \qquad \epsilon^{\mathscr{R}}_{A} &: \ !A \xrightarrow{\mathsf{e}_A} I \xrightarrow{\chi^{-1}_{\mathrm{T}}} !\mathrm{T} \\
\mu^{\mathscr{R}}_{A} &: \ !(A\&A) \xrightarrow{\chi} !A\otimes !A \xrightarrow{\nabla_A} !A & \qquad \eta^{\mathscr{R}}_{A} &: \ !\mathrm{T} \xrightarrow{\chi_{\mathrm{T}}} I \xrightarrow{I_A} !A
\end{aligned}$$

Then one can check that $(A, \delta_A, \epsilon_A, \mu_A, \eta_A)$ is a bialgebra by diagram chasing, using χ and the properties of the bialgebra modality of $\mathscr{C}$. Likewise, we check that it is compatible with the monoidal structure of $\mathscr{R}$ (Figure 7.3).

Pointed Identity. Finally, for any object A, we define the pointed identity as:

$$\mathsf{id}^{\bullet}_{A} \ : \ !A \xrightarrow{\mathsf{der}_A} A \xrightarrow{\mathsf{cod}_A} !A$$

and we check that it matches Definition 7.3:

- *idempotent:*

$$\begin{aligned}
\mathsf{id}^{\bullet}_{A} \circ \mathsf{id}^{\bullet}_{A} &= \mathsf{cod}_A \circ \mathsf{der}_A \circ \mathsf{cod}_A \circ \mathsf{der}_A \\
&= \mathsf{cod}_A \circ \mathsf{id}_A \circ \mathsf{der}_A \\
&= \mathsf{id}^{\bullet}_{A}
\end{aligned}$$

  by definition of $\mathsf{id}^{\bullet}_{A}$ and linear rule of Definition 7.25.

- *non-erasable:*

$$\begin{aligned}\epsilon^{\mathcal{R}}_A \circ \mathsf{id}^\bullet_A &= \chi_{\mathrm{T}}^{-1} \circ \mathsf{e}_A \circ \mathsf{cod}_A \circ \mathsf{der}_A \\ &= \chi_{\mathrm{T}}^{-1} \circ 0 \circ \mathsf{der}_A \\ &= 0\end{aligned}$$

by definition; constant rule of Definition 7.25 and additivity.

- *non-erasing:*

$$\begin{aligned}\mathsf{id}^\bullet_A \circ \eta^{\mathcal{R}}_A &= \mathsf{cod}_A \circ \mathsf{der}_A \circ I_A \circ \chi_{\mathrm{T}} && \text{(definition)} \\ &= \mathsf{cod}_A \circ 0 \circ \chi_{\mathrm{T}} && \text{([8, Lemma 2])} \\ &= 0 && \text{(additivity)}\end{aligned}$$

[8]: Blute, Cockett, Lemay, and Seely (2020), 'Differential Categories Revisited'

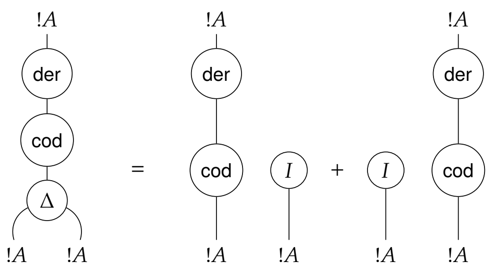
Remark: Actually $\mathsf{der}_A \circ I_A = 0$ was part of the original definition of bialgebra modalities ([7, Definition 4.8]), but it can be deduced from the other axioms and naturality of I and der ([8, Lemma 2]).

[7]: Blute, Cockett, and Seely (2006), 'Differential categories'

- *non-duplicable:*

$$\delta^{\mathcal{R}}_A \circ \mathsf{id}^\bullet_A = \chi_{A,A}^{-1} \circ \Delta_A \circ \mathsf{cod}_A \circ \mathsf{der}_A \qquad \text{(definition)}$$

and using string diagrams in $(\mathscr{C}, \otimes, I)$, we have:



by product rule (Definition 7.25, Figure 7.12) and additivity. Therefore,

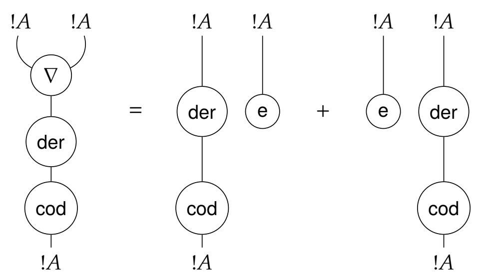
$$\delta^{\mathcal{R}}_A \circ \mathsf{id}^\bullet_A = \left(\mathsf{id}^\bullet_A \otimes_{\mathcal{R}} I_A\right) + \left(I_A \otimes_{\mathcal{R}} \mathsf{id}^\bullet_A\right)$$

again using Seely and the definition of $\mathsf{id}^\bullet$.

- *non-duplicative:*

$$\mathsf{id}^\bullet_A \circ \mu^{\mathcal{R}}_A = \mathsf{cod}_A \circ \mathsf{der}_A \circ \nabla_A \circ \chi A, A \qquad \text{(definition)}$$

which gives us, using string diagrams in $(\mathscr{C}, \otimes, I)$:



by compatibility of der and ∇ (Definition 7.23) and additivity; that is

$$\mathsf{id}^\bullet_A \circ \mu^{\mathcal{R}}_A = \left(\mathsf{id}^\bullet_A \otimes_{\mathcal{R}} \mathsf{e}_A\right) + \left(\mathsf{e}_A \otimes_{\mathcal{R}} \mathsf{id}^\bullet_A\right)$$

using Seely again and the definition of $\mathsf{id}^\bullet$. □

7.4.3 What about closeness?

Intuitively, a category $\mathscr{C}$ is closed if for any pair of objects A and B, $\mathscr{C}(A, B)$ can also be seen as an object of $\mathscr{C}$. In particular, for monoidal categories, $\mathscr{C}$ is *monoidal closed* if there exists $\multimap$ and Λ a bijection natural in A, B, C such that:

$$\Lambda_{A,B,C} : \mathscr{C}(A \otimes B, C) \cong \mathscr{C}(A, B \multimap C)$$

What happens if we consider $\mathscr{C}$ as in Theorem 7.32 a monoidal *closed* category? Does the closed structure also transport to $\mathsf{Res}(\mathscr{C})$? Let us try to prove the isomorphism above for $\mathscr{R} = \mathsf{Res}(\mathscr{C})$. Everything seems to go smoothly for the first part:

$$\begin{aligned} \mathscr{R}(A \otimes_{\mathscr{R}} B, C) &= \mathscr{C}(!\,(A \& B)\,, !C) && \text{(definition)} \\ &\cong \mathscr{C}(!A \otimes !B, !C) && \text{(Seely isomorphism)} \\ &\cong \mathscr{C}(!A, !B \multimap !C) && \text{(closed structure of } \mathscr{C}\text{)} \end{aligned}$$

All that is left to do now is to define $\multimap_{\mathscr{R}}$ such that

$$\mathscr{R}(A, B \multimap_{\mathscr{R}} C) = \mathscr{C}(!A, !B \multimap !C)\,,$$

but that is where the difficulty lies: there seems to be no obvious way to define $\multimap_{\mathscr{R}}$ such that $!(B \multimap_{\mathscr{R}} C) \cong !B \multimap !C$. In particular, it is clear that $!(B \multimap C)$ and $!B \multimap !C$ are not necessarily isomorphic. Hence, the question of whether or not we can build a *closed* resource category from a closed differential category remains open.

7.5 Conclusion and perspectives

There is still much to study on resource categories. For instance, we did not tackle yet the subject of cartesian structure for a resource category. However, the subcategory of comonoid morphisms is cartesian – to what strategies do they correspond in pointer concurrent games? Besides, morphisms interpreting finite resource terms do not form a subcategory, because they lack identities – how can we best describe their structure? What about finite strategies in general?

Resource categories were introduced to better understand the links between resource terms and strategies; we hope to generalize this correspondence to the Taylor expansion of λ-terms.

PCG and Resource-calculus 8

We now check that PCG is indeed a ressource category, such that the induced interpretation of normal forms coincides with the interpretations from Chapter 5, thus completing the proof.

8.1 PCG is a resource category

Recall that we already know PCG is a symmetric monoidal category (see Theorem 6.60).

8.1.1 Additive structure

We start by checking that PCG is enriched over commutative monoids. Consider two arenas A, B, then $\mathsf{PCG}[\mathsf{A}, \mathsf{B}]$ comes with an additive structure with, for any $\sigma, \tau \colon \mathsf{G} \vdash \mathsf{A}$, the sum $\sigma + \tau \colon \mathsf{G} \vdash \mathsf{A}$ is defined as the formal sum:

$$\sigma + \tau \stackrel{\text{def}}{=} \sum_{\mathsf{q} \in \mathsf{Isog}(\mathsf{G} \vdash \mathsf{A})} (\sigma(\mathsf{q}) + \tau(\mathsf{q})) \cdot \mathsf{q}\,,$$

and 0 is the empty strategy ($\mathsf{supp}(0) = \emptyset$). The tensor and the composition are compatible with the additive structure, hence PCG is an asmc.

8.1.2 Bialgebra laws

Now, we look at the bialgebra structure. For an arena A, we start by defining the bialgebra morphisms, using the *contraction* renaming:

$$\begin{array}{rccc} \mathsf{c}_\mathsf{A} \colon & \mathsf{A} \otimes \mathsf{A} & \to & \mathsf{A} \\ & (1, \mathsf{a}) & \mapsto & \mathsf{a} \\ & (2, \mathsf{a}) & \mapsto & \mathsf{a}\,. \end{array}$$

Then the bialgebra morphisms are:

$$\delta_\mathsf{A} = \mathsf{id}_{\mathsf{A} \otimes \mathsf{A}} \ltimes \mathsf{c}_\mathsf{A}\,, \qquad \varepsilon_\mathsf{A} = 1_{\mathsf{A} \vdash \mathsf{I}}\,, \qquad \mu_\mathsf{A} = \mathsf{c}_\mathsf{A} \rtimes \mathsf{id}_{\mathsf{A} \otimes \mathsf{A}}\,, \qquad \eta_\mathsf{A} = 1_{\mathsf{I} \vdash \mathsf{A}}\,.$$

Using lemmas on renamings from Chapter 6, we check that those morphisms follow bialgebra laws. Most of them are quite easy to prove; the composition $\delta \odot \mu$ is more subtle and requires us to be very careful about composition and partitions of positions *versus* configurations.

Reminder: We write I for the empty arena. For any arena B, 1_B is the strategy on B with only the empty isogmentation 0 in its support, with coefficient 1.

Lemma 8.1 – Coalgebra laws

Consider an arena A, then $(\mathsf{A}, \delta_\mathsf{A}, \varepsilon_\mathsf{A})$ is a commutative comonoid (Definition 1.7).

Lemmas (and proposition) used:
– 6.53: renamming of a composition;
– 6.31: neutrality of copycat;
– 6.51: identity renaming;
– 6.56: tensor of renamings;
– 6.38: tensor of identities;
– 6.54: inverse of a renaming;
– 6.52: composition of renamings.

Proof. ***Associativity.*** We have:

$$
\begin{aligned}
&\alpha_{A,A,A} \odot (\delta_A \otimes \mathsf{id}_A) \odot \delta_A \\
&= \left(\mathbf{a}_{A,A,A} \odot \mathsf{id}_{(A\otimes A)\otimes A}\right) \odot \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A \otimes \mathsf{id}_A\right) \odot \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A\right) \\
&= \mathbf{a}_{A,A,A} \ltimes \left(\mathsf{id}_{(A\otimes A)\otimes A} \odot \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A \otimes \mathsf{id}_A\right)\right) \odot \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A\right) \\
&= \mathbf{a}_{A,A,A} \rtimes \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A \otimes \mathsf{id}_A\right) \odot \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A\right) \\
&= \left(\mathbf{a}_{A,A,A} \rtimes \left(\left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A \otimes \mathsf{id}_A\right) \odot \mathsf{id}_{A\otimes A}\right)\right) \ltimes \mathbf{c}_A \\
&= \left(\mathbf{a}_{A,A,A} \rtimes \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A \otimes \mathsf{id}_A\right)\right) \ltimes \mathbf{c}_A \\
&= \left(\mathbf{a}_{A,A,A} \rtimes \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A \otimes \mathsf{id}_A \ltimes \mathbf{id}_A\right)\right) \ltimes \mathbf{c}_A \\
&= \left(\mathbf{a}_{A,A,A} \rtimes \left(\left(\mathsf{id}_{A\otimes A} \otimes \mathsf{id}_A\right) \ltimes \left(\mathbf{c}_A \times \mathbf{id}_A\right)\right)\right) \ltimes \mathbf{c}_A \\
&= \left(\left(\mathbf{a}_{A,A,A} \rtimes \mathsf{id}_{(A\otimes A)\otimes A}\right) \ltimes \left(\mathbf{c}_A \times \mathbf{id}_A\right)\right) \ltimes \mathbf{c}_A \\
&= \left(\left(\mathsf{id}_{A\otimes(A\otimes A)} \ltimes \mathbf{a}^{-1}_{A,A,A}\right) \ltimes \left(\mathbf{c}_A \times \mathbf{id}_A\right)\right) \ltimes \mathbf{c}_A \\
&= \mathsf{id}_{A\otimes(A\otimes A)} \ltimes \left(\mathbf{c}_A \circ \left(\mathbf{c}_A \times \mathbf{id}_A\right) \circ \mathbf{a}^{-1}_{A,A,A}\right) \\
&= \mathsf{id}_{A\otimes(A\otimes A)} \ltimes \left(\mathbf{c}_A \circ \left(\mathbf{id}_A \times \mathbf{c}_A\right)\right) \\
&= \left(\mathsf{id}_A \otimes \mathsf{id}_{A\otimes A}\right) \ltimes \left(\mathbf{c}_A \circ \left(\mathbf{id}_A \times \mathbf{c}_A\right)\right) \\
&= \left(\left(\mathsf{id}_A \otimes \mathsf{id}_{A\otimes A}\right) \ltimes \left(\mathbf{id}_A \times \mathbf{c}_A\right)\right) \ltimes \mathbf{c}_A \\
&= \left(\mathsf{id}_A \otimes \delta_A\right) \odot \delta_A
\end{aligned}
$$

by definition; Lemma 6.53; Proposition 6.31; Lemma 6.53; Proposition 6.31; Lemma 6.51; Lemma 6.56; Lemma 6.38; Lemma 6.54; Lemma 6.52 twice; computation of the two renamings; Lemma 6.38; Lemma 6.52; and Lemma 6.56, Proposition 6.31, and definitions.

Neutrality. We have:

$$
\begin{aligned}
&\lambda_A \odot (\varepsilon_A \otimes \mathsf{id}_A) \odot \delta_A \\
&= \left(\mathbf{l}_A \rtimes \mathsf{id}_{I\otimes A}\right) \odot \left(1_{A\vdash I} \otimes \mathsf{id}_A\right) \odot \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A\right) \\
&= \mathbf{l}_A \rtimes \left(1_{A\vdash I} \otimes \mathsf{id}_A\right) \ltimes \mathbf{c}_A \\
&= \mathsf{id}_A
\end{aligned}
$$

by definition; Lemma 6.53 and Proposition 6.31; and direct computation using the definitions.

Commutativity. We have:

$$
\begin{aligned}
&\gamma_{A,A} \odot \delta_A \\
&= \left(\mathbf{s}_{A,A} \rtimes \mathsf{id}_{A\otimes A}\right) \odot \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A\right) \\
&= \left(\left(\mathbf{s}_{A,A} \rtimes \mathsf{id}_{A\otimes A}\right) \odot \mathsf{id}_{A\otimes A}\right) \ltimes \mathbf{c}_A \\
&= \left(\mathbf{s}_{A,A} \rtimes \mathsf{id}_{A\otimes A}\right) \ltimes \mathbf{c}_A \\
&= \left(\mathsf{id}_{A\otimes A} \ltimes \mathbf{s}_{A,A}\right) \ltimes \mathbf{c}_A \\
&= \mathsf{id}_{A\otimes A} \ltimes \left(\mathbf{c}_A \circ \mathbf{s}_{A,A}\right) \\
&= \mathsf{id}_{A\otimes A} \ltimes \mathbf{c}_A \\
&= \delta_A
\end{aligned}
$$

by definition; Lemma 6.53; Proposition 6.31; Lemma 6.54; Lemma 6.52; computation of the renamings; and definition. □

Likewise, μ_{A} and η_{A} respect algebra laws – since they are completely symmetric we don't detail the proofs.

Lemma 8.2 – Algebra laws

Consider an arena A, then $(\mathsf{A}, \mu_{\mathsf{A}}, \eta_{\mathsf{A}})$ is a commutative monoid (Definition 1.5).

Finally, we look at the additional bialgebra laws.

Lemma 8.3 – Bialgebra laws

Consider an arena A. Then δ_{A}, ε_{A}, μ_{A} and η_{A} follow the additional bialgebra laws of Figure 7.1.

Proof. ***(a)*** The intuition behind the distributivity law should be rather clear. Given a strategy on $\mathsf{G} \vdash \mathsf{A} \otimes \mathsf{A}$, the multiplication μ_{A} will "flatten" the isogmentations on a single copy of A, and the comultiplication δ_{A} will distribute these isogmentations to the two sides of $\mathsf{A} \otimes \mathsf{A}$. This should be the same as taking isogmentations on $\mathsf{G} \vdash \mathsf{A} \otimes \mathsf{A}$, distributing their left and right sides, and then gathering everything to $\mathsf{A} \otimes \mathsf{A}$ again. However, the actual proof is very subtle and requires a lot of computation. In order to try and keep the current proof to a reasonnable length, the technical details for the distributivity law are presented in next subsection.

Leaving aside the first law for now, we focus on the other three.

(b) We have:

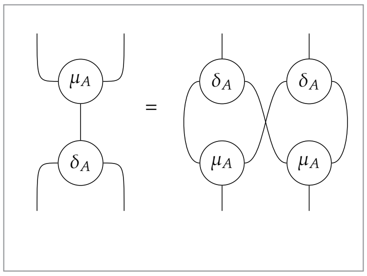
$$
\begin{aligned}
& \delta_{\mathsf{A}} \odot \eta_{\mathsf{A}} \odot \lambda_{\mathsf{I}} \\
= {} & (\mathsf{id}_{\mathsf{A}\otimes\mathsf{A}} \ltimes \mathsf{c}_{\mathsf{A}}) \odot 1_{\mathsf{I}\vdash\mathsf{A}} \odot (\mathbf{l}_{\mathsf{I}} \rtimes \mathsf{id}_{\mathsf{I}\otimes\mathsf{I}}) \\
= {} & 1_{\mathsf{I}\vdash\mathsf{A}\otimes\mathsf{A}} \odot (\mathbf{l}_{\mathsf{I}} \rtimes \mathsf{id}_{\mathsf{I}\otimes\mathsf{I}}) \\
= {} & 1_{\mathsf{I}\vdash\mathsf{A}\otimes\mathsf{A}} \odot \left(\mathsf{id}_{\mathsf{I}} \ltimes \mathbf{l}_{\mathsf{I}}^{-1}\right) \\
= {} & 1_{\mathsf{I}\vdash\mathsf{A}\otimes\mathsf{A}} \ltimes \mathbf{l}_{\mathsf{I}}^{-1} \\
= {} & (1_{\mathsf{I}\vdash\mathsf{A}} \otimes 1_{\mathsf{I}\vdash\mathsf{A}})
\end{aligned}
$$

by definition; computation of the composition; Lemma 6.54; Lemmas 6.53 and 6.30; computation of the renaming.

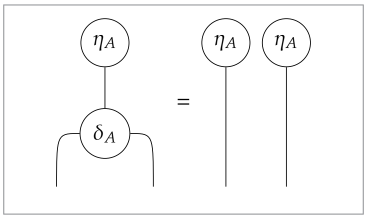
(c) Symetric to *(b)*.

(d) By definition, we have:

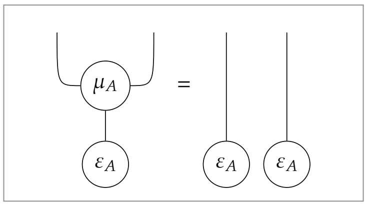
$$\eta_{\mathsf{A}} \odot \mu_{\mathsf{A}} \;=\; 1_{\mathsf{A}\vdash\mathsf{I}} \odot 1_{\mathsf{I}\vdash\mathsf{A}} \;=\; 1_{\mathsf{I}\vdash\mathsf{I}}\,. \qquad \square$$



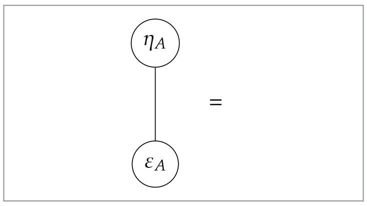
(a) Multiplication and co-multiplication.



(b) Unitor and co-multiplication.



(c) Multiplication and co-unitor.



(d) Unitor and co-unitor.

Figure 7.1: Bialgebra laws.

8.1.3 Proof of the bialgebra distributivity law

Let us look at the exchange rule between δ and μ again. First, we must introduce some additional notation.

Notation: Consider A a negative arena and $x \in \mathsf{Conf}(\mathsf{A})$. We write $x = y \uplus z$ when $y, z \in \mathsf{Conf}(\mathsf{A})$, $|x| = |y| \cup |z|$ and $|y| \cap |z| = \emptyset$.

This is analogous to $x = y * z$ (and entails $x \cong y * z$), but instead of the tagged disjoint union we have the standard set-theoretic union, which happens to be disjoint.

Reminder: Given two configurations $x_1, x_2 \in \mathsf{Conf}(\mathsf{A})$, we set $x_1 * x_2$ with:

- events $|x_1| + |x_2|$,
- display may $\partial((i, a)) = \partial_{x_i}(a)$,
- causal order inherited.

Then $x_1 * x_2 \in \mathsf{Conf}(\mathsf{A})$.

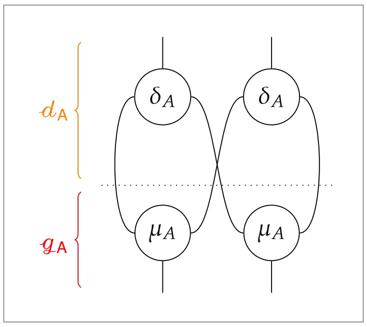


Figure 8.1: Distribute and gather.

Qualitatively. We rephrase the exchange rule: we write

$$d_{\mathsf{A}} \stackrel{\text{def}}{=} (\mathrm{id}_{\mathsf{A}} \otimes \gamma_{\mathsf{A},\mathsf{A}} \otimes \mathrm{id}_{\mathsf{A}}) \odot (\delta_{\mathsf{A}} \otimes \delta_{\mathsf{A}}), \qquad g_{\mathsf{A}} \stackrel{\text{def}}{=} \mu_{\mathsf{A}} \otimes \mu_{\mathsf{A}}$$

for "*distribute*" and "*gather*" (see Figure 8.1) which lets us phrase the desired bialgebra law as $\delta_{\mathsf{A}} \odot \mu_{\mathsf{A}} = g_{\mathsf{A}} \odot d_{\mathsf{A}}$.

Writing $S(\mathsf{x}_1, \ldots, \mathsf{x}_n) \stackrel{\text{def}}{=} \sharp\mathsf{Sym}(\mathsf{x}_1) \times \ldots \times \sharp\mathsf{Sym}(\mathsf{x}_n)$, we have:

$$\begin{aligned} d_{\mathsf{A}} &= \sum_{\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v} \in \mathsf{Pos}(\mathsf{A})} \frac{1}{S(\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v})} \cdot \mathsf{d}_{\mathsf{A}}^{\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v}} \\ g_{\mathsf{A}} &= \sum_{\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v} \in \mathsf{Pos}(\mathsf{A})} \frac{1}{S(\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v})} \cdot \mathsf{g}_{\mathsf{A}}^{\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v}} \end{aligned}$$

where $\mathsf{d}_{\mathsf{A}}^{\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v}}$ and $\mathsf{g}_{\mathsf{A}}^{\mathsf{x},\mathsf{y},\mathsf{u},\mathsf{v}}$ are the isomorphism classes of the augmentations $d_{\mathsf{A}}^{\underline{\mathsf{x}},\underline{\mathsf{y}},\underline{\mathsf{u}},\underline{\mathsf{v}}}$ and $g_{\mathsf{A}}^{\underline{\mathsf{x}},\underline{\mathsf{y}},\underline{\mathsf{u}},\underline{\mathsf{v}}}$ obtained with

$$\begin{aligned} (\!| d_{\mathsf{A}}^{x,y,u,v} |\!) &= (x * y) \otimes (u * v) \vdash (x \otimes u) \otimes (y \otimes v) \\ (\!| g_{\mathsf{A}}^{x,y,u,v} |\!) &= (x \otimes y) \otimes (u \otimes v) \vdash (x * y) \otimes (u * v) \end{aligned}$$

and the obvious copycat behaviour on each component x, y, u, v.

For our proof, the first key observation is the following lemma:

Lemma 8.4 – Qualitative behavior

Consider an arena A, and $\mathsf{x}, \mathsf{y}, \mathsf{x}', \mathsf{y}' \in \mathsf{Pos}(\mathsf{A})$. Then,

$$\delta_{\mathsf{A}}^{\mathsf{x},\mathsf{y}} \odot \mu_{\mathsf{A}}^{\mathsf{x}',\mathsf{y}'} = \sum_{\substack{\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r \text{ s.t.} \\ \mathsf{x}_l * \mathsf{x}_r = \mathsf{x},\ \mathsf{y}_l * \mathsf{y}_r = \mathsf{y} \\ \mathsf{x}_l * \mathsf{y}_l = \mathsf{x}',\ \mathsf{x}_r * \mathsf{y}_r = \mathsf{y}'}} \sum_{\substack{x = x_l \uplus x_r \text{ s.t. } x_l \in \mathsf{x}_l, x_r \in \mathsf{x}_r \\ y = y_l \uplus y_r \text{ s.t. } y_l \in \mathsf{y}_l, y_r \in \mathsf{y}_r \\ x' = x'_l \uplus x'_r \text{ s.t. } x'_l \in \mathsf{x}'_l, x'_r \in \mathsf{x}'_r \\ y' = y'_l \uplus y'_r \text{ s.t. } y'_l \in \mathsf{y}'_l, y'_r \in \mathsf{y}'_r}} \mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \odot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r}.$$

Proof. Consider a symmetry $\varphi : x * y \cong x' * y'$. This symmetry sends some events of x to x', and some others to y' – likewise, it sends some events of y to x', and some to y'. Following these partitions, symmetries $\varphi : x * y \cong x' * y'$ are in bijection with

$$\left\{ \begin{array}{lcl} x &=& x_l \uplus x_r \\ y &=& y_l \uplus y_r \\ x' &=& x'_l \uplus x'_r \\ y' &=& y'_l \uplus y'_r \end{array} \quad , \quad \begin{array}{lcl} \varphi_{l,l} : & x_l & \cong & x'_l \\ \varphi_{l,r} : & x_r & \cong & y'_l \\ \varphi_{r,l} : & y_l & \cong & x'_r \\ \varphi_{r,r} : & y_r & \cong & y'_r \end{array} \right\}.$$

Additionally, this decomposition satisfies

$$\delta_{\mathsf{A}}^{x,y} \odot_{\varphi} \mu_{\mathsf{A}}^{x',y'} = g_{\mathsf{A}}^{x'_l,x'_r,y'_l,y'_r} \odot_{(\varphi_{l,l} \otimes \varphi_{r,l}) \otimes (\varphi_{l,r} \otimes \varphi_{r,r})} d_{\mathsf{A}}^{x_l,x_r,y_l,y_r}$$

which is verified by an immediate analysis of the copycat behaviour of this composition.

We then proceed with, for arbitrary $x \in \mathsf{x}, y \in \mathsf{y}, x' \in \mathsf{x}', y' \in \mathsf{y}'$:

$$
\begin{aligned}
&\delta_{\mathsf{A}}^{\mathsf{x},\mathsf{y}} \odot \mu_{\mathsf{A}}^{\mathsf{x}',\mathsf{y}'} \\
&= \sum_{\varphi :\, x * y \cong x' * y'} \overline{\delta_{\mathsf{A}}^{x,y} \odot_{\varphi} \mu_{\mathsf{A}}^{x',y'}} \\
&= \sum_{\substack{x = x_l \uplus x_r \\ y = y_l \uplus y_r \\ x' = x'_l \uplus x'_r \\ y' = y'_l \uplus y'_r}} \sum_{\substack{\varphi_{l,l} : x_l \cong x'_l \\ \varphi_{l,r} : x_r \cong y'_l \\ \varphi_{r,l} : y_l \cong x'_r \\ \varphi_{r,r} : y_r \cong y'_r}} \underbrace{\overline{g_{\mathsf{A}}^{x'_l, x'_r, y'_l, y'_r} \odot_{(\varphi_{l,l} \otimes \varphi_{r,l}) \otimes (\varphi_{l,r} \otimes \varphi_{r,r})} d_{\mathsf{A}}^{x_l, x_r, y_l, y_r}}}_{\text{noted } \clubsuit \text{ below}} \\
&= \sum_{\substack{\mathsf{x}_l, \mathsf{x}_r, \mathsf{y}_l, \mathsf{y}_r \text{ s.t.} \\ \mathsf{x}_l * \mathsf{x}_r = \mathsf{x},\ \mathsf{y}_l * \mathsf{y}_r = \mathsf{y} \\ \mathsf{x}_l * \mathsf{y}_l = \mathsf{x}',\ \mathsf{x}_r * \mathsf{y}_r = \mathsf{y}'}} \sum_{\substack{x = x_l \uplus x_r \text{ s.t. } x_l \in \mathsf{x}_l, x_r \in \mathsf{x}_r \\ y = y_l \uplus y_r \text{ s.t. } y_l \in \mathsf{y}_l, y_r \in \mathsf{y}_r \\ x' = x'_l \uplus x'_r \text{ s.t. } x'_l \in \mathsf{x}'_l, x'_r \in \mathsf{x}'_r \\ y' = y'_l \uplus y'_r \text{ s.t. } y'_l \in \mathsf{y}'_l, y'_r \in \mathsf{y}'_r}} \sum_{\substack{\varphi_{l,l} : x_l \cong x'_l \\ \varphi_{l,r} : x_r \cong y'_l \\ \varphi_{r,l} : y_l \cong x'_r \\ \varphi_{r,r} : y_r \cong y'_r}} \clubsuit \\
&= \sum_{\substack{\mathsf{x}_l, \mathsf{x}_r, \mathsf{y}_l, \mathsf{y}_r \text{ s.t.} \\ \mathsf{x}_l * \mathsf{x}_r = \mathsf{x},\ \mathsf{y}_l * \mathsf{y}_r = \mathsf{y} \\ \mathsf{x}_l * \mathsf{y}_l = \mathsf{x}',\ \mathsf{x}_r * \mathsf{y}_r = \mathsf{y}'}} \sum_{\substack{x = x_l \uplus x_r \text{ s.t. } x_l \in \mathsf{x}_l, x_r \in \mathsf{x}_r \\ y = y_l \uplus y_r \text{ s.t. } y_l \in \mathsf{y}_l, y_r \in \mathsf{y}_r \\ x' = x'_l \uplus x'_r \text{ s.t. } x'_l \in \mathsf{x}'_l, x'_r \in \mathsf{x}'_r \\ y' = y'_l \uplus y'_r \text{ s.t. } y'_l \in \mathsf{y}'_l, y'_r \in \mathsf{y}'_r}} \mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l, \mathsf{y}_l, \mathsf{x}_r, \mathsf{y}_r} \odot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l, \mathsf{y}_l, \mathsf{x}_r, \mathsf{y}_r} .
\end{aligned}
$$

by the definition of composition of isogmentations (which does not depend on the chosen representative); the observation above; reorganizing the sum by symmetry classes; and again via the definition of composition of isogmentations. □

This is sufficient to ensure that $\delta_{\mathsf{A}} \odot \mu_{\mathsf{A}}$ and $g_{\mathcal{A}} \odot d_{\mathcal{A}}$ have the same isogmentations, but not that they occur with the same coefficient.

Quantitatively. Again, we need to introduce a new notation.

Notation: If $x = y \uplus z$ with $y \in \mathsf{y}$ and $z \in \mathsf{z}$, we write $x \blacktriangleleft \mathsf{y}, \mathsf{z}$.

But there may be several splittings of x into y and z, *i.e.* pairs (y, z) such that $x = y \uplus z$ with $y \in \mathsf{y}$ and $z \in \mathsf{z}$. We write $|x \blacktriangleleft \mathsf{y}, \mathsf{z}|$ the number of such pairs. It is easy to see that this is invariant under symmetry, thus we may write $|\mathsf{x} \blacktriangleleft \mathsf{y}, \mathsf{z}|$ for $|x \blacktriangleleft \mathsf{y}, \mathsf{z}|$ for any $x \in \mathsf{x}$.

Given this definition, Lemma 8.4 rewrites as

Corollary 8.5 – Qualitative behavior with splittings

Consider an arena A, and $\mathsf{x}, \mathsf{y}, \mathsf{x}', \mathsf{y}' \in \mathsf{Pos}(\mathsf{A})$. Then,

$$
\delta_{\mathsf{A}}^{\mathsf{x},\mathsf{y}} \odot \mu_{\mathsf{A}}^{\mathsf{x}',\mathsf{y}'} = \sum_{\substack{\mathsf{x}_l, \mathsf{x}_r, \mathsf{y}_l, \mathsf{y}_r \text{ s.t.} \\ \mathsf{x}_l * \mathsf{x}_r = \mathsf{x},\ \mathsf{y}_l * \mathsf{y}_r = \mathsf{y} \\ \mathsf{x}_l * \mathsf{y}_l = \mathsf{x}',\ \mathsf{x}_r * \mathsf{y}_r = \mathsf{y}'}} s^{\mathsf{x},\mathsf{y}}_{\mathsf{x}_l, \mathsf{x}_r, \mathsf{y}_l, \mathsf{y}_r} \cdot \mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l, \mathsf{y}_l, \mathsf{x}_r, \mathsf{y}_r} \odot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l, \mathsf{y}_l, \mathsf{x}_r, \mathsf{y}_r} ,
$$

where for each $\mathsf{x}_l, \mathsf{x}_r, \mathsf{y}_l, \mathsf{y}_r$ we note:

$$
s^{\mathsf{x},\mathsf{y}}_{\mathsf{x}_l, \mathsf{x}_r, \mathsf{y}_l, \mathsf{y}_r} \stackrel{\text{def}}{=} |\mathsf{x} \blacktriangleleft \mathsf{x}_l, \mathsf{x}_r|\, |\mathsf{y} \blacktriangleleft \mathsf{y}_l, \mathsf{y}_r|\, |\mathsf{x}' \blacktriangleleft \mathsf{x}_l, \mathsf{x}_r|\, |\mathsf{y}' \blacktriangleleft \mathsf{y}_l, \mathsf{y}_r| .
$$

To conclude the proof, the next key observation is:

Lemma 8.6 – Splitting symmetries

Consider an arena A with $\mathsf{x}, \mathsf{y} \in \mathsf{Pos}(\mathsf{A})$. Then,

$$\sharp\mathsf{Sym}\,(\mathsf{x} * \mathsf{y}) = |\mathsf{x} * \mathsf{y} \blacktriangleleft \mathsf{x}, \mathsf{y}| \times \sharp\mathsf{Sym}\,(\mathsf{x}) \times \sharp\mathsf{Sym}\,(\mathsf{y}) \ .$$

Proof. Fix arbitrary $x \in \mathsf{x}$ and $y \in \mathsf{y}$ that we assume disjoint, and $z = x \uplus y \in \mathsf{x} * \mathsf{y}$. The set of symmetries on $\mathsf{x} * \mathsf{y}$ is clearly in bijection with the set of symmetries

$$\varphi : z \cong x * y$$

which we shall study. As in the lemma above, such a symmetry sends some events of z to x and some to y; this induces a splitting $z = x' \uplus y'$ with induced $\varphi_x : x' \cong x$ and $\varphi_y : y' \cong y$, so that $x' \in \mathsf{x}$ and $y' \in \mathsf{y}$. Conversely, any such splitting of z with accompanying symmetries yields a symmetry $z \cong x * y$. From this it is straightforward to obtain a bijection witnessing the announced equality (keeping in mind that we may fix in advance a chosen $\kappa_x : x \cong \underline{\mathsf{x}}$ for all $x \in \mathsf{x}$, so as to bridge between symmetries $x' \cong x$ and endosymmetries $\underline{\mathsf{x}} \cong \underline{\mathsf{x}}$). □

Finally, we prove the exchange law of bialgebras:

Lemma 8.7 – Exchange law

Consider an arena A. Then,

$$\delta_{\mathscr{A}} \odot \mu_{\mathscr{A}} = \mathscr{g}_{\mathscr{A}} \odot \mathscr{d}_{\mathscr{A}} \ .$$

Proof. We have:

$$\begin{aligned}
& \delta_{\mathsf{A}} \odot \mu_{\mathsf{A}} \\
&= \sum_{\substack{\mathsf{x},\mathsf{y} \in \mathsf{Pos}(\mathsf{A}) \\ \mathsf{x}',\mathsf{y}' \in \mathsf{Pos}(\mathsf{A})}} \frac{1}{S(\mathsf{x},\mathsf{y},\mathsf{x}',\mathsf{y}')} \cdot \delta_{\mathsf{A}}^{\mathsf{x},\mathsf{y}} \odot \mu_{\mathsf{A}}^{\mathsf{x}',\mathsf{y}'} \\
&= \sum_{\substack{\mathsf{x},\mathsf{y} \in \mathsf{Pos}(\mathsf{A}) \\ \mathsf{x}',\mathsf{y}' \in \mathsf{Pos}(\mathsf{A})}} \sum_{\substack{\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r \text{ s.t.} \\ \mathsf{x}_l * \mathsf{x}_r = \mathsf{x},\ \mathsf{y}_l * \mathsf{y}_r = \mathsf{y} \\ \mathsf{x}_l * \mathsf{y}_l = \mathsf{x}',\ \mathsf{x}_r * \mathsf{y}_r = \mathsf{y}'}} \frac{s^{\mathsf{x},\mathsf{y}}_{\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r}}{S(\mathsf{x},\mathsf{y},\mathsf{x}',\mathsf{y}')} \cdot \mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \odot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \\
&= \sum_{\substack{\mathsf{x}_l,\mathsf{x}_r \in \mathsf{Pos}(\mathsf{A}) \\ \mathsf{y}_l,\mathsf{y}_r \in \mathsf{Pos}(\mathsf{A})}} \frac{s^{\mathsf{x},\mathsf{y}}_{\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r}}{S(\mathsf{x}_l * \mathsf{x}_r, \mathsf{y}_l * \mathsf{y}_r, \mathsf{x}_l * \mathsf{y}_l, \mathsf{x}_r * \mathsf{y}_l)} \cdot \mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \odot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \\
&= \sum_{\substack{\mathsf{x}_l,\mathsf{x}_r \in \mathsf{Pos}(\mathsf{A}) \\ \mathsf{y}_l,\mathsf{y}_r \in \mathsf{Pos}(\mathsf{A})}} \frac{1}{S(\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r)^2} \cdot \mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \odot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \\
&= \left(\sum_{\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r \in \mathsf{Pos}(\mathsf{A})} \frac{1}{S(\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r)} \cdot \mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \right) \\
&\qquad \odot \left(\sum_{\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r \in \mathsf{Pos}(\mathsf{A})} \frac{1}{S(\mathsf{x}_l,\mathsf{x}_r,\mathsf{y}_l,\mathsf{y}_r)} \cdot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \right) \\
&= \mathscr{g}_{\mathscr{A}} \odot \mathscr{d}_{\mathscr{A}}
\end{aligned}$$

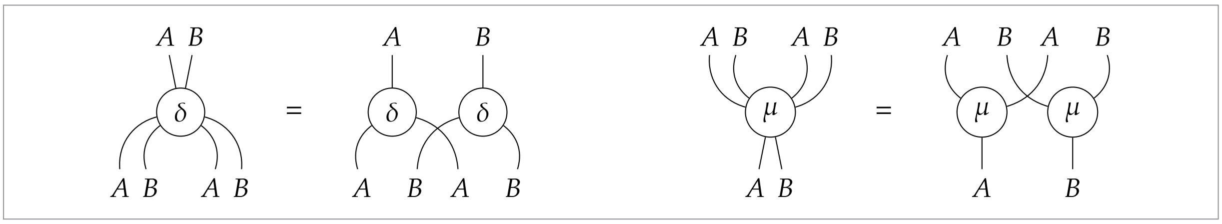


Figure 7.3: Compatibility of (co)monoids with the monoidal structure

by unfolding the definition of δ_{A} and μ_{A}; then using Corollary 8.5; then reindexing the sum in the obvious way; then applying Lemma 8.6; and finally by linearity, observing that any composition $\mathsf{g}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r} \odot \mathsf{d}_{\mathsf{A}}^{\mathsf{x}_l,\mathsf{y}_l,\mathsf{x}_r,\mathsf{y}_r}$ where one of $\mathsf{x}_l = \mathsf{x}'_l, \mathsf{y}_l = \mathsf{x}'_r, \mathsf{x}_r = \mathsf{y}'_l$ and $\mathsf{y}_r = \mathsf{y}'_r$ does not hold is null. □

8.1.4 Compatibility

Finally, we prove that the bialgebra structure is compatible with the monoidal structure of PCG.

Lemma 8.8 – Compatibility

For any arena A, the bialgebra structure $(\delta_{\mathsf{A}}, \varepsilon_{\mathsf{A}}, \mu_{\mathsf{A}}, \eta_{\mathsf{A}})$ is compatible with the monoidal structure of PCG, in the sense that the morphisms satisfy:

co-unitor with tensor:	$\varepsilon_{A\otimes B} = \lambda_I \odot (\varepsilon_A \otimes \varepsilon_B)$
unitor with tensor:	$\eta_{A\otimes B} = (\eta_A \otimes \eta_B) \odot \lambda_I$
(co-)unitors with unit:	$\varepsilon_I = \eta_I = \mathsf{id}_I$

and the equations of Figure 7.3.

Proof. ***(co)-unitor with tensor.*** Both cases are symmetric; clear by definition and computation of the composition.

(co)-unitor with unit. By definition, we have $\varepsilon_{\mathsf{I}} = 1_{\mathsf{I}\vdash\mathsf{I}} = \mathsf{id}_{\mathsf{I}} = \eta_{\mathsf{I}}$.

(co)-multiplication and tensor. Both cases are symmetric; clear using lemmas on renaming. □

8.1.5 Pointed identities

We define the pointed identities of PCG.

Reminder: For any arena A, we note $\mathsf{Conf}_\bullet(\mathsf{A})$ the *pointed*, or *well-opened*, configurations on A – that is, the configurations with a unique minimal event. Likewise, we note $\mathsf{Pos}_\bullet(\mathsf{A})$ for the *pointed* positions on A, *i.e.* the isomorphism classes of pointed configurations.

Definition 8.9 – Pointed identities in PCG

For any arena A, we define the **pointed identity** $\mathsf{id}^\bullet_{\mathsf{A}} : \mathsf{A} \vdash \mathsf{A}$ as:

$$\mathsf{id}^\bullet_{\mathsf{A}} \stackrel{\text{def}}{=} \sum_{\mathsf{x}\,\in\,\mathsf{Pos}_\bullet(\mathsf{A})} \frac{1}{\sharp\mathsf{Sym}\,(\mathsf{x})} \cdot \text{æ}_{\mathsf{x}} \,.$$

Reminder: For any arena A, we note $\mathsf{Aug}_\bullet(\mathsf{A})$ the *pointed*, or *well-opened*, augmentations on A – that is, the augmentations with a unique minimal event. Likewise, we note $\mathsf{Isog}_\bullet(\mathsf{A})$ for the *pointed* isogmentations on A, *i.e.* the isomorphism classes of pointed augmentations.

Before checking that $\mathsf{id}^\bullet_\mathsf{A}$ is a pointed identity in PCG, we give an alternate characterisation, using the restriction of strategies to their pointed isogmentations.

Definition 8.10 – Trimmed strategies

Consider arenas A, B and a strategy $\sigma : \mathsf{A} \vdash \mathsf{B}$. The restriction of σ to well-opened isogmentations, or **trimming** of σ, is the strategy:

$$\sigma_\bullet \stackrel{\text{def}}{=} \sum_{\mathsf{q} \in \mathsf{Isog}_\bullet(\mathsf{A}\vdash\mathsf{B})} \sigma(\mathsf{q}) \cdot \mathsf{q} .$$

Then it is clear that the pointed identity on A *is* the trimming of the identity on A.

Lemma 8.11 – Alternate characterisation of $\mathsf{id}^\bullet_\mathsf{A}$

For any arena A, we have $\mathsf{id}^\bullet_\mathsf{A} = (\mathsf{id}_\mathsf{A})_\bullet$.

More generally, we show that *pointed morphisms* (in the categorical sense) are *pointed strategies* (in the sense that all the isogmentations in their support are pointed).

Lemma 8.12

Consider arenas A, B and a strategy $\sigma : \mathsf{A} \vdash \mathsf{B}$. Then:

$$\mathsf{id}^\bullet_\mathsf{B} \odot \sigma = \sigma_\bullet .$$

Let us check that $\mathsf{id}^\bullet_\mathsf{A}$ satisfies the axioms of a pointed identity.

Reminder: A pointed identity on A is a endomorphism on A which is

- idempotent,
- non-erasable: $\varepsilon_A \circ \mathsf{id}^\bullet_A = 0$,
- non-erasing: $\mathsf{id}^\bullet_A \circ \eta_A = 0$,
- non-duplicable:
$$\delta_A \circ \mathsf{id}^\bullet_A = (\mathsf{id}^\bullet_A \otimes \eta_A) + (\eta_A \otimes \mathsf{id}^\bullet_A) ,$$
- non-duplicative:
$$\mathsf{id}^\bullet_A \circ \mu_A = (\mathsf{id}^\bullet_A \otimes \varepsilon_A) + (\varepsilon_A \otimes \mathsf{id}^\bullet_A) .$$

Lemma 8.13

For any arena A, $\mathsf{id}^\bullet_\mathsf{A}$ is a pointed identity.

Proof. ***Idempotence.*** Direct by Lemmas 8.12 and 8.11.

Non-erasable. By definition of ε_A:

$$\varepsilon_\mathsf{A} \odot \mathsf{id}^\bullet_\mathsf{A} = 1_{\mathsf{A}\vdash\mathsf{I}} \odot \mathsf{id}^\bullet_\mathsf{A} .$$

But the only isogmentation in $1_{\mathsf{A}\vdash\mathsf{I}}$ is 0 the empty isogmentation, and the configuration x^0_A is not pointed, hence $1_{\mathsf{A}\vdash\mathsf{I}} \odot \mathsf{id}^\bullet_\mathsf{A} = 0_{\mathsf{A}\vdash\mathsf{I}}$.

Non-erasing. We have:

$$\mathsf{id}^\bullet_\mathsf{A} \odot \eta_\mathsf{A} = \mathsf{id}^\bullet_\mathsf{A} \odot 1_{\mathsf{I}\vdash\mathsf{A}} = (1_{\mathsf{I}\vdash\mathsf{A}})_\bullet = 0_{\mathsf{I}\vdash\mathsf{A}}$$

by definition of η_A and Lemma 8.12.

Non-duplicable. By definition of δ_A:

$$\delta_\mathsf{A} \odot \mathsf{id}^\bullet_\mathsf{A} = (\mathsf{id}_{\mathsf{A}\otimes\mathsf{A}} \ltimes \mathsf{c}_\mathsf{A}) \odot \mathsf{id}^\bullet_\mathsf{A} .$$

Any isogmentation $\mathsf{q} \in \mathsf{supp}(\delta_\mathsf{A})$ is of the form $(\mathsf{q}_1 \otimes \mathsf{q}_2) \ltimes \mathsf{c}_\mathsf{A}$, where both q_i's are copycats. So the left configuration x_A^q is pointed iff q is of the form $(q_1 \otimes 0) \ltimes c_A$ with q_1 pointed, or $(0 \otimes q_2) \ltimes c_A$ with q_2 pointed. Computing the composition, we obtain:

$$(id_{A \otimes A} \ltimes c_A) \odot id_A^\bullet = \left(id_A^\bullet \otimes \eta_A\right) + \left(\eta_A \otimes id_A^\bullet\right) \ .$$

Non-duplicative. We have:

$$id_A^\bullet \odot \mu_A = id_A^\bullet \odot (c_A \rtimes id_{A \otimes A}) = (c_A \rtimes id_{A \otimes A})_\bullet$$

by definition of μ_A and Lemma 8.12. But any $q \in supp(\mu_A)$ is of the form $c_A \rtimes (q_1 \otimes q_2)$, so pointed isogmentations of μ_A are either of the form $c_A \rtimes (q_1 \otimes 0)$ with q_1 pointed, or of the form $c_A \rtimes (0 \otimes q_2)$ with q_2 pointed. Computing the composition, we obtain:

$$(c_A \rtimes id_{A \otimes A})_\bullet = \left(id_A^\bullet \otimes \varepsilon_A\right) + \left(\varepsilon_A \otimes id_A^\bullet\right) \ . \qquad \square$$

8.1.6 Closed structure

We already know that PCG is a SMCC from Theorem 6.63. We need to check that Λ is compatible with the pointed identity as required by Definition 7.6.

First, remark that since Λ preserves well-openedness for isogmentations, it is immediate that it also preserves trimming:

$$\text{for any } \sigma : G \otimes A \vdash B, \quad \Lambda_{G,A,B}\,(\sigma_\bullet) = (\Lambda_{G,A,B}(\sigma))_\bullet \ .$$

From this we easily deduce the next lemma.

Lemma 8.14 – Compatibility of Λ with $id^\bullet$ in PCG

Consider two arenas A, B. We have:

$$id_{A \Rightarrow B}^\bullet \quad = \quad \Lambda_{A \Rightarrow B, A, B}\left(id_B^\bullet \odot ev_{A,B}\right) \ .$$

Proof. For any A, B, we have

$$\begin{aligned} & \Lambda_{A \Rightarrow B, A, B}\left(id_B^\bullet \odot ev_{A,B}\right) \\ = \ & \Lambda_{A \Rightarrow B, A, B}\left((ev_{A,B})_\bullet\right) && \text{(Lemma 8.12)} \\ = \ & (\Lambda_{A \Rightarrow B, A, B}\,(ev_{A,B}))_\bullet && (\Lambda \text{ preserves trimming}) \\ = \ & (id_{A \Rightarrow B})_\bullet && \text{(Definition of ev)} \\ = \ & id_{A \Rightarrow B}^\bullet && \text{(Lemma 8.11)} \end{aligned} \qquad \square$$

Hence PCG has the desired structure, and we conclude:

Theorem 8.15 – Closed structure

PCG is a closed resource category.

$$
\begin{aligned}
[\![\Gamma \vdash_{\mathsf{Tm}} \lambda x.s : A \multimap B]\!] &\stackrel{\text{def}}{=} \Lambda_{[\![\Gamma]\!],[\![A]\!],[\![B]\!]}([\![\Gamma, x : A \vdash_{\mathsf{Tm}} s : B]\!] \circ \mathbin{/\!\!\!\backslash\!\!\!\backslash}_{[\![\Gamma]\!],[\![x:A]\!]}) \\
[\![\Gamma \vdash_{\mathsf{Tm}} x\,\vec{t} : \alpha]\!] &\stackrel{\text{def}}{=} \mathrm{ev}_{[\![\vec{A}]\!],[\![\alpha]\!]} \circ \langle \mathsf{id}^{\bullet}_{[\![\vec{A}]\!]\Rightarrow o} \circ \mathrm{var}^{\Gamma}_{x}, \langle\!|[\![\Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{A}]\!]|\!\rangle\rangle \\
[\![\Gamma \vdash_{\mathsf{Tm}} s\,\bar{t} : B]\!] &\stackrel{\text{def}}{=} \mathrm{ev}_{[\![A]\!],[\![B]\!]} \circ \langle [\![\Gamma \vdash_{\mathsf{Tm}} s : A \multimap B]\!], \Pi[\![\Gamma \vdash_{\mathsf{Bg}} \bar{t} : A]\!]\rangle \\
[\![\Gamma \vdash_{\mathsf{Bg}} [s_1, \dots, s_n] : A]\!] &\stackrel{\text{def}}{=} [\, [\![\Gamma \vdash_{\mathsf{Tm}} s_i : A]\!] \mid 1 \le i \le n \,] \\
[\![\Gamma \vdash_{\mathsf{Sq}} \langle \bar{s}_1, \dots, \bar{s}_n\rangle : \vec{A}]\!] &\stackrel{\text{def}}{=} \langle\, [\![\Gamma \vdash_{\mathsf{Bg}} \bar{s}_i : A_i]\!] \mid 1 \le i \le n \,\rangle
\end{aligned}
$$

Figure 7.7: Interpretation of the resource calculus

$$
\begin{aligned}
\|\Gamma \vdash_{\mathsf{Tm}} \lambda x.s : A \multimap B\|_{\mathsf{Tm}} &\stackrel{\text{def}}{=} \Lambda^{\mathsf{Isog}\bullet}_{[\![\Gamma]\!],[\![A]\!],[\![B]\!]}(\|\Gamma, x : A \vdash_{\mathsf{Tm}} s : B\|_{\mathsf{Tm}}) \\
\|\Gamma \vdash_{\mathsf{Tm}} x\,\vec{t} : \alpha\|_{\mathsf{Tm}} &\stackrel{\text{def}}{=} \Box_i(\|\Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{A}\|_{\mathsf{Sq}}) \\
\|\Gamma \vdash_{\mathsf{Bg}} [s_1, \dots, s_n] : A\|_{\mathsf{Bg}} &\stackrel{\text{def}}{=} \Pi_{\mathsf{Isog}}[\, \|\Gamma \vdash_{\mathsf{Tm}} s_i : A\|_{\mathsf{Tm}} \mid 1 \le i \le n \,] \\
\|\Gamma \vdash_{\mathsf{Sq}} \langle \bar{s}_1, \dots, \bar{s}_n\rangle : \vec{A}\|_{\mathsf{Sq}} &\stackrel{\text{def}}{=} \langle\, \|\Gamma \vdash_{\mathsf{Bg}} \bar{s}_i : A_i\|_{\mathsf{Bg}} \mid 1 \le i \le n \,\rangle_{\mathsf{Isog}}
\end{aligned}
$$

Figure 5.7: Isomorphism for normal forms of the resource calculus

8.2 Compatibility with normal forms

Finally, we show that, up to the bijection $\|-\|_{\mathsf{Tm}}$ between normal resource terms and isogmentations, the interpretation of a resource term in the resource category PCG coincides with its normal form.

Proposition 8.16 – Compatibility with normal forms

Consider $s \in \mathsf{Tm}_{\mathsf{nf}}(\Gamma; A)$. Then $[\![s]\!]$ is the sum having $\|s\|_{\mathsf{Tm}}$ with coefficient 1, and 0 everywhere else.

Proof. Recall the interpretation of the resource calculus in a resource category given in Figure 7.7. Restricting to normal forms rules out the third clause. We inductively prove the remaining cases:

- if $s \in \mathsf{Tm}_{\mathsf{nf}}(\Gamma; A)$, then $[\![s]\!] = \|s\|_{\mathsf{Tm}}$;
- if $\bar{s} \in \mathsf{Bg}_{\mathsf{nf}}(\Gamma; A)$, then $\Pi[\![\bar{s}]\!] = \|\bar{s}\|_{\mathsf{Bg}}$;
- if $\vec{s} \in \mathsf{Sq}_{\mathsf{nf}}(\Gamma; \vec{A})$, then $\langle\!|\, [\![\vec{s}]\!] \,|\!\rangle = \|\vec{s}\|_{\mathsf{Sq}}$.

The identity follows immediately from the induction hypothesis for sequences, bags and abstraction terms, since the interpretation matches the bijection for normal forms described in Figure 5.7.

The case of a fully applied variable is less obvious. Recall the construction of the i-lifting, as presented in Figure 5.6.

We want this construction to match:

$$\mathrm{ev}_{[\![\vec{B}]\!],o} \circ \langle \mathsf{id}^{\bullet}_{[\![\vec{B}]\!]\Rightarrow o} \circ \mathrm{var}^{\Gamma}_{x}, \langle\!|[\![\Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{B}]\!]|\!\rangle\rangle\,.$$

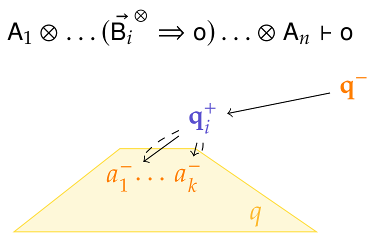


Figure 5.6: $\Box_i(q)$.

Let us look at the behavior of $\mathrm{ev}_{[\![\vec{B}]\!],[\![\alpha]\!]}$ in PCG. We know that:

$$\mathrm{ev}_{[\![\vec{B}]\!],o} = \Lambda^{-1}_{[\![\vec{B}]\!]\Rightarrow o,[\![\vec{B}]\!],o}\left(\mathrm{id}_{[\![\vec{B}]\!]\Rightarrow o}\right) .$$

Thus the isogmentations in $\mathrm{ev}_{[\![\vec{B}]\!],o}$ all look as in Figure 8.2, where the left and right sides are copies of the same position, with causal links given by copycat as usual. When composing $\mathrm{ev}_{[\![\vec{B}]\!],o}$ with

$$\langle \mathrm{id}^{\bullet}_{[\![\vec{B}]\!]\Rightarrow o} \circ \mathrm{var}^{\Gamma}_{x}, (\![[\![\Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{B}]\!]]\!) \rangle ,$$

the right side will interact with $(\![[\![\Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{B}]\!]]\!)$ – which, by induction hypothesis, is $\|\Gamma \vdash_{\mathsf{Sq}} \vec{t} : \vec{B}\|_{\mathsf{Sq}}$. After the hiding, we are left with exactly the i-lifting construction. □

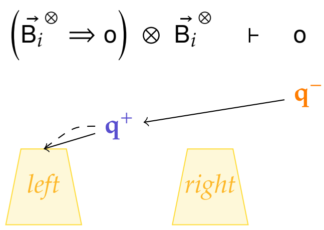


Figure 8.2: An isogmentation in $\mathrm{ev}_{[\![\vec{A}]\!],o}$.

It immediately follows that our interpretation in PCG computes a representation of the normal form:

Theorem 8.17 – Interpretation and normal form

If $s \in \mathsf{Tm}(\Gamma; A)$ has normal form $\sum_{i\in I} s_i$, then $[\![s]\!] = \sum_{i\in I} \|s_i\|_{\mathsf{Tm}}$.

8.3 Conclusion

Thanks to the interpretation of resource terms from Chapter 7, we only had to check that PCG is indeed a closed resource category to obtain the interpretation in PCG. Moreover, this interpretation is not disconnected from our previous construction linking PCG and the resource calculus! Although we define the interpretation of a term s with a purely categorical construction, we show that this interpretation *is* the sum of isogmentations obtained *via* the isomorphism between normal terms and PCG from the normal form of s.

Conclusion

Conclusion

We presented the construction of *Pointer Concurrent Games*, as well as results regarding its connections with HO games, the relational model, and the resource calculus.

PCG and HO games. Augmentations in PCG were constructed to match plays of HO quotiented by homotopy. There is a bijection:

$$\mathbf{Plays}(-)\colon \mathsf{Isog}(\mathsf{A}) \cong \mathsf{VisPlays}^{+}(\mathsf{A})_{/\sim_E} \qquad \text{[Theorem 3.27].}$$

We showed that the equivalent of innocent strategies in HO is either:

- $-$*-linear isogmentations*, for meagre strategies [Theorem 3.40];
- or *isoexpansions* of $-$-linear isogmentations, for fat strategies.

Finally, we showed that this static correspondence extends to the categorical structure of PCG: $\mathsf{Plays}^{\Rightarrow}(-)$ is a strict cartesian closed functor between FII and $\mathsf{HO}_f^{\mathsf{Inn}}$ [Theorem 6.81].

PCG and $\mathsf{Rel}_!$. We showed that meagre innocent isogmentations in PCG are *positionally injective* [Theorem 4.31]. This result translates to total finite innocent strategies in HO [Theorem 4.32].

PCG and the resource calculus. We have a direct isomorphism between normal resource terms and isogmentations [Theorem 5.18]. Thanks to the interpretation of resource terms in a closed resource category [Theorem 7.20], we have a sound interpretation of resource terms in PCG. Moreover, the interpretation of a resource term is the sum of the isogmentations obtained from its normal form *via* the isomorphism for normal terms [Theorem 8.17], which gives us the diagram from Figure 9.

$$\begin{array}{ccc} \text{resource term } s & \longrightarrow & \mathsf{NF}(s) = \sum s'_i \\ \downarrow & & \downarrow \\ [\![s]\!] & = & \sum \|s'_i\| \end{array}$$

Figure 9: The interpretation behaves nicely!

Perspectives: ongoing and future works.

Taylor expansion. The diagram from Figure 9 is the first step in studying the links between the Taylor expansion of λ-terms and game semantics.

Given a λ-term, its *Taylor expansion* is the sum of its approximations as resource terms. We want to extend the results from Figure 9 to show:

$$\begin{array}{ccccc} M & \longrightarrow & \mathcal{T}(M) & \longrightarrow & \mathsf{NF}(\mathcal{T}(M)) \\ \downarrow & (1) & \downarrow & (2) & \downarrow \\ [\![M]\!] & = & [\![\mathcal{T}(M)]\!] & = & \|\mathsf{NF}(\mathcal{T}(M))\| \end{array}$$

where M is a λ-term, $\mathcal{T}(M)$ is its Taylor expansion, and $\mathsf{NF}(-)$ is the normalisation.

We need to define $\mathcal{T}(-)$ as a Taylor expansion sending simply-typed λ-terms to terms of the simply-typed, η-long resource calculus – then (2) is obtained from Figure 9.

To show that (1) commutes, we also need to describe the interpretation of λ-terms in PCG. Given a closed resource category, how do we construct a *cartesian closed category* – which is the categorical structure usually needed for the target of the interpretation of λ-calculus? This is the subject of ongoing work.

Untyped calculi. For now, we focused on *typed* λ-calculus. Indeed, augmentations and strategies live in *arenas*, so in order to interpret untyped λ-terms, we need a way to "type" them. This alone is not an obstacle: following [29], untyped λ-terms can be interpreted in HO games as strategies on a *universal arena*.

[29]: Ker, Nickau, and Ong (2002), 'Innocent game models of untyped lambda-calculus'

However, the correspondence between *resource* terms and game semantics requires the resource terms to be η-expanded – but what would it mean for an untyped term to be η-expanded? Although this document only presents results in the typed setting, we are – at the date of writing – working on an untyped *extensional resource calculus*. The details are presented in [6] (unpublished yet); we give but a brief overview here.

[6]: Blondeau-Patissier, Clairambault, and Vaux Auclair (2025), *Extensional Taylor Expansion*

In the syntax of the extensional resource calculus, we allow for infinite sequences of abstractions, and for applications to infinite sequences of (almost always empty) bags. Intuitively, we replace η-longness in the typed setting with infinite η-expansion in the untyped setting. The resulting terms are called *extensional*. We define $\mathcal{T}_{\mathsf{ext}}(-)$ the extensional Taylor expansion, sending an untyped λ-term to a linear combination of extensional resource terms.

In the usual λ-calculus, the structure of a λ-term is captured by its *Böhm tree* (see [2]). The normal form of the Taylor expansion of a term M is the Taylor expansion of its *Böhm tree*:

[2]: Barendregt (1984), *The lambda calculus - its syntax and semantics*

$$\mathsf{NF}(\mathcal{T}(M)) = \mathcal{T}(\mathcal{B}(M))$$

as proved first in [21, Corollary 1] or in a more direct way in [41].

[21]: Ehrhard and Regnier (2006), 'Böhm Trees, Krivine's Machine and the Taylor Expansion of Lambda-Terms'

Nakajima trees (defined in [34]) correspond to Böhm trees up to infinite η-expansion [2, Exercise 19.4.4]. The extensional Taylor expansion has the same link with Nakajima trees as the usual Taylor expansion with Böhm trees:

$$\mathsf{NF}(\mathcal{T}_{\mathsf{ext}}(M)) = \mathcal{T}_{\mathsf{ext}}(\mathcal{N}(M))$$

[41]: Vaux (2019), 'Normalizing the Taylor expansion of non-deterministic λ-terms, via parallel reduction of resource vectors'

[34]: Nakajima (1975), 'Infinite normal forms for the lambda - calculus'

[2]: Barendregt (1984), *The lambda calculus - its syntax and semantics*

This extensional resource calculus allows us to extend the connections between Taylor expansions and game semantics made in a typed setting to the untyped setting.

APPENDICES

Bibliography

Here are the references in alphabetical order.

[1] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. 'Full Abstraction for PCF'. In: *Inf. Comput.* 163.2 (2000), pp. 409–470. DOI: 10.1006/inco.2000.2930 (cited on page 2).

[2] Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*. Vol. 103. Studies in logic and the foundations of mathematics. North-Holland, 1984 (cited on pages 3, 16, 187).

[3] Lison Blondeau-Patissier. 'Resource Categories from Differential Categories'. In: *35es Journées Francophones des Langages Applicatifs (JFLA 2024)*. Saint-Jacut-de-la-Mer, France, Jan. 2024 (cited on page 90).

[4] Lison Blondeau-Patissier and Pierre Clairambault. 'Positional Injectivity for Innocent Strategies'. In: *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*. Ed. by Naoki Kobayashi. Vol. 195. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 17:1–17:22. DOI: 10.4230/LIPIcs.FSCD.2021.17 (cited on page 38).

[5] Lison Blondeau-Patissier, Pierre Clairambault, and Lionel Vaux Auclair. 'Strategies as Resource Terms, and Their Categorical Semantics'. In: *8th International Conference on Formal Structures for Computation and Deduction (FSCD 2023)*. Ed. by Marco Gaboardi and Femke van Raamsdonk. Vol. 260. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 13:1–13:22. DOI: 10.4230/LIPIcs.FSCD.2023.13 (cited on page 90).

[6] Lison Blondeau-Patissier, Pierre Clairambault, and Lionel Vaux Auclair. *Extensional Taylor Expansion*. 2025. URL: https://arxiv.org/abs/2305.08489 (cited on page 187).

[7] Richard Blute, J. Robin B. Cockett, and Robert A. G. Seely. 'Differential categories'. In: *Mathematical Structures in Computer Science* 16 (2006), pp. 1049–1083 (cited on pages 8, 15, 147, 165, 167, 168, 171).

[8] Richard Blute et al. 'Differential Categories Revisited'. In: *Appl. Categorical Struct.* 28.2 (2020), pp. 171–235. DOI: 10.1007/s10485-019-09572-y (cited on pages 8, 165–169, 171).

[9] Pierre Boudes. 'Thick Subtrees, Games and Experiments'. In: *TLCA*. Vol. 5608. Lecture Notes in Computer Science. Springer, 2009, pp. 65–79 (cited on pages 4, 41, 42).

[10] Gérard Boudol. 'The lambda-calculus with multiplicities'. In: *CONCUR'93*. Ed. by Eike Best. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 1–6 (cited on page 3).

[11] Gérard Boudol, Pierre-Louis Curien, and Carolina Lavatelli. 'A semantics for lambda calculi with resources'. In: *Mathematical Structures in Computer Science* 9 (1999), pp. 437–482 (cited on page 3).

[12] Simon Castellan and Pierre Clairambault. *Disentangling Parallelism and Interference in Game Semantics*. 2021. URL: https://arxiv.org/abs/2103.15453 (cited on page 105).

[13] Simon Castellan et al. 'The concurrent game semantics of Probabilistic PCF'. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*. Ed. by Anuj Dawar and Erich Grädel. ACM, 2018, pp. 215–224. DOI: 10.1145/3209108.3209187 (cited on page 42).

[14] Alonzo Church. 'A Formulation of the Simple Theory of Types'. In: *The Journal of Symbolic Logic* 5.2 (1940), pp. 56–68. (Visited on 07/29/2025) (cited on page 3).

[15] Pierre Clairambault. 'Causal Investigations in Interactive Semantics'. Habilitation à diriger des recherches. Aix-Marseille Université, 2024 (cited on page 6).

[16] Pierre Clairambault and Marc de Visme. 'Full abstraction for the quantum lambda-calculus'. In: *Proc. ACM Program. Lang.* 4.POPL (2020), 63:1–63:28. DOI: 10.1145/3371131 (cited on page 42).

[17] V. Danos, H. Herbelin, and L. Regnier. 'Game semantics and abstract machines'. In: *Proceedings 11th Annual IEEE Symposium on Logic in Computer Science*. 1996, pp. 394–405. DOI: 10.1109/LICS.1996.561456 (cited on page 31).

[18] Daniel de Carvalho. 'The Relational Model Is Injective for Multiplicative Exponential Linear Logic'. In: *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*. 2016, 41:1–41:19. DOI: 10.4230/LIPIcs.CSL.2016.41 (cited on pages 5, 6, 65).

[19] Thomas Ehrhard. 'The Scott model of linear logic is the extensional collapse of its relational model'. In: *Theor. Comput. Sci.* 424 (2012), pp. 20–45. DOI: 10.1016/j.tcs.2011.11.027 (cited on page 41).

[20] Thomas Ehrhard and Laurent Regnier. 'The differential lambda-calculus'. In: *Theoretical Computer Science* 309.1 (2003), pp. 1–41. DOI: https://doi.org/10.1016/S0304-3975(03)00392-X (cited on pages 8, 165).

[21] Thomas Ehrhard and Laurent Regnier. 'Böhm Trees, Krivine's Machine and the Taylor Expansion of Lambda-Terms'. In: *Logical Approaches to Computational Barriers*. Ed. by Arnold Beckmann et al. Springer Berlin Heidelberg, 2006, pp. 186–197 (cited on page 187).

[22] Thomas Ehrhard and Laurent Regnier. 'Uniformity and the Taylor expansion of ordinary lambda-terms'. In: *Theoretical Computer Science* 403.2-3 (2008), pp. 347–372. DOI: 10.1016/j.tcs.2008.06.001 (cited on pages 8, 19, 21).

[23] Marcelo P. Fiore. 'Differential Structure in Models of Multiplicative Biadditive Intuitionistic Linear Logic'. In: *Typed Lambda Calculi and Applications*. Ed. by Simona Ronchi Della Rocca. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 163–177 (cited on page 167).

[24] Jean-Yves Girard. 'Linear logic'. In: *Theoretical Computer Science* 50.1 (1987), pp. 1–101. DOI: https://doi.org/10.1016/0304-3975(87)90045-4 (cited on pages 3, 166).

[25] Jean-Yves Girard. 'Normal functors, power series and λ-calculus'. In: *Ann. Pure Appl. Log.* 37 (1988), pp. 129–177 (cited on pages 3, 4).

[26] Russ Harmer. *Innocent game semantics*. Lecture notes. 2006. URL: https://perso.ens-lyon.fr/russell.harmer/GS.pdf (cited on page 33).

[27] J. M. E. Hyland and C.-H. Luke Ong. 'On Full Abstraction for PCF: I, II, and III'. In: *Inf. Comput.* 163.2 (2000), pp. 285–408. DOI: 10.1006/inco.2000.2917 (cited on pages 2, 23, 33).

[28] André Joyal and Ross Street. 'The geometry of tensor calculus, I'. In: *Advances in Mathematics* 88 (1991), pp. 55–112 (cited on page 14).

[29] Andrew D. Ker, Hanno Nickau, and C.-H. Luke Ong. 'Innocent game models of untyped lambda-calculus'. In: *Theor. Comput. Sci.* 272.1-2 (2002), pp. 247–292. DOI: 10.1016/S0304-3975(00)00353-4 (cited on page 187).

[30] David P. Kierstead. 'A Semantics for Kleene's j-expressions'. In: *The Kleene Symposium*. Ed. by Jon Barwise, H. Jerome Keisler, and Kenneth Kunen. Vol. 101. Studies in Logic and the Foundations of Mathematics. Elsevier, 1980, pp. 353–366. DOI: https://doi.org/10.1016/S0049-237X(08)71268-0 (cited on page 44).

[31] Saunders Mac Lane. 'Natural Associativity and Commutativity'. In: *Rice Institute Pamphlet - Rice University Studies* 49 (1963), pp. 28–46 (cited on page 14).

[32] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics, Vol. 5. New York: Springer-Verlag, 1971, pp. ix+262 (cited on pages 13, 14, 147).

[33] Paul-André Melliès. 'Asynchronous games 2: The true concurrency of innocence'. In: *Theor. Comput. Sci.* 358.2-3 (2006), pp. 200–228. DOI: 10.1016/j.tcs.2006.01.016 (cited on pages 5, 35, 42, 43, 49).

[34] Reiji Nakajima. 'Infinite normal forms for the lambda - calculus'. In: *Lambda-Calculus and Computer Science Theory, Proceedings of the Symposium Held in Rome, Italy, March 25-27, 1975*. Ed. by Corrado Böhm. Vol. 37. Lecture Notes in Computer Science. Springer, 1975, pp. 62–82. DOI: 10.1007/BFb0029519 (cited on page 187).

[35] Hanno Nickau. 'Hereditarily Sequential Functionals'. In: *Logical Foundations of Computer Science, Third International Symposium, LFCS'94, St. Petersburg, Russia, July 11-14, 1994, Proceedings*. Ed. by Anil Nerode and Yuri V. Matiyasevich. Vol. 813. Lecture Notes in Computer Science. Springer, 1994, pp. 253–264. DOI: 10.1007/3-540-58140-5_25 (cited on page 2).

[36] Gordon D. Plotkin. 'LCF Considered as a Programming Language'. In: *Theoretical Computer Science* 5.3 (1977), pp. 223–255. DOI: 10.1016/0304-3975(77)90044-5 (cited on page 2).

[37] Dana S. Scott. 'A Type-Theoretical Alternative to ISWIM, CUCH, OWHY'. In: *Theoretical Computer Science* 121.1&2 (1993), pp. 411–440. DOI: 10.1016/0304-3975(93)90095-B (cited on page 2).

[38] Robert A. G. Seely. 'Linear Logic, $*$-Autonomous Categories and Cofree Coalgebras'. In: 1989 (cited on page 168).

[39] Peter Selinger. 'A Survey of Graphical Languages for Monoidal Categories'. In: *New Structures for Physics*. Springer Berlin Heidelberg, 2010, pp. 289–355. DOI: 10.1007/978-3-642-12821-9_4 (cited on page 14).

[40] Takeshi Tsukada and C.-H. Luke Ong. 'Plays as Resource Terms via Non-idempotent Intersection Types'. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*. Ed. by Martin Grohe, Eric Koskinen, and Natarajan Shankar. ACM, 2016, pp. 237–246. DOI: 10.1145/2933575.2934553 (cited on pages 5, 6, 21, 35, 44, 91).

[41] Lionel Vaux. 'Normalizing the Taylor expansion of non-deterministic λ-terms, via parallel reduction of resource vectors'. In: *Logical Methods in Computer Science* Volume 15, Issue 3 (2019). DOI: 10.23638/lmcs-15(3:9)2019 (cited on page 187).

Alphabetical Index

Nomenclature

Here we present several symbols that are used within the body of the document.

Categories

$\mathscr{C}, \mathscr{D}$ Categories

f, g, h Morphisms

A, B, C Objects

Calculus

$\bar{s}, \bar{t}, \bar{u}$ Bags of resource terms

M, N, L Terms

Γ, Δ, Ω Contexts

s, t, u Resource terms

S, T, U Sums of resource terms

A, B, C Types

x, y, z Variables

$\vec{s}, \vec{t}, \vec{u}$ Sequences of resource terms

$\vec{A}, \vec{B}, \vec{C}$ Sequences of types

Games

$\mathsf{a}, \mathsf{b}, \mathsf{c}$ Arena events

$\mathsf{t}, \mathsf{u}, \mathsf{v}$ Linearisations

$\mathsf{A}, \mathsf{B}, \mathsf{C}$ Arenas

q, p, r Augmentations

x, y, z Configurations

a, b, c Configuration or augmentation events

u, v, w Interactions

$\mathsf{q}, \mathsf{p}, \mathsf{r}$ Isogmentations

s, t Plays

$\mathsf{x}, \mathsf{y}, \mathsf{z}$ Positions

f, g, h Renamings

σ, τ Strategies