

Internship Report

Deterministic Strategies in Concurrent Games and their Relation with Semantics and Logic

1 February — 23 August 2021

Samuel Debray

École Normale Supérieure Paris-Saclay
2nd year, Computer Science Department

Under the supervision of Prof. Glynn Winskel

University of Strathclyde
Mathematically Structured Programming Group

and Dr. Simon Castellan

INRIA Rennes-Bretagne-Atlantique
Celtique Team

Abstract

Games and strategies are informal concepts that are very useful in studying interactive systems such as open programs interacting with its environment. Most formalisations of these concepts use trees and, as such, are bound to only represent sequential systems. Recently Winskel and Rideau have introduced a new understanding of these concepts in terms of event structures, a well-known model of concurrency and nondeterminism that focusses on understanding the causal constraints of a system.

The aim of Glynn Winskel's work is now to prove how general concurrent games can be. This internship's purpose was to follow in his footsteps and prove equivalences between some particular concurrent games and other structures, namely domains (section 2) and dialectica categories (section 3). So as to do so, we show correspondences between morphisms of these structures: deterministic strategies and Berry-stable maps on the one hand, and deterministic winning strategies and dialectica maps on the other hand.

Parts of the proofs are unachieved, but we believe that all the stated theorems are true, and that there is not much work to do to complete the proofs.

Table of contents

| | |
|---|-----------|
| Introduction | 1 |
| Practical context of the internship | 1 |
| Context and motivations | 1 |
| State of the art | 1 |
| Notations and conventions | 1 |
| 1 Games as event structures | 3 |
| 1.1 Event structures | 3 |
| 1.2 Stable families | 5 |
| 1.3 Games and strategies | 7 |
| 2 Concurrent games and domains | 11 |
| 2.1 Berry-stable maps and domains | 11 |
| 2.2 Correspondence between deterministic strategies and Berry-stable maps | 12 |
| 3 Concurrent games and dialectica categories | 15 |
| 3.1 Some refinements of concurrent games | 15 |
| 3.2 Dialectica games | 16 |
| 3.3 Correspondence between deterministic winning strategies and dialectica maps | 16 |
| Conclusion | 18 |
| Contributions of the internship | 18 |
| Possible continuations | 18 |
| References | i |
| A The basics of category theory | ii |
| B Proofs of section 2 | iv |
| C Proofs of section 3 | vi |

Introduction

Practical context of the internship

This M1 internship took place during the Covid-19 pandemic. It was therefore impossible for me to go to Glasgow and the whole internship was conducted remotely. All the exchanges between my supervisors and me took place either by email or by zoom meetings, approximately once every month for the latters.

The biggest drawback of this situation is that communication with my supervisors was sometimes episodic, and it was very difficult to set up regular monitoring, Glynn Winskel being very busy elsewhere. The other pitfall is that remote work allows no integration in the laboratory whatsoever, and I feel like I have been missing an entire aspect of the life of a researcher. In particular, I think my work would very much have benefited from informal conversations that may happen next to the coffee machine, and which enable ideas to flow between interns and researchers.

These difficulties are the reason for the constant evolution of the topic of the internship: working on results that were already well known, although unproved, enabled my supervisors to help me more than they would have been able to with the initial project.

Context and motivations

Glynn Winskel is a big advocate of the ubiquity of games. The duality between two players, whether they are called "player" and "opponent", "prover" and "disprover", "input" and "output", etc. indeed exists in such various domains as proof theory, semantics, programming, verification or even economics: games are a good model of processes that involve *interaction* between two agents. One player, Opponent, represents the environment, while the other, Player, represents the system at hand. The way the two agents interact is described both by the strategies and the underlying structures, which set the rules for possible move for each agent.

To quote but a few applications, games are used to study equilibrium of systems, e.g. Nash equilibrium which allows several agent to maximise a payoff while taking into account the payoff of the other agents. The way agents maximise their payoff is prescribed by a strategy. Another application is game semantics, which grounds the concepts of truth or validity on game-theoretic concepts such as the existence of a winning strategy for a player.

However, games are often considered from a sequential point of view, and are therefore not suitable to be used in concurrency, when resources may be shared and threads may need to synchronise, or to express causal dependencies. This is the main motivation for games as event structures: they are one way of modelling such parallel procedures with causality.

Equipped with a structure which is compatible with parallelisation and causal dependency, games are likely to show up virtually anywhere in theoretical computer science. It therefore is very tempting to try to unify this science under the general notion of games on event structures. This internship is a step to this end since it links two major branches of computer science to game theory: semantics and logic.

State of the art

Concerning games and event structures, the main discoveries have been made by Glynn Winskel, together with others such as Simon Castellan, Pierre Clairambault, Claudia Faggian, Martin Hyland, Paul-André Melliès, Samuel Mimram, Mauro Piccolo and Sylvain Rideau. Several article on the subject have been published. Glynn Winskel wrote an introduction to games and event structures [Win17], which compiles most of his research on the topic and is more than a sufficient resource given the scope of this internship. However, it is nowhere near exhaustive because of the contributions of Simon Castellan, Pierre Clairambault and others. Related work on concurrent games with causality include that of Abramsky and Melliès [AM99], representing games as domains of positions and strategies as closure operators. Later Melliès and Mimram [MM07] related this latter position-based approach to the more usual play-based one in the framework of asynchronous games. In parallel, Faggian and Piccolo [FP09] developed a setting where partial orders were built up instead of being recovered a posteriori. The whole work was then generalised by Rideau and Winskel in 2011.

Only few equivalences between concurrent games and other structures have been proved yet, for instance with affine-stable maps, dI-domains or predicate calculus. More equivalences are known, but have not been published for the time being.

Notations and conventions

As a rule of thumb, events of an event structure E will be denoted e, e', \dots whereas configurations of E we be denoted x, y, \dots . If $f: A \rightarrow B$ and $X \subseteq A$, the direct image of X by f will be denoted by $f[X]$, and must not be confused with the downward-closure of a set of events (notation $1.G$).

This report is supposed to be self-contained to a certain extend. We only assume the reader knows the basic definitions of discrete mathematics and especially order theory. We also use the language of category theory, which is redefined in the appendix.

A symbol ✈ in the left margin means that the facing statement has not been proved in this report, by lack of time: the statement is thus parachuted from the plane.

Finally, we shall assume the axiom of choice is true.

The first section provides some general background about concurrent games, which are represented by event structures. The second section is dedicated to the proof of the correspondence between deterministic strategies and Berry-stable maps of domains. The last section tackles the equivalence between deterministic winning strategies and dialectica maps.

1 Games as event structures

In this section, we present the main definitions and results about games and event structures. For more details and for formal proofs, see [Win17].

1.1 Event structures

Event structures arise both from concurrency and domain theory. Historically, they first emerged in semantics and domain theory. While cpos allow to represent the behaviour of a program by iterating operations, dI-domains provide a structure to the stages of the iteration: they can be seen as sets of atoms representing special events of the execution thread. Each stage can thus be seen as a subset of a lattice 2^E where E is a set of events. Winskel showed that the right structure to provide to E is this of event structure: he found a correspondence between dI-domains and (configurations of) event structures [Win80].

On another note, event structures prove to be a powerful tool to describe concurrent processes. The most common way to describe a process is to give a trace of it: consider a lift consisting of a single cabin, three doors and as many calling devices, the process of calling the lift from the third floor and going to the first floor can be described as

CallLiftFromFloor3 · GoToFloor3 · OpenDoorFloor3 · GoToFloor1 · OpenDoorFloor1.

Now consider that two people are parallelly calling the lift from two different floor, the system then becomes almost impossible to describe sequentially: the number of states explodes exponentially; in addition, dependencies between events become very difficult to spot. Event structures provide a much more practical framework to represent such concurrent systems.

1.1.1 Definition

So as to be able to represent accurately concurrent computations, along with causal dependency and mutual exclusion, Winskel introduced event structures.

Definition 1.A (event structure). An *event structure* comprises a tuple $\langle E, \leq_E, \text{Con}_E \rangle$ consisting of a set of events E partially ordered by a causal dependency relation \leq_E , along with a set Con_E of finite consistent subsets of E . In addition, the following must hold:

- (i) for every $e \in E$, $\{e' \mid e' \leq_E e\}$ is finite;
- (ii) for every $e \in E$, $\{e\} \in \text{Con}_E$;
- (iii) for every X and Y , if $X \subseteq Y$ and $Y \in \text{Con}_E$ then $X \in \text{Con}_E$;
- (iv) for every $e, e' \in E$ and $X \in \text{Con}_E$, if $e \leq_E e'$ and $e' \in X$ then $X \cup \{e\} \in \text{Con}_E$.

Remark. Throughout this report, E might either denote an event structure or its underlying set, depending on the context.

The intuition is that event structures represent a system as a set of events and describe how events can depend on others, or prevent others from occurring.

Event structures are very general objects. Hereafter, we define some restrictions of its relations of causal dependency and consistency so as to make definitions and proofs more readable.

Definition 1.B (immediate causal dependency). Let E be an event structure and $e, e' \in E$. We write $e \rightarrow e'$ iff $e \leq_E e'$ and there exists no $e'' \in E$ such that $e \leq_E e'' \leq_E e'$. This relation is called *immediate causal dependency*.

Remark. \leq_E is the reflexive transitive closure of \rightarrow .

Definition 1.C (concurrency relation). Two events e and e' are said to be *concurrent* iff $\{e, e'\} \in \text{Con}_E$ and e and e' are incomparable with respect to \leq_E .

Notation 1.D. If e and e' are concurrent, we denote $e \text{ co } e'$ to express this relation.

Concurrent events can occur together, independently from each other.

Here is a first example of event structure. This example will be used and enriched throughout this subsection to illustrate various concepts and definitions.

Example 1.i. We can represent the functioning of a vending machine with an event structure. The set of events is

$\{\text{InsertCoin}, \text{SelectTea}, \text{SelectCoffee}, \text{PrepareTea}, \text{PrepareCoffee}\}$

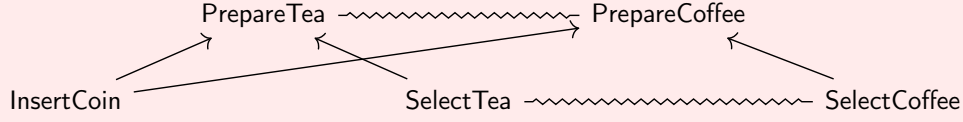
with immediate causal dependencies

$\text{InsertCoin} \rightarrow \text{PrepareTea}, \quad \text{InsertCoin} \rightarrow \text{PrepareCoffee}, \quad \text{SelectTea} \rightarrow \text{PrepareTea}, \quad \text{SelectCoffee} \rightarrow \text{PrepareCoffee}$

and

$$\{\text{SelectTea}, \text{SelectCoffee}\}, \{\text{PrepareTea}, \text{PrepareCoffee}\} \notin \text{Con}.$$

This event structure can be diagrammatically represented as follows



with an arrow $e \longrightarrow e'$ stating that $e \leq e'$ and a drawing $e \rightsquigarrow e' \rightsquigarrow e''$ stating that $\{e, e', e''\}, \{e, e'\}, \{e, e''\}, \{e', e''\} \notin \text{Con}$. In practice, it is not necessary to state the conflict $\text{PrepareTea} \rightsquigarrow \text{PrepareCoffee}$ since it is inherited from the conflict $\text{PrepareTea} \rightsquigarrow \text{PrepareCoffee}$ by axiom (iv) of definition 1.A.

1.1.2 Configurations

Processes often have a temporal execution. So as to be able to represent a trace of an execution, we need to introduce configurations.

Definition 1.E (configuration). Let $\langle E, \leq_E, \text{Con}_E \rangle$ be an event structure. A *configuration* of E is a subset x of E such that:

- (i) for every finite subset X of x , we have $X \in \text{Con}_E$; *consistency*
- (ii) for every $e, e' \in E$ such that $e \leq_E e'$ and $e' \in x$, we have $e \in x$. *downward-closure*

Notation 1.F. Let E be an event structure. The set of configurations of E is denoted by $\mathcal{C}^\infty(E)$ and the subset of *finite* configurations is denoted by $\mathcal{C}(E)$.

Configurations are the history of a process. They are the set of events which may have occurred at some stage of the execution.

Remark. Note that $\langle \mathcal{C}^\infty(E), \subseteq \rangle$ is a poset.

Example 1.ii. The configurations of the event structure defined in example 1.i are: \emptyset , $\{\text{InsertCoin}\}$, $\{\text{SelectTea}\}$, $\{\text{SelectTea}, \text{InsertCoin}\}$, $\{\text{InsertCoin}, \text{SelectTea}\}$, $\{\text{InsertCoin}, \text{SelectCoffee}\}$, $\{\text{InsertCoin}, \text{SelectTea}, \text{PrepareTea}\}$ and $\{\text{InsertCoin}, \text{SelectCoffee}, \text{PrepareCoffee}\}$.

Note that *causal dependency is conjunctive*: an event can occur only if all the events he depends on have occurred.

As often, it will be useful to look at the least configuration which contains a certain event.

Notation 1.G. Let E be an event structure and $e \in E$. We shall denote by $[e]$ the downward-closure of e in E with respect to \leq_E .

Remark. We shall also overload this notation and denote by $[X]$ the downward-closure of any subset $X \subseteq E$.

Proposition 1.1. Given an event e in an event structure E , $[e]$ is the least configuration of E that contains e .

The above proposition justifies that $[e]$ is called the *configuration generated by e* .

Example 1.iii. In example 1.i, the configuration generated by InsertCoin is $\{\text{InsertCoin}\}$ and that generated by PrepareTea is $\{\text{InsertCoin}, \text{SelectTea}, \text{PrepareTea}\}$.

Finally, the two following notations will prove to be useful later. They basically help expressing that some configurations can extend each other: it is possible for a process to have two concurrent execution threads.

Notation 1.H. Let E be an event structure and $x, y \in \mathcal{C}^\infty(E)$. We write $x \uparrow y$ iff there exists some $z \in \mathcal{C}^\infty(E)$ such that $x \subseteq z$ and $y \subseteq z$.

Notation 1.I. Let E be an event structure, $e \in E$ and $x \in \mathcal{C}(E)$. We write $x \xrightarrow{e} \cdot$ iff $x \cup \{e\} \in \mathcal{C}(E)$, that is iff there exists some $y \in \mathcal{C}(E)$ such that $y = x \cup \{e\}$. In this case, we also write $x \xrightarrow{e} y$.

1.1.3 Maps of event structures

As always when we define new structures, it is important to define morphisms. By doing so, we almost define a category. The heuristic of this definition is that a map $f: E \rightarrow F$ is a way of emulating system E inside system F .

Definition 1.J (map of event structures). Let E and F be two event structures. A partial map $f: E \rightarrow F$ is said to be a *map of event structures* iff the following hold:

- (i) for every $x \in \mathcal{C}^\infty(E)$, $f[x] \in \mathcal{C}^\infty(F)$; *preservation of configurations*
- (ii) for every $x \in \mathcal{C}^\infty(E)$ and $e, e' \in x \cap \text{dom}(f)$, if $f(e) = f(e')$ then $e = e'$. *local injectivity*

In general, maps of event structures do not preserve causality but they reflect it locally, as well as concurrency.

Proposition 1.2. Let $f: E \rightarrow E'$ be a map of event structures.

- (i) f *locally reflects causal dependency*: for every $x \in \mathcal{C}^\infty(E)$ and $e, e' \in x$ such that $f(e)$ and $f(e')$ are both defined, if $f(e) \leq_{E'} f(e')$ then $e \leq_E e'$;
- (ii) f *preserves the concurrency relation*, when defined: if $e \text{ co } e'$ in E and $f(e)$ and $f(e')$ are both defined then $f(e) \text{ co } f(e')$.

We now have all the definitions needed to state the following proposition-definition.

Proposition 1.3. The set of event structures with partial maps form a category \mathcal{E} .

1.2 Stable families

Event structures make it thorny to express that an event may occur in two different cases: dependency is conjunctive. For instance, in example 1.i, one could decide that the machine prepares tea either when the customer inserts a coin and selects tea, or when he inserts his credit card and selects tea. Should we want to express this with an event structure, we would need to introduce another event $\text{PrepareTea}'$ and decide that PrepareTea occurs when customer pays cash and $\text{PrepareTea}'$ occurs when he pays card, even though both events are indistinguishable. Stable families are designed to make it easier to introduce disjunctive dependency.

1.2.1 Definition

The notion of stable family is designed to extend this of finite configurations event structure by enabling an event to occur in several possibly incompatible ways.

Notation 1.K. Given a family \mathcal{F} and a subset $Z \subseteq \mathcal{F}$, we write $Z \uparrow$ iff there exists some $x \in \mathcal{F}$ such that for every $z \in Z$, $z \subseteq x$. Similarly, we shall denote $Z \uparrow^{\subseteq+}$ (resp. $Z \uparrow^{\subseteq-}$) iff there exists some $x \in \mathcal{F}$ such that for every $z \in Z$, $z \subseteq^+ x$ (resp. $z \subseteq^- x$).

This notation is useful to state the following definition of stable families.

Definition 1.L (stable family). A *stable family* comprises a nonempty family of finite configurations satisfying:

- (i) for every $Z \subseteq \mathcal{F}$, if $Z \uparrow$ then $\bigcup Z \in \mathcal{F}$; *completeness*
 - (ii) for every $Z \subseteq \mathcal{F}$, if $Z \uparrow$ and $Z \neq \emptyset$ then $\bigcap Z \in \mathcal{F}$; *stability*
 - (iii) for every $x \in \mathcal{F}$ and $e \neq e' \in x$, there exists some $y \in \mathcal{F}$ such that $y \subseteq x$ and $e \in y \iff e' \notin y$. *coincidence-freeness*
- The elements of $\bigcup \mathcal{F}$ are called *events* of \mathcal{F} .

Example 1.iv. If E is an event structure, it is routine to check that $\mathcal{C}(E)$ is a stable family.

Remark. This notation expresses the compatibility of Z . Notation 1.H is nothing but a syntactic sugar for $\{x, y\} \uparrow$ in the stable family $\mathcal{C}(E)$.

Here again, we define morphisms between stable families.

Definition 1.M (map of stable families). Let \mathcal{F} and \mathcal{G} be two stable families. A partial map $f: \bigcup \mathcal{F} \rightarrow \bigcup \mathcal{G}$ is said to be a *map of stable families* iff the following hold:

- (i) for every $x \in \mathcal{F}$, $f[x] \in \mathcal{G}$; *preservation of configurations*
- (ii) for every $x \in \mathcal{F}$ and $e, e' \in x \cap \text{dom}(f)$, if $f(e) = f(e')$ then $e = e'$. *local injectivity*

Remark. If $f: \bigcup \mathcal{F} \rightarrow \bigcup \mathcal{G}$ is a map of stable families we shall denote $f: \mathcal{F} \rightarrow \mathcal{G}$ the spec of f , even though $\text{dom}(f) \subseteq \bigcup \mathcal{F}$ and $\text{codom}(f) \subseteq \bigcup \mathcal{G}$.

Without any surprise, we now define a category of stable families.

Proposition 1.4. The set of stable families with maps form a category \mathcal{S} .

1.2.2 Relation with event structures

So far, we have built two categories: \mathcal{E} and \mathcal{S} . The aim of this paragraph is to provide a relation between these two categories. More precisely we are going to build functors between these, that is way a of turning objects (resp. maps) of \mathcal{E} into objects (resp. maps) of \mathcal{S} which behaves well with respect to composition.

Given an event structure E , one can easily check that $\mathcal{C}(E)$ is a stable family (example 1.iv). Hence we have built a functor $\mathcal{C}: \mathcal{E} \rightarrow \mathcal{S}$. We will now provide a right adjoint to this functor, that is a kind of inverse of \mathcal{C} .

Proposition 1.5. *Let x be a configuration of a stable family \mathcal{F} . For $e, e' \in x$ define*

$$e' \leq_x e \quad \text{iff} \quad \forall y \in \mathcal{F} \cdot y \subseteq x \wedge e \in y \implies e' \in y.$$

When $e \in x$ define the prime configuration

$$[e]_x := \bigcap \{y \in \mathcal{F} \mid y \subseteq x \wedge e \in y\}.$$

Then \leq_x is a partial order and $[e]_x$ is a configuration such that

$$[e]_x = \{e' \in x \mid e' \leq_x e\}.$$

Moreover the configurations $y \in \mathcal{F}$ such that $y \subseteq x$ are exactly those subsets that are closed under \leq_x .

The above proposition allows to let the following definition.

Definition 1.N. Let \mathcal{F} be a stable family. We set $\text{Pr}(\mathcal{F}) := \langle P, \leq, \text{Con} \rangle$ to be the event structure defined by:

- $P = \{[e]_x \mid e \in x \wedge x \in \mathcal{F}\};$
- for every $Z \subseteq P$, $Z \in \text{Con}$ iff $\bigcup Z \in \mathcal{F};$
- for every $p, p' \in P$, $p \leq p'$ iff $p \subseteq p'.$

We can now verify that Pr is right adjoint to \mathcal{C} .

Lemma 1.6. *If \mathcal{F} is a stable family then $\text{Pr}(\mathcal{F})$ is an event structure.*

Theorem 1.7. *Pr is a functor $\mathcal{S} \rightarrow \mathcal{E}$ and it is right adjoint to \mathcal{C} .*

Corollary 1.7.1. • *The unit of the adjunction is the family of isomorphisms $\eta_E: E \rightarrow \text{Pr}(\mathcal{C}(E))$ such that, for every $E \in \text{ob}(\mathcal{E})$, η_E maps an event e to the configuration $[e]$ it generates.*

$$\begin{array}{ccc} \mathcal{C}(E) & \xrightarrow{g} & \mathcal{F} \\ \text{Pr}(\mathcal{C}(E)) & \xrightarrow{\text{Pr}(g)} & \text{Pr}(\mathcal{F}) \\ \eta_E \uparrow & \nearrow f & \\ E & & \end{array}$$

- *The co-unit is the family of morphisms $\text{top}_{\mathcal{F}}: \mathcal{C}(\text{Pr}(\mathcal{F})) \rightarrow \mathcal{F}$, each of them mapping $[e]_x$ to e .*

$$\begin{array}{ccc} E & \xrightarrow{f} & \text{Pr}(\mathcal{F}) \\ \mathcal{C}(E) & \xrightarrow{\mathcal{C}(f)} & \mathcal{C}(\text{Pr}(\mathcal{F})) \\ & \searrow g & \downarrow \text{top}_{\mathcal{F}} \\ & & \mathcal{F} \end{array}$$

Beware that, contrary to E and $\text{Pr}(\mathcal{C}(E))$ which are isomorphic as event structures, \mathcal{F} and $\mathcal{C}(\text{Pr}(\mathcal{F}))$ are not isomorphic as stable families, as shown in the following example.

Example 1.v. Let

$$\mathcal{F} := \{\{\text{InsertCoin}\}, \{\text{Hack}\}, \{\text{InsertCoin}, \text{PrepareTea}\}, \{\text{Hack}, \text{PrepareTea}\}\}.$$

Since \mathcal{F} provides two different ways of getting a cup of tea, either by paying for it or by hacking the machine, $\text{Pr}(\mathcal{F})$

will contain two distinct events that correspond to the initial PrepareTea:

$$\text{Pr}(\mathcal{F}) = \{\text{InsertCoin}, \text{Hack}, \text{PrepareTea}_{\text{InsertCoin}}, \text{PrepareTea}_{\text{Hack}}\}.$$

There can therefore be no isomorphism from \mathcal{F} to $\mathcal{C}(\text{Pr}(\mathcal{F}))$ since the latter contains more events than the former.

However, the domains $\langle \mathcal{F}, \subseteq \rangle$ and $\langle \mathcal{C}(\text{Pr}(\mathcal{F})), \subseteq \rangle$ are isomorphic.

Theorem 1.8. *Let \mathcal{F} be a stable family. Then there exists an order isomorphism $\hat{\cdot}$*

$$\hat{\cdot}: \begin{cases} \mathcal{C}(\text{Pr}(\mathcal{F})) & \longrightarrow & \mathcal{F} \\ x & \longmapsto & \bigcup x \end{cases}.$$

Its inverse is the morphism $[\cdot]$

$$[\cdot]: \begin{cases} \mathcal{F} & \longrightarrow & \mathcal{C}(\text{Pr}(\mathcal{F})) \\ x & \longmapsto & \{[e]_x \mid e \in \mathcal{F}\} \end{cases}.$$

1.3 Games and strategies

Now that we have defined the very general event structures, we want to provide a way of interpreting games as such structures. The main asset of these games is their generality and the possibility of using the tools of category theory to deal with composition, strategies and so on, whereas graph theoretic games for instance would force us to use ad hoc methods. These games are more expressive than standard sequential games: for instance, moves do not need to alternate between Player and Opponent.

1.3.1 Games and their operations

Since games must have (at least) two players, we introduce polarities in our event structures.

Definition 1.0 (event structure with polarities). An *event structure with polarities* is a tuple $\langle E, \text{pol}_E \rangle$ where E is an event structure and $\text{pol}_E: E \rightarrow \{+, -\}$ labels every event of E with a polarity. An event structure with polarities is said to be *positive* (resp. *negative*) when all its events have polarity $+$ (resp. $-$).

A game G can be represented by an event structure with polarities $\langle E, \text{pol}_E \rangle$ whose events are moves in G and where polarity $+$ denotes Player's moves and polarity $-$ denotes Opponent's moves. Dependency and consistency relations stand for the constraints imposed by the game. Beware, though, that thinking of an event structure as a game, albeit mathematically correct, is not always relevant. This is why we will keep distinguishing games and event structures with polarities.

We now define two major operations on games: dual and parallel composition (which juxtaposes two games).

Definition 1.P (dual game). Let A be a game. The *dual game* A^\perp of A comprises a copy the event structure A with reversed polarities.

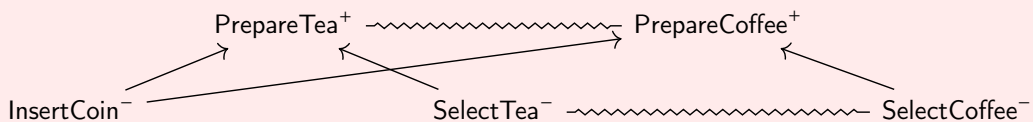
Notation 1.Q. The complementary event of $e \in A$ is denoted by $\bar{e} \in A^\perp$.

Definition 1.R (parallel composition). Let A and B be two games. The *parallel composition* $A \parallel B$ of A and B is defined as follows:

- the events of $A \parallel B$ are $\{1\} \times A \cup \{2\} \times B$;
- $e \leq_{A \parallel B} e'$ iff $e = (1, a)$, $e' = (1, a')$ and $a \leq_A a'$ or $e = (2, b)$, $e' = (2, b')$ and $b \leq_B b'$;
- $C \in \text{Con}_{A \parallel B}$ iff $\{a \mid (1, a) \in C\} \in \text{Con}_A$ and $\{b \mid (2, b) \in C\} \in \text{Con}_B$.

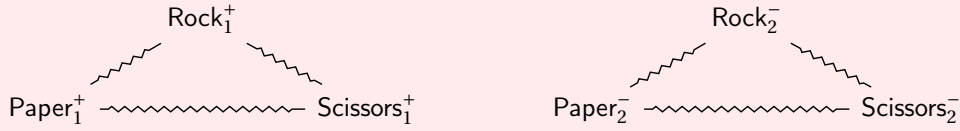
Remark. Parallel composition can be intuitively extended to subsets of configurations $X \subseteq \mathcal{C}^\infty(A)$ and $Y \subseteq \mathcal{C}^\infty(B)$ as follows: $X \parallel Y := \{\{1\} \times x \cup \{2\} \times y \mid x \in X \wedge y \in Y\}$.

Example 1.vi. We can consider the event structure from example 1.i as a game where Player is the machine and Opponent is the customer:



The following example is the one we will now use to illustrate concepts on games.

Example 1.vii. We can represent the well-known Rock-Paper-Scissors by the following event structure with polarities:



1.3.2 Pre-strategies and copycat

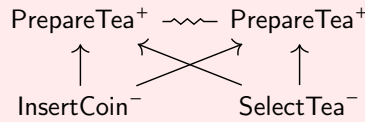
Since games are now represented by event structures, it seems only natural to represent strategies on games by maps of event structures, since the former can be considered "morphisms of games".

Definition 1.S (pre-strategy). A *pre-strategy* on a game A is total map of event structures $\sigma: S \rightarrow A$ which preserves polarity, where S is an event structure with polarities.

A pre-strategy represents a nondeterministic play of the game: all moves are allowed, the only rules are these of the game. Pre-strategies have negative and positive moves; negative moves amount to receiving a request from the context, and positive moves to issuing a request or answering a previous request.

They are to be thought of as labelling functions $\sigma: S \rightarrow A$; A brings constraints that the labelling has to respect.

Example 1.viii. The following is a pre-strategy on the game defined in example 1.vi, with obvious labelling:



Note that this strategy does not allow SelectCoffee to occur for instance. Besides, this strategy, albeit locally injective, is not injective: the machine secretly decides what event PrepareTea occurs, but this is invisible for A since it prepares tea anyway. Therefore, the strategy is neither injective nor surjective.

Given two games A and B , it is also relevant to define what a pre-strategy from A to B is: it is a pre-strategy on $A^\perp \parallel B$. This definition is similar to that of implication in classical logic: $A \Rightarrow B$ is defined by $\neg A \vee B$. In practice, $\sigma: S \rightarrow A^\perp \parallel B$ adds causal dependencies between A and B ; these account for dependencies between input (in A) and output (in B).

We now show how the notion of pre-strategy is too relaxed to get a categorical setting. To do so, we need to define the copy-cat strategy, a strategy for Player based on copying the latest move of Opponent.

Notation 1.T. For $c \in A^\perp \parallel A$, we denote by \bar{c} the corresponding copy of c , of opposite polarity, in the alternative component, that is $(1, a) = (2, \bar{a})$ and $(2, a) = (1, \bar{a})$.

Proposition 1.9. Let A be a game. There exists an event structure with polarities \mathbb{C}_A having the same events and polarity as $A^\perp \parallel A$ but with causal dependency $\leq_{\mathbb{C}_A}$ given as the transitive closure of the relation

$$\leq_{A^\perp \parallel A} \cup \{(\bar{c}, c) \mid c \in A^\perp \parallel A \wedge \text{pol}_{A^\perp \parallel A}(c) = +\}$$

and finite subsets of \mathbb{C}_A are consistent iff their downward-closure with respect to $\leq_{\mathbb{C}_A}$ are consistent in $A^\perp \parallel A$.

Definition 1.U (copy-cat). Let A be a game. The *copy-cat* strategy from A to A is a pre-strategy $\alpha_A: \mathbb{C}_A \rightarrow A^\perp \parallel A$ where \mathbb{C}_A comprises the event structure with polarities $A^\perp \parallel A$ together with extra causal dependencies $\bar{c} \leq_{\mathbb{C}_A} c$ for every event c such that $\text{pol}_{A^\perp \parallel A}(c) = +$, and α_A is the identity on the set of events common to both \mathbb{C}_A and $A^\perp \parallel A$.

α_A is a concurrent strategy where Player's moves (of polarity $+$) always copy previous corresponding moves of Opponent (of polarity $-$).

Given two strategies, $\sigma: S \rightarrow A^\perp \parallel B$ from A to B and $\tau: T \rightarrow B^\perp \parallel C$ from B to C , [Win17] provides two ways of composing σ and τ to form a strategy $\tau \odot \sigma: U \rightarrow A^\perp \parallel C$ from A to C . The first way, by hiding, is more intuitive and ad hoc, and illustrates the notion of synchronisation. The second is more categorical and uses pullbacks. Anyway, the idea of composition comes both from the world of categories and from the intuition that processes which share a common resource may need to synchronise at some point.

However, α happens not to behave well with respect to composition. We would indeed expect α to be the unit of composition, which is not the case. Strategies are therefore defined to be those pre-strategies for which α behaves as the unit with respect to composition.

1.3.3 From pre-strategies to strategies

It is shown in [Win17] that the following definition is suitable for strategies, that is that these strategies coincide with strategies as we defined them in section 1.3.2.

Definition 1.V (strategy). A pre-strategy $\sigma: S \rightarrow A$ is called a *strategy* iff the following hold:

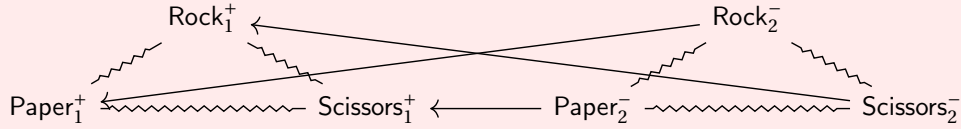
- (i) for every $a \in A$ and $x \in \mathcal{C}(S)$, if $\sigma[x] \xrightarrow{a} \sqsubset$ and $\text{pol}_A(a) = -$ then there exists a unique $s \in S$ such that $x \xrightarrow{s} \sqsubset$ and $\sigma(s) = a$; receptivity
- (ii) for every $s, s' \in S$, if $s \rightarrow s'$ and $\text{pol}_S(s) = +$ then $\sigma(s) \rightarrow \sigma(s')$; + -innocence
- (iii) for every $s, s' \in S$, if $s \rightarrow s'$ and $\text{pol}_S(s') = -$ then $\sigma(s) \rightarrow \sigma(s')$. - -innocence

Receptivity ensures that no Opponent's move that is possible is disallowed by σ .

Innocence gives Player the right to await Opponent's moves before playing, but prevents Player to force Opponent to wait for his move. This is done by allowing Player to introduce locally relations of immediate causal dependency of the form $\ominus \rightarrow \oplus$ but disallowing those of the form $\oplus \rightarrow \ominus$ (other than those stipulated in the game A).

Example 1.ix. • The pre-strategy presented in example 1.viii is not a strategy because it is not receptive: take $x := \{\text{InsertCoin}^-\}$ and $a = \text{SelectCoffee}^-$. However, it is a strategy if we consider that the vending machine only sells tea.

- The following is a (cheaty) strategy on the game defined in example 1.vii, with obvious labelling:



In this strategy, Player waits for Opponent to choose his move and then decides to play the move that wins over Opponent's move.

The following proposition provides a characterisation of strategies.

Proposition 1.10. Let S and A be two event structures with polarity and $\sigma: S \rightarrow A$ be a total map of event structures that preserves polarity. Then σ is a strategy on A iff the following hold for every $x \in \mathcal{C}^\infty(S)$ and $y \in \mathcal{C}^\infty(A)$:

- (i) if $y \sqsubseteq^+ \sigma[x]$ then there exists some (necessarily unique) $x' \in \mathcal{C}^\infty(S)$ such that $x' \sqsubseteq x$ and $\sigma[x'] = y$;

$$\begin{array}{ccc} x' & \dots \sqsubseteq & x \\ \sigma \downarrow & & \downarrow \sigma \\ y & \sqsubseteq^+ & \sigma[x] \end{array}$$

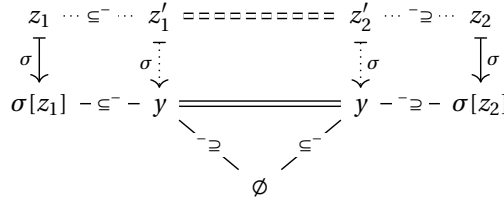
- (ii) if $\sigma[x] \sqsubseteq^- y$ then there exists a unique $x' \in \mathcal{C}^\infty(S)$ such that $x \sqsubseteq x'$ and $\sigma[x'] = y$.

$$\begin{array}{ccc} x & \dots \sqsubseteq & x' \\ \sigma \downarrow & & \downarrow \sigma \\ \sigma[x] & \sqsubseteq^- & y \end{array}$$

Corollary 1.10.1. Let $\sigma: S \rightarrow A$ be a strategy on a game A and z_1, z_2 be two configurations in $\mathcal{C}^\infty(S)$ only made up of negative events and such that $\sigma[z_1] \uparrow^{\sqsubseteq^-} \sigma[z_2]$. Then $z_1 \uparrow^{\sqsubseteq^-} z_2$.

Proof. Since $\sigma[z_1] \uparrow^{\sqsubseteq^-} \sigma[z_2]$ there exists some $y \in \mathcal{C}^\infty(A)$ such that $\sigma[z_1] \sqsubseteq^- y$ and $\sigma[z_2] \sqsubseteq^- y$. Now proposition 1.10 ensures that there exists some unique $z'_1, z'_2 \in \mathcal{C}^\infty(S)$ such that $\sigma[z'_1] = y = \sigma[z'_2]$ and $z_1 \sqsubseteq z'_1$ and $z_2 \sqsubseteq z'_2$. However z_1 is only made up of negative events thus so is y and proposition 1.10 ensures the uniqueness of $z' \in \mathcal{C}^\infty(S)$ such that $\sigma[z'] = y$. Therefore

$z'_1 = z' = z'_2$ which proves that $z_1 \uparrow^{\leq^-} z_2$.



■

1.3.4 Deterministic strategies

The above-presented strategies are nondeterministic, which means that, given a move for Opponent, Player may have several ways of playing. We want to limit the moves such that consistency is only influenced by Opponent's moves.

Notation 1.W (negative part). Let us denote by $\text{Neg}[X] := \{s' \in S \mid \text{pol}(s') = - \wedge \exists s \in X \cdot s' \leq s\}$ the *negative part* of $X \subseteq S$ where S is an event structure with polarity.

The intuition is that $\text{Neg}[X]$ is the set of Opponent's moves on which moves in X causally depend.

Definition 1.X (deterministic strategy). An event structure with polarity S is said to be *deterministic* iff for every finite subset $X \subseteq S$ such that $\text{Neg}[X] \in \text{Con}_S$ we have $X \in \text{Con}_S$.
A strategy $\sigma: S \rightarrow A$ is *deterministic* iff S is.

An event structure with polarities S is deterministic iff any set of moves is consistent when it causally depends on a consistent set of moves for Opponent.

Example 1.x. • The strategy on the vending machine selling only tea presented in example 1.viii is not deterministic.
• The strategy on Rock-Paper-Scissors presented in example 1.ix is deterministic.

The following proposition provides a useful characterisation of determinism.

Proposition 1.11. *An event structure with polarity S is deterministic iff*

$$\forall s, s' \in S, \forall x \in \mathcal{C}(S) \cdot x \xrightarrow{s} \subset \wedge x \xrightarrow{s'} \subset \wedge \text{pol}_S(s) = + \implies x \cup \{s, s'\} \in \mathcal{C}(S).$$

2 Concurrent games and domains

The aim of this section is to prove that deterministic strategies between two positive games (that is games where only Player is allowed to move) are exactly Berry-stable maps.

2.1 Berry-stable maps and domains

The study of domains arose in semantics with the work of Scott and Plotkin. Dana Scott first tried to give semantics to programming languages in the late 1960s. One of the major problems he faced was to give a semantics to fixed-point combinators, such as Curry's Y combinator defined as $\lambda f \cdot (\lambda x \cdot f(xx))(\lambda x \cdot f(xx))$ and such that $Yg =_\beta g(Yg)$. So as to give a model of these combinators, it is necessary to find fixpoints of functions. A first answer was brought by Knaster-Tarski theorem for lattices, but it may require transfinite iteration, in addition to the fact that lattices are very constraining structures. Indeed, the only property we would expect from models of λ -calculus is monotonicity. The reason for monotonicity is that Scott uses partial orders to represent partial results of computation (think of the evaluation of the command `while true`, which cannot return exact result): the higher an element in the hierarchy, the more precise it is. Cpos are structures where bare monotonicity almost ensures the existence of a fixpoint thanks to Kleene theorem, which not only ensures their existence, but also states that the lowest fixpoint can be reached by a countable number of successive approximations: $\text{lfp}(f) = \bigcup_{n \in \mathbb{N}} f^n(\perp)$. The only other hypothesis is that functions must preserve supremum of directed families, which means that functions turn consistent computations into consistent computation: directed families indeed model sets of partial results in which no two elements are contradictory. Those maps that verify both hypothesis of Kleene theorem are called Scott-continuous, and cpos with Scott-continuous maps are the cornerstone of domain theory.

Here, we consider maps that are not only Scott-continuous but also stable, following Gérard Berry's work [Ber78]. Stable maps historically appeared when Berry tried to give a semantics to sequential languages like ALGOL. The idea behind these is that a stable map h makes it possible for every α and $\beta \subseteq h(\alpha)$ to seize the information needed from α to reach β by applying h .

Definition 2.A (Berry-stable map). A map $f: \langle \mathcal{C}^\infty(A), \subseteq \rangle \rightarrow \langle \mathcal{C}^\infty(B), \subseteq \rangle$ is said to be *Berry-stable* iff the following hold:

- (i) f is Scott-continuous;
- (ii) for every $x, y \in \mathcal{C}^\infty(A)$ such that $x \uparrow y$ we have $f(x \cap y) = f(x) \cap f(y)$.

Remark. From the definition, it is straightforward to see that if $f: \mathcal{C}^\infty(A) \rightarrow \mathcal{C}^\infty(B)$ and $g: \mathcal{C}^\infty(B) \rightarrow \mathcal{C}^\infty(C)$ are Berry-stable, then so is $g \circ f$.

First, we prove a lemma stating that condition (ii) is actually valid for any consistent set of strategies.

Lemma 2.1. *let $f: \mathcal{C}^\infty(A) \rightarrow \mathcal{C}^\infty(B)$ be a Berry-stable map. Then for every non-empty $Z \subseteq \mathcal{C}^\infty(A)$ such that $Z \uparrow$,*

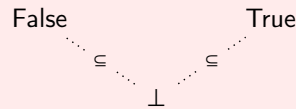
$$f\left(\bigcap Z\right) = \bigcap f(Z).$$

Proof. See appendix B. ■

Example 2.i. Consider the event structure Bool of booleans:

$$\text{False} \rightsquigarrow \text{True}$$

with all events of polarity +. The stable family of its configurations $\mathcal{C}^\infty(\text{Bool})$ is



where \perp denotes the empty configuration, which is to be interpreted as "unknown argument".

There are three evaluation strategies for the operator $\wedge: \mathcal{C}^\infty(\text{Bool})^2 \rightarrow \mathcal{C}^\infty(\text{Bool})$: two lazy evaluations (left-first and right-first) and the parallel one. They are defined hereafter:

$$\wedge_{\text{left}}: \begin{cases} \perp, b & \mapsto \perp \\ \text{False}, b & \mapsto \text{False} \\ \text{True}, b & \mapsto b \end{cases} \quad \wedge_{\text{right}}: \begin{cases} b, \perp & \mapsto \perp \\ b, \text{False} & \mapsto \text{False} \\ b, \text{True} & \mapsto b \end{cases} \quad \wedge_{\text{par}}: \begin{cases} \perp, \perp & \mapsto \perp \\ \text{False}, b & \mapsto \text{False} \\ b, \text{False} & \mapsto \text{False} \\ \text{True}, \text{True} & \mapsto \text{True} \end{cases}$$

for every $b \in \mathcal{C}^\infty(\text{Bool})$. Note that $\mathcal{C}^\infty(\text{Bool})^2 \cong \mathcal{C}^\infty(\text{Bool} \parallel \text{Bool})$, which is why we chose the domain of \wedge_{par} , \wedge_{left} and \wedge_{right} to be $\mathcal{C}^\infty(\text{Bool} \parallel \text{Bool})$ to respect the signature of Berry-stable maps. Among these three maps, only \wedge_{left} and \wedge_{right} are Berry-stable: take $x := \{\text{False}\} \parallel \perp$ and $y := \perp \parallel \{\text{False}\}$, then $x, y \in \mathcal{C}^\infty(\text{Bool} \parallel \text{Bool})$ and $x \cap y = \perp \parallel \perp$ but $\wedge_{\text{par}}(x) \cap \wedge_{\text{par}}(y) = \text{False}$ and $\wedge_{\text{par}}(x \cap y) = \perp$.

2.2 Correspondence between deterministic strategies and Berry-stable maps

2.2.1 From strategies to Berry-stable maps

The following lemma simply states that strategies behave good with respect to intersection of direct images of consistent sets. Its proof is straightforward and uses local injectivity.

Lemma 2.2. *Let $\sigma: S \rightarrow A$ be a strategy and $x, y \in \mathcal{C}^\infty(S)$ such that $x \uparrow y$. Then*

$$\sigma[x \cap y] = \sigma[x] \cap \sigma[y].$$

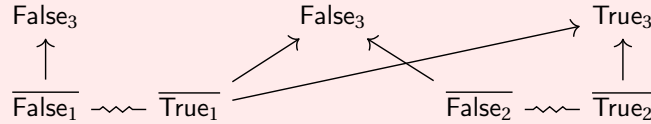
Theorem 2.3. *Let A and B be two positive games and $\sigma: S \rightarrow A^\perp \parallel B$ be a deterministic strategy from A to B . Then σ induces a Berry-stable map $\tilde{\sigma}$*

$$\tilde{\sigma}: \begin{cases} \mathcal{C}^\infty(A) & \longrightarrow & \mathcal{C}^\infty(B) \\ x & \longmapsto & \bigcup \{y \mid y \in \mathcal{C}^\infty(B) \wedge \exists z \in \mathcal{C}^\infty(S) \cdot \bar{x} \parallel y = \sigma[z]\} \end{cases}.$$

Proof. See appendix B. ■

The intuition behind the definition of $\tilde{\sigma}$ is that an input $x \in \mathcal{C}^\infty(A)$ induces, by receptivity and determinism, a unique +-maximal configuration (see definition 3.B) z of S . In addition, z is such that $\sigma[z] = \bar{x} \parallel y$ for some $y \in \mathcal{C}^\infty(B)$ (see lemma 3.1). Thus y is uniquely determined and can be seen as the output of σ on input x . Besides, this transformation turns out to be Berry-stable, that is why we let $\tilde{\sigma}(x) = y$.

Example 2.ii. Let us write what we would expect a strategy $S \rightarrow (\text{Bool} \parallel \text{Bool})^\perp \parallel \text{Bool}$ for lazy left-first "and" evaluation to be:



This strategy basically says the following:

- the first argument is either True or False but not both;
- the second argument is either True or False but not both;
- in order to get output False, you must either have first argument set to False or first argument set to True and second argument set to False;
- in order to get output True, you must have both arguments set to True.

The Berry-stable map associated with this strategy by theorem 2.3 turns out to be \wedge_{left} .

2.2.2 From Berry-stable maps to strategies

We now show the converse of theorem 2.3. First, we shall introduce the notion of *infinitary* stable families, where configurations need not be finite.

Definition 2.B (infinitary stable family). An *infinitary stable family* \mathcal{F} is a stable family whose configurations may be infinite and such that for every event $e \in \bigcup \mathcal{F}$, there exists a finite subconfiguration x_f such that $e \in x_f$.

Denoting \mathcal{IS} the category of infinitary stable families (with the natural definition of maps of infinitary stable families), the additional condition of finiteness ensures that Pr extends to a functor Pr^∞ from \mathcal{IS} to \mathcal{E} . In fact, theorem 1.7 and its corollary still hold, as proved in [Win83].

Proposition 2.4. $\text{Pr}^\infty: \mathcal{IS} \rightarrow \mathcal{E}$ and $\mathcal{C}^\infty: \mathcal{E} \rightarrow \mathcal{IS}$ are two functors and $\mathcal{C}^\infty \dashv \text{Pr}^\infty$.

The co-unit of the adjunction is the family of morphisms $\text{top}^\infty_{\mathcal{F}}: \mathcal{C}^\infty(\text{Pr}^\infty(\mathcal{F})) \rightarrow \mathcal{F}$, each of them mapping $[e]_x$ to e .

$$\begin{array}{ccc} E & \xrightarrow{f} & \text{Pr}^\infty(\mathcal{F}) \\ \mathcal{C}^\infty(E) & \xrightarrow{\mathcal{C}^\infty(f)} & \mathcal{C}^\infty(\text{Pr}^\infty(\mathcal{F})) \\ & \searrow g & \downarrow \text{top}^\infty_{\mathcal{F}} \\ & & \mathcal{F} \end{array}$$

For the sake of compactness, we shall overload the notations Pr and top .

Notation 2.C. When the context is clear, we shall write Pr instead of Pr^∞ and top instead of top^∞ .

As it happens, theorem 1.8 also remains true.

Theorem 2.5. *Let \mathcal{F} be an infinitary stable family. Then there exists an order isomorphism $\hat{\cdot}: \langle \mathcal{C}^\infty(\text{Pr}(\mathcal{F})), \subseteq \rangle \rightarrow \langle \mathcal{F}, \subseteq \rangle$*

$$\hat{\cdot}: \begin{cases} \mathcal{C}^\infty(\text{Pr}(\mathcal{F})) & \longrightarrow & \mathcal{F} \\ x & \longmapsto & \bigcup x \end{cases}.$$

Its inverse is the morphism $[\cdot]$

$$[\cdot]: \begin{cases} \mathcal{F} & \longrightarrow & \mathcal{C}^\infty(\text{Pr}(\mathcal{F})) \\ x & \longmapsto & \{[e]_x \mid e \in x\} \end{cases}.$$

The following theorem is a corollary of theorem 8.12 from [Win17].

Theorem 2.6. *Let $f: \mathcal{C}^\infty(A) \rightarrow \mathcal{C}^\infty(B)$ be a Berry-stable map between two positive games A and B . Then*

$$\mathcal{F} := \{\bar{x} \parallel y \mid y \in \mathcal{C}^\infty(A^\perp \parallel B) \mid y \subseteq_B f(x)\}$$

is an infinitary stable family and the map $\tilde{f}: \text{Pr}(\mathcal{F}) \rightarrow A^\perp \parallel B$ defined for every $x \in \text{Pr}(\mathcal{F})$ (with polarity inherited from $A^\perp \parallel B$) by $\tilde{f}(x) = \text{top}_{\mathcal{F}}(x)$ is a deterministic strategy from A to B .

Proof. See appendix B. ■

Let us explain a little the heuristic of this theorem. Given a Berry-stable map $f: \mathcal{C}^\infty(A) \rightarrow \mathcal{C}^\infty(B)$, we would like to build a strategy $\tilde{f}: S \rightarrow A^\perp \parallel B$ whose $+$ -maximal configurations are of the form $\bar{x} \parallel f(x)$, so that this construction and that of theorem 2.3 are mutual inverses. The configurations of S are thus $\{\bar{x} \parallel y \mid y \in \mathcal{C}^\infty(A^\perp \parallel B) \mid y \subseteq_B f(x)\}$. The stable family construction is a way to build an event structure out of its configurations via Pr and top .

This construction notably provides a solution to the conjunctive causality problem. Indeed, consider the Berry-stable function $f: \mathcal{C}^\infty(\text{Bool}) \rightarrow \mathcal{C}^\infty(\text{Bool})$ such that $f(b) = \perp$ if $b = \perp$ and $f(b) = \text{True}$ otherwise. Then the output True_2 can be obtained either with input True_1 or input False_1 , which will induce two events $\{\overline{\text{True}}_1, \text{True}_2\}$ and $\{\overline{\text{False}}_1, \text{True}_2\}$ with distinct dependencies. This duplication allows an event to have several different incompatible histories.

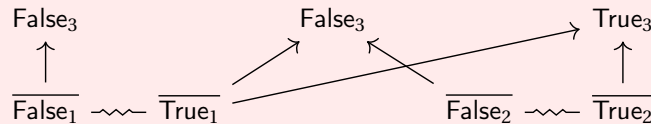
Example 2.iii. Consider \wedge_{left} as defined in example 2.i. The associated stable family is:

$$\begin{aligned} \mathcal{F}_{\text{left}} = \{ & \perp \parallel \perp \parallel \perp, \perp \parallel \overline{\text{False}}_2 \parallel \perp, \perp \parallel \overline{\text{True}}_2 \parallel \perp, \\ & \overline{\text{False}}_1 \parallel \perp \parallel \perp, \overline{\text{False}}_1 \parallel \overline{\text{False}}_2 \parallel \perp, \overline{\text{False}}_1 \parallel \overline{\text{True}}_2 \parallel \perp, \\ & \overline{\text{False}}_1 \parallel \perp \parallel \text{False}_3, \overline{\text{False}}_1 \parallel \overline{\text{False}}_2 \parallel \text{False}_3, \overline{\text{False}}_1 \parallel \overline{\text{True}}_2 \parallel \text{False}_3, \\ & \overline{\text{True}}_1 \parallel \perp \parallel \perp, \overline{\text{True}}_1 \parallel \overline{\text{False}}_2 \parallel \perp, \overline{\text{True}}_1 \parallel \overline{\text{True}}_2 \parallel \perp, \\ & \overline{\text{True}}_1 \parallel \overline{\text{False}}_2 \parallel \text{False}_3, \overline{\text{True}}_1 \parallel \overline{\text{True}}_2 \parallel \text{True}_3 \} \end{aligned}$$

and

$$\text{Pr}(\mathcal{F}_{\text{left}}) = \left\{ \{\overline{\text{False}}_1\}, \{\overline{\text{False}}_2\}, \{\overline{\text{True}}_1\}, \{\overline{\text{True}}_2\}, \{\overline{\text{False}}_1, \text{False}_3\}, \{\overline{\text{True}}_1, \overline{\text{False}}_2, \text{False}_3\}, \{\overline{\text{True}}_1, \overline{\text{True}}_2, \text{True}_3\} \right\}.$$

The strategy $\sigma_{\text{left}}: \text{Pr}(\mathcal{F}_{\text{left}}) \rightarrow (\text{Bool} \parallel \text{Bool})^\perp \parallel \text{Bool}$ induced by \wedge_{left} is defined as follows:



To conclude the proof, one would need to prove that the above correspondence is one-to-one. However, we ran out of time and were not able to prove the following theorem.



Theorem 2.7. *The above correspondence is true, that is, for every Berry-stable map $f: \mathcal{C}^\infty(A) \rightarrow \mathcal{C}^\infty(B)$ and for every deterministic strategy $\sigma: S \rightarrow A^\perp \parallel B$, we have*

$$\tilde{\sigma} = \sigma \quad \text{and} \quad \tilde{\tilde{f}} = f.$$

Example 2.iv. Examples 2.ii and 2.iii illustrate both equations of this theorem.

3 Concurrent games and dialectica categories

The aim of this section is to prove that deterministic winning strategies between two dialectica games are exactly dialectica maps (see 3.2 for explanations).

3.1 Some refinements of concurrent games

We introduce two refinements of concurrent games, namely winning conditions and imperfect information, that will help us define dialectica games.

3.1.1 Games with winning conditions

Here, we want to define what it means to win a concurrent game. The following ideas were first introduced in [CJW12].

Definition 3.A (game with winning conditions). A *game with winning conditions* G comprises a game A along with a set $W_A \subseteq \mathcal{C}^\infty(A)$ of winning configurations for Player.

Winning conditions extend to game operations in the following way:

- *dual*: let $G^\perp := \langle A^\perp, W_{A^\perp} \rangle$ where $W_{A^\perp} := \{x \in \mathcal{C}^\infty(A^\perp) \mid \bar{x} \notin W_A\}$;
- *parallel composition*: for $G = \langle A, W_A \rangle$ and $H = \langle B, W_B \rangle$, let $G \parallel H := \langle A \parallel B, W_A \parallel \mathcal{C}^\infty(B) \cup \mathcal{C}^\infty(A) \parallel W_B \rangle$. A winning configuration on $G \parallel H$ is a configuration that is winning in either G or H .

We want to define winning strategies to be strategies such that whenever Player plays *optimally*, he ends in a winning configuration. Here, optimally means that Player cannot make any other move, which leads to the following definitions.

Definition 3.B (+-maximal configuration). Let S be an event structure with polarities. A +-*maximal configuration* of S is a configuration $x \in \mathcal{C}^\infty(S)$ such that, for every $s \in S$, $x \xrightarrow{s} \text{ implies } \text{pol}_S(s) = -$.

Definition 3.C (winning strategy). A *winning strategy* on a game with winning conditions $G = \langle A, W_A \rangle$ is a strategy $\sigma: S \rightarrow A$ on A such that for every +-maximal configuration $x \in \mathcal{C}^\infty(S)$, $\sigma[x] \in W_A$.

A winning strategy prescribes moves so that Player always ends in a winning condition, no matter what Opponent does. This is why we consider only +-maximal configurations.

Remark. The same work could have been done with losing conditions. Indeed, define L_A to be $\mathcal{C}^\infty(A) \setminus W_A$ the set of losing conditions, then a winning strategy on $\langle A, W_A \rangle$ is a strategy σ such that the image of any +-maximal configuration x is non-losing, i.e. $\sigma[x] \notin L_A$.

Example 3.i. In Rock-Paper-Scissors defined in example 1.vii with usual rules, the winning configurations are $\{\text{Rock}_1^+, \text{Scissors}_2^-\}$, $\{\text{Paper}_1^+, \text{Rock}_2^-\}$ and $\{\text{Scissors}_1^+, \text{Paper}_2^-\}$. The strategy defined in example 1.ix is winning.

3.1.2 Games with imperfect information

The strategy on Rock-Paper-Scissors described in example 1.ix illustrates the need for games where players must play *independently*. The strategy from example 1.ix may be winning, but it makes Player's moves depend on Opponent's, which is something we want to avoid, since the two players are supposed to play simultaneously. Imperfect information is a way of forcing independence by masking some moves.

To extend games to games with imperfect information, we supposed a fixed preorder of *levels* (Λ, \leq) . The following ideas were first introduced in [Win12].

Definition 3.D (game with imperfect info). A *game with imperfect information* (or Λ -*game*) $\langle G, l \rangle$ comprises a game with winning conditions $G = \langle A, W_A \rangle$ together with a *level function* $l: A \rightarrow \Lambda$ such that

$$\forall a, a' \in A. a \leq_A a' \implies l(a) \leq l(a').$$

Imperfect information extends to game operations in the following way:

- *dual*: let $G^\perp := \langle A^\perp, l_{A^\perp} \rangle$ where $l_{A^\perp}(\bar{a}) := l_A(a)$ for every event $a \in A$;
- *parallel composition*: let $G = \langle A, l_A \rangle$ and $H = \langle B, l_B \rangle$, we define $G \parallel H := \langle A \parallel B, l_{A \parallel B} \rangle$ where $l_{A \parallel B}((1, a)) := l_A(a)$ and $l_{A \parallel B}((2, b)) := l_B(b)$ for every events $a \in A$ and $b \in B$.

We also add constraints to strategies in the following way. The point is that strategies with dependencies on masked moves are no longer considered.

Definition 3.E (Λ -strategy). Given a game with imperfect information $\langle G, l \rangle$, a Λ -strategy (or *strategy of game with imperfect information*) on G is a strategy $\sigma: S \rightarrow A$ such that

$$\forall s, s' \in S \cdot s \leq_S s' \implies l(\sigma(s)) \leq l(\sigma(s')). \quad (1)$$

Levels are to be thought of as permission. Moves in games and strategies are to respect levels: moves will be assigned levels in such a way that a move is only permitted to causally depend on moves at equal or lower levels; it is as if from a level only moves of equal or lower level can be seen.

Example 3.ii. So as to accurately model the game Rock-Paper-Scissors, it is necessary to use imperfect information. It can be represented by the event structure with polarities defined in example 1.vii, and the imperfect information is added as follows: we let $\Lambda := \{1, 2\}$ and $< := \emptyset$, positive events are taken to level 1 and negative ones to level 2. These settings prevents the cheaty strategy defined in example 1.ix from being valid.

Remark. In the above example, we could as well have let $1 < 2$ in Λ . Imperfect information would have indeed prevented $\ominus \rightarrow \oplus$ dependencies in S , and undesirable $\oplus \rightarrow \ominus$ are anyway disabled by innocence.

3.2 Dialectica games

In 1958, Kurt Gödel published a way to reduce Heyting Arithmetics to a functional theory with finite types [Gö58]. This method, combined with the previously known double negation interpretation, allows one to reduce both intuitionistic and classical theories to functional theories with finite types. This reduction is called the *Dialectica interpretation*, it is presented in details in [AF98]. This interpretation is to be thought of as a kind of Curry-Howard isomorphism with another, maybe closer to mathematics, functional theory than λ -calculus. In 1991, Valeria de Paiva and Martin Hyland showed a connexion between Girard's linear logic and the so-called Dialectica Categories [CVdP91], following the lines of Lambek's correspondence between classical logic and cartesian closed categories. The category of dialectica games is the dialectica category associated with the category of event structures with polarities and games with imperfect information. As it happens, the category of dialectica games is a full subcategory of the category of deterministic concurrent strategies (see proposition 3.2).

Definition 3.F (dialectica game). A *dialectica game* A is a game with imperfect information $A_p \parallel A_n$ where all events of A_p (resp. A_n) have polarity $+$ (resp. $-$) and

$$\forall a, a' \in A \cdot l_A(a) < l_A(a') \iff a \in A_p \wedge a' \in A_n.$$

By letting $\Lambda = \{1, 2\}$ and $1 < 2$, this means that $l_A(a) = 1$ for every $a \in A_p$ and $l_A(a') = 2$ for every $a' \in A_n$.

These games therefore comprise a game A_p for Player in parallel with a game A_n for Opponent where Opponent can see Player's moves but not the converse.

By adapting the definition of [CVdP91], we get the following definition of dialectica maps.

Definition 3.G (dialectica map). A *dialectica map* between two dialectica games A and B is a pair of Berry-stable maps $\langle f, g \rangle$ with $f: \mathcal{C}^\infty(A_p) \rightarrow \mathcal{C}^\infty(B_p)$ and $g: \mathcal{C}^\infty(A_p \parallel B_n) \rightarrow \mathcal{C}^\infty(A_n)$ such that for all $x \in \mathcal{C}^\infty(A_p)$ and $y \in \mathcal{C}^\infty(B_n)$,

$$x \parallel g(x \parallel y) \in W_A \implies f(x) \parallel y \in W_B.$$

3.3 Correspondence between deterministic winning strategies and dialectica maps

3.3.1 From strategies to dialectica maps

First, given a deterministic strategy $\sigma: S \rightarrow A^\perp \parallel B$, let us characterise the $+$ -maximal configurations of S thanks to proposition 2.3.

Lemma 3.1. *Let $\sigma: S \rightarrow A^\perp \parallel B$ be a deterministic strategy from A to B with A and B purely positive. Then the set $\mathcal{C}_{+-\max}^\infty(S)$ of $+$ -maximal configurations of S is in bijection with $\mathcal{C}^\infty(A)$. More specifically, the map θ*

$$\theta: \begin{cases} \mathcal{C}^\infty(A) & \longrightarrow & \mathcal{C}_{+-\max}^\infty(S) \\ x & \longmapsto & z \text{ such that } \sigma[z] = \bar{x} \parallel \bar{\sigma}(x) \end{cases}$$

is a bijection.

Proof. See appendix C. ■

This lemma will be useful to prove the following proposition.

Proposition 3.2. *Let A and B be two dialectica games and $\sigma: S \rightarrow A^\perp \parallel B$ be a deterministic winning strategy from A to B . Then σ induces a dialectica map $\langle f, g \rangle$ with $f: \mathcal{C}^\infty(A_p) \rightarrow \mathcal{C}^\infty(B_p)$ and $g: \mathcal{C}^\infty(A_p \parallel B_n) \rightarrow \mathcal{C}^\infty(A_n)$.*

Proof. See appendix C. ■

Notation 3.H. We shall denote by $\text{stab}(\sigma)$ the Bery-stable map $h: \mathcal{C}^\infty(A_p \parallel B_n) \rightarrow \mathcal{C}^\infty(B_p) \times \mathcal{C}^\infty(A_n)$ defined in the proof of proposition 3.2.

3.3.2 From dialectica maps to strategies

We now show the converse.

Proposition 3.3. *Let A and B be two dialectica games and let $\langle f, g \rangle$ be a dialectica map with $f: \mathcal{C}^\infty(A_p) \rightarrow \mathcal{C}^\infty(B_p)$ and $g: \mathcal{C}^\infty(A_p \parallel B_n) \rightarrow \mathcal{C}^\infty(A_n)$. Then $\langle f, g \rangle$ induces a deterministic winning strategy $\sigma: S \rightarrow A^\perp \parallel B$ from A to B .*

Proof (partial). See appendix C. ■

Finally, We need to prove that the correspondence is one-to-one. To do so, we start by proving the following theorem.

Theorem 3.4. *Let $\sigma: S \rightarrow A^\perp \parallel B$ be a deterministic strategy from A to B where A and B are dialectica games. Let us denote $\phi := \text{stab}(\sigma)$. Then, for every $x \in \mathcal{C}^\infty(A_p)$ and $y, y' \in \mathcal{C}^\infty(B_n)$, we have $\pi_1(\phi(x \parallel y)) = \pi_1(\phi(x \parallel y'))$. In particular, the transformation $\text{stab}(\sigma) \mapsto \langle f, g \rangle$ described in the proof of proposition 3.2 is injective.*

Proof. See appendix C. ■

Together with theorem 2.7, the previous theorem proves that the correspondence is indeed one-to-one.

Corollary 3.4.1. *The two constructions from propositions 3.2 and 3.3 are mutual inverses.*

Proof. See appendix C. ■

Conclusion

Contributions of the internship

This internship enabled me to discover a whole facet of research in theoretical computer science: concurrent games. I followed last year an introductory course in semantics and wanted to learn more about it, which is why I chose to do an internship on a related topic. It also allowed me to continue my formation in category theory, for event structures have a very strong categorical structure. Finally, I discovered during this internship the very peculiar feeling of spending a (very, very) long time trying to prove a result, and being told after three days that I am not looking in the right direction; but I also discovered the just as special – but much more rewarding – feeling of finally understanding a thorny concept or finishing a long and complicated proof, which makes research so engaging an activity.

The work I have done in this internship was of two different kinds: I spent the first months trying to get familiar with the vast world of event structures, and build an intuition of the brand new objects I was dealing with. Once this work was (partially) done, I started focusing on results my supervisors asked me to prove. The process of proving these results was very long and help from my supervisors was often needed, as I did not have at first all the background needed to step back and have a global vision of the work being done. Nonetheless, I managed to finish almost all the proofs I initiated, although time prevented me to finish them all.

Regarding technical contribution, the core of my work lies in sections 2 and 3. Section 2 is a special case of chapter 8 from [Win17], but I made the proofs shorter and more readable in the case I was considering. Section 3 is new but the idea is due to Glynn Winskel, my main contribution certainly is to have dealt with technical details and subtelties. As far as section 1 is concerned, it is meant to be a crash course in concurrent games and event structures, I therefore spent some time trying to present ideas and concepts in the most natural and understandable way, which can also be considered a (pedagogical) contribution of mine.

Possible continuations

The first and obvious continuation of this internship would be to finish the proofs I did not have time to complete. The facts that were left unproved in this report are recapitulated hereafter.

Admitted hypothesis 1. Theorem 2.7.

Admitted hypothesis 2. In the proof of proposition 3.3, the fact that σ preserves the structure of imperfect information.

Admitted hypothesis 3. In the proof of proposition 3.3, the fact that σ is winning.

So as to broaden the scope of concurrent games, it may be interesting to prove equivalences between games and other structures. For instance, if we consider strategies from section 3 without imperfect information, one could prove the equivalence with Jean-Yves Girard's geometry of interaction.

References

- [AF98] Jeremy Avigad and Solomon Feferman. Chapter v - gödel's functional ("dialectica") interpretation. In Samuel Buss, editor, *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 337–405. Elsevier, 1998. <https://www.andrew.cmu.edu/user/avigad/Papers/dialect.pdf>.
- [AM99] Samson Abramsky and Paul-André Melliès. Concurrent games and full completeness. In *LICS*, pages 431–442, 1999. <https://www.cs.ox.ac.uk/files/316/cg.pdf>.
- [Ber78] Gérard Berry. Stable models of typed λ -calculi. In Giorgio Ausiello and Corrado Böhm, editors, *Automata, Languages and Programming*, pages 72–89. Springer, 1978.
- [CCRW17] Simon Castellan, Pierre Clairembault, Sylvain Rideau, and Glynn Winskel. Games and strategies as event structures. *Logical Methods in Computer Science*, Volume 13, Issue 3, September 2017. <https://lmcs.episciences.org/3966>.
- [CJW12] Pierre Clairambault, Gutierrez Julian, and Glynn Winskel. The winning ways of concurrent games. In *LICS*, 2012. https://www.cl.cam.ac.uk/~gw104/lics2012_submission_14.pdf.
- [CVdP91] Valeria Correa Vaz de Paiva. The dialectica categories. Technical Report 213, University of Cambridge, January 1991. <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-213.pdf>.
- [FP09] Claudia Faggian and Mauro Piccolo. Partial orders, event structures and linear strategies. In *Lecture Notes in Computer Science*, volume 5608, pages 95–111. Springer, 2009. <https://www.irif.fr/~faggian/pubs/es09.pdf>.
- [Gö58] Kurt Gödel. Über eine bisher noch nicht benützte erweiterung des finiten standpunktes. *Dialectica*, Volume 12:241–251, 1958.
- [Joy77] André Joyal. Remarques sur la théorie des jeux à deux personnes. *Gazette des sciences mathématiques du Québec*, Volume 1, Issue 4, 1977.
- [ML98] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.
- [MM07] Paul-André Melliès and Samuel Mimram. Asynchronous games: Innocence without alternation. In *Lecture Notes in Computer Science*, volume 4703, pages 395–411. Springer, 2007. <https://hal.archives-ouvertes.fr/hal-00152707/document>.
- [Win80] Glynn Winskel. *Events in Computation*. PhD thesis, University of Edinburgh, 1980. <https://www.cl.cam.ac.uk/~gw104/Winskel1980.Pdf>.
- [Win83] Glynn Winskel. Event structure semantics for ccs and related languages. Technical report, University of Aarhus, April 1983. <https://www.cl.cam.ac.uk/~gw104/eventStructures82.pdf>.
- [Win93] Glynn Winskel. *The Formal Semantics of Programming Languages: an Introduction*. MIT Press, 1993.
- [Win12] Glynn Winskel. Winning, losing and drawing in concurrent games with perfect or imperfect information. In *Lecture Notes in Computer Science*, volume 7230. Springer, 2012. <https://www.cl.cam.ac.uk/~gw104/Dexter.pdf>.
- [Win17] Glynn Winskel. Event structures, stable families and concurrent games. <https://www.cl.cam.ac.uk/~gw104/ecsym-notes.pdf>, February 2017.

A The basics of category theory

In this section, we recall basic definitions of category theory. See [ML98] for more details.

Definition A.A (category). A *category* \mathcal{A} consists of:

- a class of *objects* $\text{ob}(\mathcal{A})$,
- for each $A, B \in \text{ob}(\mathcal{A})$, a class $\text{hom}_{\mathcal{A}}(A, B)$ of *morphisms* from A to B ,
- for each $A, B, C \in \text{ob}(\mathcal{A})$, a function

$$\circ: \begin{cases} \text{hom}_{\mathcal{A}}(B, C) \times \text{hom}_{\mathcal{A}}(A, B) & \longrightarrow & \text{hom}_{\mathcal{A}}(A, C) \\ \langle g, f \rangle & \longmapsto & g \circ f \end{cases}$$

called *composition*,

- for each $A \in \text{ob}(\mathcal{A})$, an element $\text{id}_A \in \text{hom}_{\mathcal{A}}(A, A)$ called *identity* on A ;

satisfying the following axioms:

- (i) for every $f \in \text{hom}_{\mathcal{A}}(A, B)$, $g \in \text{hom}_{\mathcal{A}}(B, C)$ and $h \in \text{hom}_{\mathcal{A}}(C, D)$, we have $(h \circ g) \circ f = h \circ (g \circ f)$, *associativity*
- (ii) for every $f \in \text{hom}_{\mathcal{A}}(A, B)$, we have $f \circ \text{id}_A = \text{id}_B \circ f = f$. *identity laws*

Definition A.B (opposite category). The *opposite category* \mathcal{A}^{op} of \mathcal{A} is the category obtained from \mathcal{A} by reversing the arrows: $\text{ob}(\mathcal{A}^{\text{op}}) := \text{ob}(\mathcal{A})$ and, for every $A, B \in \text{ob}(\mathcal{A})$, $\text{hom}_{\mathcal{A}^{\text{op}}}(A, B) := \text{hom}_{\mathcal{A}}(B, A)$.

Remark. Identities in \mathcal{A}^{op} are the same as in \mathcal{A} . Composition in \mathcal{A}^{op} is the same as in \mathcal{A} but with the arguments reversed: if $A \xrightarrow{f} B \xrightarrow{g} C$ in \mathcal{A}^{op} then $A \xleftarrow{f} B \xleftarrow{g} C$ in \mathcal{A} that is $A \xleftarrow{f \circ g} C$ in \mathcal{A} .

As for every algebraic structure, we need to define how categories interact with each other. Functors are to be thought of as morphisms of categories.

Definition A.C (functor). Let \mathcal{A} and \mathcal{B} be two categories. A (*covariant*) *functor* $F: \mathcal{A} \rightarrow \mathcal{B}$ consists of:

- a function

$$\text{ob}(\mathcal{A}) \rightarrow \text{ob}(\mathcal{B})$$

written as $A \mapsto F(A)$,

- for each $A, A' \in \text{ob}(\mathcal{A})$, a function

$$\text{hom}_{\mathcal{A}}(A, A') \rightarrow \text{hom}_{\mathcal{B}}(F(A), F(A'))$$

written as $f \mapsto F(f)$;

satisfying the following axioms:

- (i) $F(f' \circ f) = F(f') \circ F(f)$ whenever $A \xrightarrow{f} A' \xrightarrow{f'} A''$ in \mathcal{A} ,
- (ii) $F(\text{id}_A) = \text{id}_{F(A)}$ for every $A \in \text{ob}(\mathcal{A})$.

A *contravariant functor* from \mathcal{A} to \mathcal{B} is a functor $\mathcal{A}^{\text{op}} \rightarrow \mathcal{B}$.

Example A.i. Let \mathcal{A} , \mathcal{B} and \mathcal{C} be three categories.

- The identity functor $\text{Id}_{\mathcal{A}}: \mathcal{A} \rightarrow \mathcal{A}$ behaves like identity on objects and on morphisms of \mathcal{A} .
- If $F: \mathcal{B} \rightarrow \mathcal{C}$ and $G: \mathcal{A} \rightarrow \mathcal{B}$ are two functors, we can naturally define $F \circ G$ which still is a functor.

The generality of categories now allows us to define morphisms of functors.

Definition A.D (natural transformation). Let \mathcal{A} and \mathcal{B} be two categories and $F: \mathcal{A} \rightarrow \mathcal{B}$ and $G: \mathcal{B} \rightarrow \mathcal{A}$ be two functors. A *natural transformation* $\alpha: F \rightarrow G$ from F to G is a family

$$(\alpha_A: F(A) \rightarrow G(A))_{A \in \text{ob}(\mathcal{A})}$$

of maps in \mathcal{B} such that, for every $f \in \text{hom}_{\mathcal{A}}(A, A')$, the following diagram commutes:

$$\begin{array}{ccc} F(A) & \xrightarrow{F(f)} & F(A') \\ \alpha_A \downarrow & & \downarrow \alpha_{A'} \\ G(A) & \xrightarrow{G(f)} & G(A') \end{array}$$

Finally, we define a kind of inverse for functors.

Definition A.E (adjunction). Let \mathcal{A} and \mathcal{B} be two categories and $L: \mathcal{A} \rightarrow \mathcal{B}$ and $R: \mathcal{B} \rightarrow \mathcal{A}$ be two functors. We say that L is *left adjoint* to R , and R is *right adjoint* to L , and write $L \dashv R$ iff, for every $A \in \text{ob}(\mathcal{A})$ and $B \in \text{ob}(\mathcal{B})$, there exists a bijection

$$\phi_{A,B}: \text{hom}_{\mathcal{B}}(L(A), B) \cong \text{hom}_{\mathcal{A}}(A, R(B))$$

such that $\phi_{A,B}$ transforms every commutative diagram

$$\begin{array}{ccc} L(A) & \xrightarrow{g} & B \\ L(h_A) \uparrow & & \downarrow h_B \\ L(A') & \xrightarrow{f} & B' \end{array}$$

into a commutative diagram

$$\begin{array}{ccc} A & \xrightarrow{\phi_{A,B}(g)} & R(B) \\ h_A \uparrow & & \downarrow R(h_B) \\ A' & \xrightarrow{\phi_{A',B'}(f)} & R(B') \end{array}$$

Units and co-units are defined in the following proposition.

Proposition A.1. • In the previous definition, if we let $B := L(A)$ and $g := \text{id}_{L(A)}$, the image η_A of g by $\phi_{A,L(A)}$ is a morphism from A to $R(L(A))$. The family $(\eta_A)_{A \in \text{ob}(\mathcal{A})}$ defines a natural transformation η from the identity functor $\text{Id}_{\mathcal{A}}$ to the functor $R \circ L$. This transformation is called *unit* of the adjunction $L \dashv R$.

- Similarly, if we let $A := R(B)$, the preimage ε_B of $\text{id}_{R(B)}$ by $\phi_{R(B),B}$ is a morphism from $L(R(B))$ to B . The family $(\varepsilon_B)_{B \in \text{ob}(\mathcal{B})}$ defines a natural transformation ε from the functor $L \circ R$ to $\text{Id}_{\mathcal{B}}$. This transformation is called *co-unit* of the adjunction $L \dashv R$.

Remark. Unit and co-unit allow to rebuild bijections $\phi_{A,B}$: for every morphism $g: L(A) \rightarrow B$, $\phi_{A,B}(g) = R(g) \circ \eta_A$ and, for every morphism $h: A \rightarrow R(B)$, $\phi_{A,B}(h) = \varepsilon_B \circ L(h)$.

B Proofs of section 2

Proof of lemma 2.1. Let us denote $Z = (z_i)_{i \in I}$ and, for every $i \in I$, $z_i = \bigcup \{z_i^\circ \mid z_i^\circ \in \mathcal{C}(A) \wedge z_i^\circ \subseteq z_i\}$ the (directed) union of its finite subconfigurations. For the sake of clarity, we shall also denote $z_i = \bigcup \{z_i^j \mid j \in J_i\}$ the previous union. Then

$$\begin{aligned}
 f(\bigcap Z) &= f\left(\bigcap_{i \in I} \bigcup_{j \in J_i} z_i^j\right) \\
 &= f\left(\bigcup_{\theta \in \prod_{i \in I} J_i} \bigcap_{i \in I} z_i^{\theta(i)}\right) && \text{by the axiom of choice,} \\
 &= \bigcup_{\theta \in \prod_{i \in I} J_i} f\left(\bigcap_{i \in I} z_i^{\theta(i)}\right) && \text{by Scott-continuity,} \\
 &= \bigcup_{\theta \in \prod_{i \in I} J_i} \bigcap_{i \in I} f(z_i^{\theta(i)}) && \text{generalising condition (ii) by induction,} \\
 &= \bigcap_{i \in I} \bigcup_{j \in J_i} f(z_i^j) \\
 &= \bigcap f(Z).
 \end{aligned}$$

■

Proof of theorem 2.3. First notice that $\tilde{\sigma}$ is a total map. Indeed, let $x \in \mathcal{C}^\infty(A)$, then $\emptyset \subseteq^- \bar{x}$ i.e. $\sigma[\emptyset] \subseteq^- \bar{x}$; thus, by proposition 1.10, there exists some $z \in \mathcal{C}^\infty(S)$ such that $\sigma[z] = \bar{x}$, which ensures the existence of $\tilde{\sigma}(x)$.

In addition, if $x \in \mathcal{C}^\infty(A)$ we shall make sure that $\tilde{\sigma}(x) \in \mathcal{C}^\infty(B)$. It is clear that $\tilde{\sigma}(x)$ is downward-closed, since it is the union of downward-closed sets, and that $\tilde{\sigma}(x) \subseteq B$.

Let us also prove that $\tilde{\sigma}(x)$ is consistent. Let X be a finite subset of $\tilde{\sigma}(x)$. If X is such that there exists some $z \in S$ such that $\sigma[z] = \bar{x} \parallel y$ and $X \subseteq y$ then X is obviously consistent. Else, w.l.o.g. we can suppose that $X = y \cup y'$ for some $y, y' \in \mathcal{C}^\infty(B)$ such that $\sigma[z] = \bar{x} \parallel y$ and $\sigma[z'] = \bar{x} \parallel y'$ with $z, z' \in \mathcal{C}^\infty(S)$. By proposition 1.10 applied to the empty configuration in S , there exists a unique $z_- \in \mathcal{C}^\infty(S)$ such that $\sigma[z_-] = \bar{x}$. In addition, taking $z_+ := z \setminus z_-$ and $z'_+ := z' \setminus z_-$, local injectivity of σ ensures that $\sigma[z_+] = y$ and $\sigma[z'_+] = y'$. As σ reflects polarity, we must have $\text{Neg}[z_+ \cup z'_+] \subseteq z_-$ and, since the latter is consistent, $z_+ \cup z'_+$ must be consistent too by determinism of σ . Hence $[z_+] \cup [z'_+] = [z_+ \cup z'_+] \in \mathcal{C}^\infty(S)$ and thus $\sigma[[z_+]] \cup \sigma[[z'_+]] = \sigma[[z_+ \cup z'_+]] \in \mathcal{C}^\infty(A^\perp \parallel B)$. In particular, since $y, y' \subseteq \sigma[[z_+]] \cup \sigma[[z'_+]]$, we have $y \uparrow y'$, yielding the consistency of X .

► We shall now verify that $\tilde{\sigma}$ is Berry-stable.

$\tilde{\sigma}$ is monotonic. Let $x, x' \in \mathcal{C}^\infty(A)$ such that $x' \subseteq x$. Assume there exists some $z' \in \mathcal{C}^\infty(S)$ such that $\sigma[z'] = \bar{x}' \parallel y$. Then $\bar{x} \parallel y$ also is a configuration of $A^\perp \parallel B$ and $\sigma[z'] \subseteq^- \bar{x} \parallel y$. Hence, by proposition 1.10, there exists some configuration $z \supseteq z'$ such that $\sigma[z] = \bar{x} \parallel y$. Therefore

$$\{y \mid \exists z \in \mathcal{C}^\infty(S) \cdot \bar{x}' \parallel y = \sigma[z]\} \subseteq \{y \mid \exists z \in \mathcal{C}^\infty(S) \cdot \bar{x} \parallel y = \sigma[z]\},$$

which proves that $\tilde{\sigma}(x') \subseteq \tilde{\sigma}(x)$.

$\tilde{\sigma}$ preserves the supremum of directed families. Let $\{x_i \mid i \in I\}$ be a directed family in $\mathcal{C}^\infty(A)$ which has a supremum $\bigvee_{i \in I} x_i$. Let us prove that $\bigcup_{i \in I} \tilde{\sigma}(x_i) = \tilde{\sigma}(\bigcup_{i \in I} x_i)$.

≤ By monotonicity, we have $\tilde{\sigma}(x_i) \subseteq \tilde{\sigma}(\bigcup_{i \in I} x_i)$ for every $i \in I$. Thus $\bigcup_{i \in I} \tilde{\sigma}(x_i) \subseteq \tilde{\sigma}(\bigcup_{i \in I} x_i)$.

≥ Let $e \in \tilde{\sigma}(\bigcup_{i \in I} x_i)$. We are to prove that there exists some $i \in I$ and $z' \in \mathcal{C}^\infty(S)$ such that $\sigma[z'] = \bar{x}_i \parallel [e]$, which will yield that $e \in \tilde{\sigma}(x_i)$.

Let us denote by z_e the unique subset of z (which is not necessarily a configuration) such that $\sigma[z_e] = [e]$. Then z_e is finite and so is its downward-closure; thus, by local injectivity, we can consider $i_0 \in I$ such that $\sigma[[z_e]]^- \subseteq x_{i_0}$, where $\sigma[[z_e]]^-$ denotes the projection of $\sigma[[z_e]]$ to its negative events. Proposition 1.10 ensures the existence of a unique $z_{i_0} \in \mathcal{C}^\infty(S)$ such that $\sigma[z_{i_0}] = x_{i_0}$ thus, by letting $z' := z_{i_0} \cup [z_e] \in \mathcal{C}^\infty(S)$, it follows that $\sigma[z'] = x_{i_0} \parallel [e]$, which ends the proof.

For every $x_1, x_2 \in \mathcal{C}^\infty(A)$ such that $x_1 \uparrow x_2$ we have $\tilde{\sigma}(x_1 \cap x_2) = \tilde{\sigma}(x_1) \cap \tilde{\sigma}(x_2)$. Let $x_1, x_2 \in \mathcal{C}^\infty(A)$ such that $x_1 \uparrow x_2$.

≤ By monotonicity we have $\tilde{\sigma}(x_1 \cap x_2) \subseteq \tilde{\sigma}(x_1)$ and $\tilde{\sigma}(x_1 \cap x_2) \subseteq \tilde{\sigma}(x_2)$, hence $\tilde{\sigma}(x_1 \cap x_2) \subseteq \tilde{\sigma}(x_1) \cap \tilde{\sigma}(x_2)$.

≥ Suppose $e \in \tilde{\sigma}(x_1) \cap \tilde{\sigma}(x_2)$. Then there exists some $z_1, z_2 \in \mathcal{C}^\infty(S)$ such that $\sigma[z_1] = \bar{x}_1 \parallel y_1$ and $\sigma[z_2] = \bar{x}_2 \parallel y_2$ with $e \in y_1$ and $e \in y_2$. We want to show that there exists some $z \in \mathcal{C}^\infty(S)$ such that $\sigma[z] = \bar{x}_1 \cap \bar{x}_2 \parallel y_1 \cap y_2$, which will prove that $e \in \tilde{\sigma}(x_1 \cap x_2)$.

Let z_1^- (resp. z_2^-) be the projection of z_1 (resp. z_2) to its negative events. These are configurations of S by innocence since $A^\perp \parallel B$ stipulates no causal dependency from positive to negative events. Then $z_1^-, z_2^- \in \mathcal{C}^\infty(S)$ are only made up of negative events and $\sigma[z_1^-] = x_1 \uparrow^- x_2 = \sigma[z_2^-]$, hence $z_1^- \uparrow z_2^-$ by corollary 1.10.1. Since S is deterministic, we can conclude that $z_1 \uparrow z_2$. Thus, by lemma 2.2, we have $\sigma[z] = \bar{x}_1 \cap \bar{x}_2 \parallel y_1 \cap y_2$ with $z := z_1 \cap z_2$.

■

Proof of theorem 2.6. First notice that is $[e]_x \in \text{Pr}(\mathcal{F})$ then $\tilde{f}([e]_x) = e$ is indeed an event of $A^\perp \parallel B$, hence \tilde{f} is well defined (provided that \mathcal{F} actually is a stable family).

► *Let us start by proving that \mathcal{F} is a stable family.*

Finiteness. Let $\bar{x} \parallel y \in \mathcal{F}$ and $e \in \bar{x} \parallel y$.

If $\text{pol}(e) = -$ then $[e]$ is a finite configuration of $A^\perp \parallel B$ which obviously belongs to \mathcal{F} .

If $\text{pol}(e) = +$ let us denote $x = \bigcup \{x^\circ \mid x^\circ \in \mathcal{C}(A) \wedge x^\circ \subseteq x\}$ the directed union of its finite subconfigurations. Then, by Scott-continuity, $f(x) = \bigcup \{f(x^\circ) \mid x^\circ \in \mathcal{C}(A) \wedge x^\circ \subseteq x\}$ hence, since $e \in y \subseteq f(x)$, there exists some finite subconfiguration x° of x such that $e \in f(x^\circ)$. Thus $[e] \subseteq f(x^\circ)$, and $\bar{x}^\circ \parallel [e]$ is a finite configuration of $A^\perp \parallel B$ that contains e . Therefore $\bar{x}^\circ \parallel [e]$ suits to the definition of the finiteness property.

Completeness. Let $Z \subseteq \mathcal{F}$ such that $Z \uparrow$ and let $x \in \mathcal{F}$ given by notation 1.K. Since $\bigcup Z \subseteq x$, it is consistent and $\bigcup Z$ is downward-closed as the union of downward-closed sets. Hence $\bigcup Z \in \mathcal{C}^\infty(A^\perp \parallel B)$.

Let us write $\bigcup Z := \bar{x} \parallel y$. It remains to be proved that $y \subseteq f(x)$. For this matter, let us denote $Z = (z_i)_{i \in I}$ and, for every $i \in I$, $z_i = \bar{x}_i \parallel y_i$. Then $x = \bigcup_{i \in I} x_i$ and $y = \bigcup_{i \in I} y_i$. It follows that $f(x) = \bigcup_{i \in I} f(x_i)$ and, since $y_i \subseteq f(x_i)$ for every $i \in I$, $y \subseteq f(x)$.

Stability. Let $Z = (z_i)_{i \in I}$ be a non-empty family of elements of \mathcal{F} such that $Z \uparrow$ and let $x \in \mathcal{F}$ given by notation 1.K. Then $\bigcap Z \subseteq x$ hence it is consistent and $\bigcap Z$ is downward-closed as the intersection of downward-closed sets. Thus $\bigcap Z \in \mathcal{C}^\infty(A^\perp \parallel B)$.

Now prove that $y \subseteq f(x)$ where x and y are defined by $\bigcap Z = \bar{x} \parallel y$. Denoting $z_i = \bar{x}_i \parallel y_i$ for all $i \in I$, we have $x = \bigcap_{i \in I} x_i$ and $y = \bigcap_{i \in I} y_i$. By lemma 2.1, we have $f(x) = \bigcap_{i \in I} f(x_i)$ hence $y \subseteq f(x)$ since $y_i \subseteq f(x_i)$ for every $i \in I$.

Coincidence-freeness. Let $z = \bar{x}_z \parallel y_z \in \mathcal{F}$ and e, e' be two distinct events of z . We shall discuss two cases.

- Either $\text{pol}(e) \neq \text{pol}(e')$ and w.l.o.g. $\text{pol}(e) = -$, and then $y = [e] \in \mathcal{F}$ suits;
- or $\text{pol}(e) = \text{pol}(e')$.
 - If both events have polarity $-$ then take y to be $[e]$ if $e' \not\leq_E e$ and $[e']$ otherwise. Noticing that if $e' \leq_E e$ it cannot be that $e \leq_E e'$ since $e \neq e'$, one can easily see that $y \in \mathcal{F}$ suits.
 - If both events have polarity $+$ then take y to be $[e] \parallel y_z$ if $e' \not\leq_E e$ and $[e'] \parallel y_z$ otherwise. Here again, $y \in \mathcal{F}$ and thus y suits.

► *We now prove that \tilde{f} is a deterministic strategy.*

As an intermediate result, we shall prove that for every $x \in \mathcal{C}^\infty(\text{Pr}(\mathcal{F}))$

$$\text{top}_{\mathcal{F}}[x] = \hat{x} \quad \text{i.e.} \quad \tilde{f}[x] = \hat{x}. \quad (2)$$

As $\tilde{f}[x] = \{e \mid \exists x' \in \mathcal{F} \cdot [e]_{x'} \in x\}$ the direct inclusion is clear. Now, if $e \in \hat{x}$, there exists some $[e']_{x'} \in x$ such that $e \in [e']_{x'}$. Since x is downward-closed we must have $[e]_{x'} \in x$ because $[e]_{x'} \subseteq_{\text{Pr}(\mathcal{F})} [e']_{x'}$. Hence $e = \tilde{f}([e]_{x'}) \in \tilde{f}[x]$, which completes the proof.

\tilde{f} is a strategy. We shall use proposition 1.10 here. Let $x \in \mathcal{C}^\infty(\text{Pr}(\mathcal{F}))$ and $y \in \mathcal{C}^\infty(A^\perp \parallel B)$.

Suppose $y \subseteq^+ \tilde{f}[x]$. Then it is clear that $y \in \mathcal{F}$. Let $x' := [y] \in \mathcal{C}^\infty(\text{Pr}(\mathcal{F}))$. Then $x' \subseteq x$ since $y = \hat{x}' \subseteq \hat{x} = \tilde{f}[x]$ by equation 2; and $\tilde{f}[x'] = \hat{x}' = y$.

Now suppose $\tilde{f}[x] \subseteq^- y$. Then monotonicity of f ensures that $y \in \mathcal{F}$. Let $x' := [y] \in \mathcal{C}^\infty(\text{Pr}(\mathcal{F}))$. Then $x \subseteq x'$ since, by equation 2, $\tilde{f}[x] = \hat{x} \subseteq \hat{x}' = y$; and $\tilde{f}[x'] = y$.

Uniqueness of x' is a consequence of injectivity of $\hat{\cdot}$ and equation 2.

$\text{Pr}(\mathcal{F})$ is deterministic. Let us use proposition 1.11 and take $z \in \mathcal{C}(\text{Pr}(\mathcal{F}))$ and $[e_1]_{z_1}, [e_2]_{z_2}$ two events of $\text{Pr}(\mathcal{F})$ such that $\text{pol}_{A^\perp \parallel B}(e_1) = +$ and $z \xrightarrow{[e_1]_{z_1}} \hat{z}$ and $z \xrightarrow{[e_2]_{z_2}} \hat{z}$. Let us prove that $\hat{z}' \cup \hat{z}'' \in \mathcal{F}$, which will yield $z' \cup z'' \in \mathcal{C}^\infty(\text{Pr}(\mathcal{F}))$ via $[\cdot]$, and finally $z' \cup z'' \in \mathcal{C}(\text{Pr}(\mathcal{F}))$ since z' and z'' are finite.

First notice that, because of 2, we have $\hat{z}' = \text{top}[z \cup [e_1]_{z_1}] = \hat{z} \cup \{e_1\}$ and similarly $\hat{z}'' = \hat{z} \cup \{e_2\}$. Denoting $\hat{z} = \bar{x} \parallel y$ it follows that $\hat{z}' = \bar{x}' \parallel y'$ where $(x', y') = (x, y \cup \{e_1\})$ and $\hat{z}'' = x'' \parallel y''$ where $(x'', y'') = (x \cup \{e_2\}, y)$ if $\text{pol}_{A^\perp \parallel B}(e_2) = -$ and $(x, y \cup \{e_2\})$ otherwise.

Now, since $\hat{z}', \hat{z}'' \in \mathcal{F}$, $z' \cup z''$ is downward-closed as union of such sets. Besides, if $\text{pol}_{A^\perp \parallel B}(e_2) = -$ then $x' \cup x'' = x \cup \{e_1\} = x'' \in \text{Con}_A$ and else $x' \cup x'' = x = x' \in \text{Con}_A$. Therefore $x' \cup x'' \in \mathcal{C}^\infty(A)$. Because $\hat{z}', \hat{z}'' \in \mathcal{F}$, we have $y' \subseteq f(x') \subseteq f(x' \cup x'')$ by monotonicity of f and similarly $y'' \subseteq f(x'') \subseteq f(x' \cup x'')$, hence

$$y' \cup y'' \subseteq f(x' \cup x'').$$

In particular, since $f(x' \cup x'') \in \mathcal{C}^\infty(B)$, $y' \cup y''$ is consistent, yielding $\hat{z}' \cup \hat{z}'' \in \mathcal{C}^\infty(A^\perp \parallel B)$ and, together with the previous equation, this proves that $\hat{z}' \cup \hat{z}'' \in \mathcal{F}$.

■

C Proofs of section 3

Proof of lemma 3.1. First, let us prove that θ is well-defined. Let $x \in \mathcal{C}^\infty(A)$. Then, since $\emptyset \subseteq^- \bar{x}$, proposition 1.10 ensures that there exists a unique $z \in \mathcal{C}^\infty(S)$ such that $\sigma[z] = x$. Now consider a $+$ -maximal extension z' of z . Since z' is obtained from z only by adding positive events, $\sigma[z']$ is of the form $\bar{x} \parallel y$ because σ preserves polarity. We need to prove that z' is unique and that $y = \tilde{\sigma}(x)$.

Suppose that $z' \neq z''$ are two $+$ -maximal extensions of z . Then the negative parts of z' and z'' are equal (they are both obtained from z only by adding positive events), thus $\text{Neg}[z' \cup z''] \in \text{Con}_S$. By determinism of σ , this implies $z' \cup z'' \in \text{Con}_S$ and it follows that $z' \cup z'' \in \mathcal{C}^\infty(S)$, which is absurd. Hence the uniqueness of z' .

Now let us prove that $y = \tilde{\sigma}(x)$. It is clear that $y \subseteq \tilde{\sigma}(x)$ so we only have to prove that $\tilde{\sigma}(x) \subseteq y$. To do so, we will prove by contradiction that for every $y' \in \mathcal{C}^\infty(B)$ such that there exists some $t \in \mathcal{C}^\infty(S)$ such that $\bar{x} \parallel y' = \sigma[t]$, we have $y' \subseteq y$. If $y' \not\subseteq y$ then there exists some $e \in y' \setminus y$. It follows that $\bar{x} \parallel [e] \subseteq^+ \bar{x} \parallel y'$ and, since $y' = \sigma[t]$, proposition 1.10 ensures that there exists a unique $t'' \in \mathcal{C}^\infty(S)$ such that $\sigma[t''] = \bar{x} \parallel [e]$ and $t'' \subseteq t$. Let t' be the (unique) $+$ -maximal extension of t'' . Since t' and z' have the same negative part and are both $+$ -maximal, we have $t' = z'$. This entails that $\sigma[t'] = \bar{x} \parallel y$ which is absurd because $\bar{x} \parallel [e] \subseteq \sigma[t']$.

To prove that θ is bijective, we will prove that it is mutual inverse with $\xi := \tau \circ \pi_1 \circ \sigma$. It is clear that $\xi \circ \theta = \text{id}_{\mathcal{C}^\infty(A)}$. Now let $z \in \mathcal{C}_{+-\max}^\infty(S)$ and $x \in \mathcal{C}^\infty(A)$ such that $\sigma[z] := \bar{x} \parallel y$ for some $y \in \mathcal{C}^\infty(B)$. We need to prove that $\theta(x) = z$.

Let z'' be the unique configuration of S (provided by proposition 1.10) such that $\sigma[z''] = \bar{x}$. By definition, $\theta(x)$ is the only $+$ -maximal extension z' of z'' . But, by denoting z^- the projection of z on its negative events, we have $\sigma[z^-] = \bar{x}$ since σ preserves polarity. Thus z and z' have the same negative part and are both $+$ -maximal, which proves that $z' = z$ i.e. $\theta(x) = z$. ■

Proof of proposition 3.2. σ is a strategy on $A^\perp \parallel B = (A_p^\perp \parallel B_n) \parallel (A_n^\perp \parallel B_p)$ hence, by theorem 2.3, it induces a Berry-stable map $\tilde{\sigma}: \mathcal{C}^\infty(A_p \parallel B_n^\perp) \rightarrow \mathcal{C}^\infty(A_n^\perp \parallel B_p)$ that is, up to isomorphism, a Berry-stable map $h: \mathcal{C}^\infty(A_p \parallel B_n) \rightarrow \mathcal{C}^\infty(B_p) \times \mathcal{C}^\infty(A_n)$. Let then $f' := \pi_1 \circ h$ and $g := \pi_2 \circ h$, where π_1 (resp. π_2) denotes the projection on the first (resp. second) coordinate. It is clear that f' and g are still Berry-stable maps (because π_1 and π_2 are). Finally, by defining f to be the restriction of f' to $\mathcal{C}^\infty(A_p)$ (that is, $\forall x \in \mathcal{C}^\infty(A_p) \cdot f(x) := f'(x \parallel \emptyset)$), we are provided with a couple $\langle f, g \rangle$ of Berry-stable maps of the right signature.

Now suppose that $x \parallel g(x \parallel y) \in W_A$ for some $x \in \mathcal{C}^\infty(A_p)$ and $y \in \mathcal{C}^\infty(B_n)$. We need to prove that $f(x) \parallel y \in W_B$. By lemma 3.1, since $\bar{x} \parallel y \in \mathcal{C}^\infty(A_p^\perp \parallel B_n)$, there exists some $z \in \mathcal{C}_{+-\max}^\infty(S)$ such that $\sigma[z] = (\bar{x} \parallel y) \parallel \tilde{\sigma}(\bar{x} \parallel y)$. Besides $\tilde{\sigma}(\bar{x} \parallel y) = x \parallel g(x \parallel y)$ by definition and $x \parallel g(x \parallel y) \in W_A$ by assumption, therefore the determinism of σ entails $f(x) \parallel y \in W_B$. ■

Proof (partial) of proposition 3.3. Let

$$f': \begin{cases} \mathcal{C}^\infty(A_p \parallel B_n) & \longrightarrow & \mathcal{C}^\infty(B_p) \\ x \parallel y & \longmapsto & f(x) \end{cases} \quad \text{and} \quad h: \begin{cases} \mathcal{C}^\infty(A_p \parallel B_n) & \longrightarrow & \mathcal{C}^\infty(A_n) \times \mathcal{C}^\infty(B_p) \\ x \parallel y & \longmapsto & (f(x), g(x \parallel y)) \end{cases}.$$

Then, since $f' = f \circ \pi_1$ is Berry-stable (with $\pi_1: \mathcal{C}^\infty(A_p \parallel B_n) \rightarrow \mathcal{C}^\infty(A_p)$), h can be seen up to isomorphism as a Berry-stable map $h': \mathcal{C}^\infty(A_p \parallel B_n^\perp) \rightarrow \mathcal{C}^\infty(A_n^\perp \parallel B_p)$ hence, by theorem 2.6, it induces a deterministic strategy $\sigma: S \rightarrow A^\perp \parallel B$ from A to B where $S = \text{Pr}(\mathcal{F})$ with $\mathcal{F} := \{\bar{x} \parallel y \in \mathcal{C}^\infty((A_p^\perp \parallel B_n) \parallel (A_n^\perp \parallel B_p)) \mid y \subseteq h'(x)\}$.

By lack of time, we must admit the two following verifications:

- ✈ • σ preserves the structure of imperfect information, that is, for every events $s, s' \in S$, if $s \leq_S s'$ then $l_{A^\perp \parallel B}(\sigma(s)) \leq l_{A^\perp \parallel B}(\sigma(s'))$;
- ✈ • σ is winning.

■

Proof of theorem 3.4. By symetry arguments, it suffices to prove that $\pi_1(\phi(x \parallel y)) \subseteq \pi_1(\phi(x \parallel y'))$.

First, let us prove that $b \in \pi_1(\phi(x \parallel y))$ iff there exists some $s \in S$ and $y_+ \in \mathcal{C}^\infty(B_p)$ such that $\sigma(s) = b$ and $\sigma[[s]] \subseteq \bar{x} \parallel y \parallel y_+$ (\star). Let $b \in \pi_1(\phi(x \parallel y))$. Then, by definition of ϕ , there exists some $\bar{x}^- \in \mathcal{C}^\infty(A_n^\perp)$, $y_+ \in \mathcal{C}^\infty(B_p)$ and $z \in \mathcal{C}^\infty(S)$ such that $b \in y_+$ and $\sigma[z] = \bar{x}^- \parallel y \parallel \bar{x}^- \parallel y_+$ and $b \in y_+$. Let $s \in z$ such that $\sigma(s) = b$, then $\sigma[[s]] \subseteq \bar{x}^- \parallel y \parallel y_+$ and s suits for \star . Indeed, suppose that there exists some $e \in [s]$ such that $\sigma(e) \in \bar{x}^-$; then, by denoting $e = e_0 \rightarrow e_1 \rightarrow \dots \rightarrow e_n = s$ (which is possible since $\{e \in S \mid e \leq s\}$ is finite) and letting $k := \max\{i \in [0, n-1] \mid \sigma(e_i) \in \bar{x}^-\}$, $+$ -innocence of σ entails the dependency $\sigma(e_k) \rightarrow \sigma(e_{k+1})$ which cannot exist since $\sigma(e_k) \in A_n^\perp$ and $\sigma(e_{k+1}) \notin A_n^\perp$ by maximality of k .

Conversely, suppose that there exists some $s \in S$ and $y_+ \in \mathcal{C}^\infty(B_p)$ such that $\sigma(s) = b$ and $\sigma[[s]] \subseteq \bar{x} \parallel y \parallel y_+$. Then $\sigma[[s]]$ is of the form $\bar{x}' \parallel y' \parallel y'_+ \in \mathcal{C}^\infty(A_n^\perp \parallel B_n \parallel B_p)$ with $x' \subseteq x$, $y' \subseteq y$, $y'_+ \subseteq y_+$ and $b \in y'_+$ (because $b \in \sigma[[s]]$). Since $\bar{x}' \parallel y' \parallel y'_+ \subseteq^- \bar{x} \parallel y \parallel y'_+$, proposition 1.10 ensures the existence of $z \in \mathcal{C}^\infty(S)$ such that $\sigma[z] = \bar{x} \parallel y \parallel y'_+$. Therefore, $y'_+ \subseteq \pi_1(\phi(x \parallel y))$ and, in particular, $b \in \pi_1(\phi(x \parallel y))$.

Now we are able to conclude. Let $b \in \pi_1(\phi(x \parallel y))$. Then, by \star , there exists some $s \in S$ and $y_+ \in \mathcal{C}^\infty(B_p)$ such that $\sigma(s) = b$ and $\sigma[[s]] \subseteq \bar{x} \parallel y \parallel y_+$. As it happens, we have $\sigma[[s]] \subseteq \bar{x} \parallel y_+$. Indeed, if there existed $s' \in [s]$ such that $\sigma(s') \in y$, we would have $\sigma(s') \in B_n$ and equation 1 would entail $l_B(\sigma(s')) \leq l_B(\sigma(s))$, which contradicts the definition of B since $\sigma(s) \in B_p$. In particular, $\sigma[[s]] \subseteq \bar{x} \parallel y' \parallel y_+$ which means, by \star again, that $b \in \pi_1(\phi(x \parallel y'))$. ■

Proof of corollary 3.4.1. Theorem 3.4, together with proposition 2.7, proves that the transformation described in propositions 3.2 $\theta: \sigma \mapsto \langle f, g \rangle$ is injective : indeed, $\sigma \mapsto \text{stab}(\sigma)$ and $\text{stab}(\sigma) \mapsto \langle f, g \rangle$ are both injective. Furthermore, it is clear from the proof of proposition 3.3 that θ is also surjective. Hence θ is bijective. ■