Probability and nondeterminism in compositional game theory

Josef Bolt

St. Anne's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2019

Abstract

We substantially extend previous work in the emerging field of compositional game theory.

We generalise work by Escardó and Oliva relating the selection monad to game theory. Escardó and Oliva showed that the tensor operation of selection functions computes a subgame perfect equilibirium of a sequential game. We investigate game theoretic interpretations of selection functions generalised over a monad. In particular we focus on the finite non-empty powerset monad which we use to model nondeterministic games. We prove a negative result: that nondeterministic selection functions do not compute the collection of all subgame perfect plays of a sequential game. We then define a solution concept related to the iterated removal of strongly dominated strategies, and then show that the tensor of nondeterministic selection functions computes the plays of strategy profiles satisfying this solution concept.

In the second part of this thesis we greatly expand the expressive power of open games, first introduced by Jules Hedges [Hed16]. In the current literature, open games are defined using the category of sets and functions as an ambient category. We define a category of open games that can use any symmetric monoidal category as an ambient category. This is accomplished using coend lenses which can be used to model certain bidirectional processes. Generalising open games to arbitrary symmetric monoidal categories allows us to, in particular, model probablistic games involving Bayesian agents in an open games formalism.

Acknowledgements

The work in this thesis could not have been completed without the help of many others. Outside of academia, I am grateful to my family for their continuing support of my studies. Concerning research, top thanks must go to my co-supervisor Jules Hedges for his insight and a steady stream of interesting problems to work on. I have also benefited greatly from conversations with Bob Coecke (co-supervisor), Dan Marsden, Philipp Zahn, Maaike Zwart, Guillaume Boisseau, and Mitchell Riley. I must also extend my thanks to my examiners Jeremy Gibbons and Neil Ghani for their helpful constructive feedback and suggestions. I gratefully acknowledge that my DPhil was funded by an EPSRC doctoral training grant, reference OUCL/2015/JMB.

Preface

I had originally planned, at the start of my DPhil in 2015, to work on incorporating dynamic epistemic logic into the string diagrams of monoidal category theory by extending the work in the paper [BCS07]. Conceptually, the leap from dynamic epistemic logic to game theory is not so great. Both involve epistemic states that can be updated in the face of new information or events. When Bob Coecke mentioned in 2016 that a post-doc called Jules Hedges had just arrived to work on making game theory compositional using monoidal category theory, my interest was piqued and I got in touch. The field of open games was even younger then than it is now, and the list of interesting open problems was long. It took several months to get to grips with the existing literature (Jules' own PhD thesis for the most part). It was then suggested that I attempt to generalise open games such that players in an open game would be able to perform Bayesian updating. The problem seemed like it would be straightforward at first, requiring only that the type structure of open games be modified slightly. Before long, however, it became clear that the problem was more subtle than originally thought and it would take roughly six months for the first category of Bayesian open games to be defined. This first pass at a definition bore little resemblance to the category presented in the second part of this thesis, and the process of streamlining it into its current form was greatly expedited by two fortuitous encounters. The first was a meeting between Jules, Mitchell Riley, and myself, in which we realised that Mitchell's work on *coend lenses* in the excellent paper [Ril18] was perfectly suited to open games. The second was a conversation between myself and Guillaume Boisseau, in which Guillaume suggested a modified definition for the best response function of an open game. This insight allowed the use of more coend lenses in the definition of an open game, and brought open games to a pleasing level of generality and theoretical unity.

During bouts of frustration over open games, I found time to work on nondeterministic sequential games. Sequential games, introduced by Paolo Oliva and Martin Escardó in [EO10b], are another approach to compositional game theory that make use of the selection monad to model backwards induction. I answered some open questions regarding selection functions generalised over the finite non-empty powerset monad and, in doing so, laid some of the foundations for further work on sequential games generalised over monads. This work is presented in Part I of this thesis.

Contents

Chapter 1

Introduction

This thesis is about game theory and its methods are from applied category theory. It is written with the applied category theorist, rather than the game theorist, in mind. Interesting categories are defined, but their categorical structure is not probed too deeply. Rather, we prove correctness results demonstrating that we have chosen our categories well and give examples showing how various games can be translated into an appropriate category. It is not assumed that the reader has prior knowledge of game theory, but an understanding of basic category theory is assumed. Anyone approaching this thesis from a purely game theoretic background will find it tough going, for which the author can only apologise.

1.1 Compositionality

Good design requires that attention be paid to the possible contexts in which a structure or object will be placed and how that context may change over time. An upgrade-able machine is superior to a non-upgrade-able machine; a machine that interfaces easily with other machines is superior to a machine that does not; and the more varied conditions under which a machine can operate, the better. These observations illustrate aspects of good *compositional design* (we might also call it good *modular design*). When we design, it is not sufficient to consider how well a structure carries out its narrow function. We must also consider how robustly that structure will compose with other structures to make larger structures.

An example of the power of good compositional design is the advent of *interchangeable* parts. Prior to the eighteenth century, gun components were made to fit one particular gun currently under construction. Every gun was custom-made, and it was unlikely that the components of any given gun would function in any other gun. As a consequence, if a component broke, either a new component would have to be custom-made , or else the gun would have to be replaced in its entirety. Gun components were only able to perform their function in the very specific context of the gun they were originally designed for. The introduction of standardised components, which worked in the contexts of various types of gun, led to scalable manufacturing, ease of repair, and predictable performance.

Good compositional design is as important in applied mathematics as it is in engineering. Given the choice between two modeling techniques, both of which model a particular phenomenon equally well, we should prefer the technique with superior compositional properties. We should ask questions such as, 'which generalises more easily?', 'which is more easily related to other areas of mathematics?', 'which scales better?', and 'which model's instances are easier to compose?'

1.1.1 Compositional game theory

Compositional game theory applies good compositional design in the following ways:

1. Games are defined such that there exist meaningful composition operators. *Sequential games* (Part I) can be sequentially composed, *open games* (Part II) can be composed in sequence and in parallel;

2. These operators, moreover, preserve information about the solution concepts of their operands. That is, the solutions of a composite game are computed by the composition operators given the solutions of the component games;

3. Atomic games are defined, out of which a large class of more complex games can be constructed using the compositional operators.

In order to achieve the above, we (loosely) shift our perspective on games as follows. In classical game theory, games are typically defined by some structure describing how a game is played (a tuple of sets of strategies in the case of simultaneous games, a tree for games with sequential play) together with an outcome function for each player. In compositional game theory, games have a play structure but take an outcome function (or generalisation thereof) as an *argument*, and return information about the optimal strategies of the game. The technical advantages of considering games in this way are

1. Game structures readily compose; and

2. An outcome function for a composite game can recursively be broken down into outcome functions for the subgames, which can then be supplied as arguments to the 'outcome function to solution' mapping of those subgames.

The benefits of this compositional approach to the resulting theory are

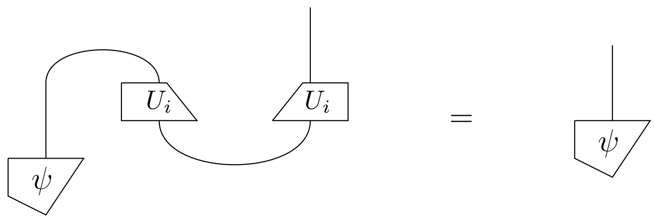
1. Many interesting types of game can be constructed from a small collection of atoms and the composition operators;

2. Games can be analysed by looking at their compositional structure. It is no longer necessary to consider a given game as a monolith;

3. The theory is sufficiently expressive to describe many different kinds of game, unifying much of game theory into one framework.

1.1.2 Compositional tools

In Part II of this thesis, we will be using monoidal category theory extensively. Monoidal category theory and its associated string diagrams provide a powerful toolkit for doing compositional mathematics with a comparatively low bar to entry. Many facts which, obscure in their 'native' language, are proven much more easily once phrased in the language of an appropriate monoidal category. Monoidal categories have already been applied, with great success, to quantum theory (Coecke and Kissinger's book [CK17] is an excellent introduction to the monoidal category theory of quantum mechanics). An example of the expressive power of monoidal category for quantum theory is *quantum teleportation*, a phenomenon not considered until 1993 in the paper [BBC$^+$93], many decades after the initial introduction of quantum theory. In the diagrammatic language of the category of Hilbert spaces, quantum teleportation is simply described by the diagram



where ψ is the quantum state to be teleported and the U_i represent the introduction and correction of quantum error (represented as unitary matrices in a Hilbert space).

1.2 Category theory

This thesis assumes some basic competence in category theory. In particular the definitions of *category*, *functor*, *natural transformation*, *(co)product*, and *(co)equaliser* are assumed to be known to the reader. Categorical notions beyond these will be introduced as and when they are needed (it is not assumed that the reader is familiar with monads or monoidal categories, for instance, but their introductions will be terse). For the reader unfamiliar with category theory, the following texts are excellent for background reading and as references. Good general category theory references are the textbooks by Leinster [Lei16] and Awodey [Awo10].The textbook [ML71] is the classic reference for monoidal categories, and the paper [Sel10] is a comprehensive guide to the various diagrammatic calculi associated with monoidal categories. Chapter 6 relies heavily on *coends* and a visual short-hand for them. The literature on coends is scant, but the paper [Lor15] provides good background and the paper [Ril18] is the original source for the diagrams we will be using.

1.3 Game theory

It is not assumed that the reader has any prior knowledge of game theory. Game theoretic notions will be introduced and discussed when they are needed, but the game theory covered is not strenuous. The game theory literature is fragmented and foundational works are often written with economists in mind. Good references with a more mathematical style are the textbooks [MSZ13] and [LB08].

Part I of this thesis can be seen as an extension of the work on sequential games in the papers [EO10b] and [EO10a]. These papers will be discussed at length in Chapter 2, as a thorough understanding of them is essential for understanding Chapter 3.

1.4 Structure of this thesis

The two parts of this thesis pursue a similar goal. Namely, a compositional account of a particular class of games. In spite of their similar aims, however, the two parts differ markedly in the style of their mathematics. Part I has a 'close-up' view of its subject matter, focusing in particular on the interaction between the selection monad and the nondeterminism monad and the resulting game theoretic implications. Part II takes a more general categorical perspective before specialising to probability distributions in order to model games with Bayesian components.

In part I:

- Chapter 2 covers the existing literature on sequential games. A key result is that selection functions compute a subgame perfect equilibrium play for a sequential game. This chapter is included for completeness and as a prerequisite for the following chapter.

- Chapter 3 generalises sequential games in order to model games with nondeterminism. We prove that nondeterministic selection functions do not compute the set of subgame perfect Nash equilibrium plays and also prove that there is a solution concept, the plays of which *are* computed by nondeterministic selection functions.

- Chapter 4 contains concluding remarks on selection functions and possible directions for future work.

In part II:

- Chapter 5 covers existing literature on *lenses* and *open games* in some detail. In this chapter these notions are considered only over the category **Set**. The work in this chapter is mostly not original, but it does fill in some gaps in the existing literature and, in places, takes a slightly different approach.

- Chapter 6 presents a generalisation of lenses over (more) arbitrary symmetric monoidal categories, and also generalises open games similarly. This generalisation is non-trivial, and this chapter is the most conceptually difficult in this thesis.

- Chapter 7 climbs back down from the abstraction of Chapter 6 to analyse *Bayesian open games*, which are open games over the Kleisli category of the finite distribution monad. It is established that Bayesian open games model certain classes of Bayesian games from classical game theory.

- Chapter 8 offers some concluding remarks about sequential games and selection functions, considering directions for future work and where future difficulties are likely to lie.

1.5 Contributions

In Part I, the definitions of nondeterministic sequential games, generalised selection functions and their tensor operations, are due to the papers [EO10b] and [EO14] rather than the author. The remainder of Part I (from 3.6 onward), mainly comprising in definitions and results concerning generalised selection functions, is original work. This work includes

- A definition of 'well-behaved' nondeterministic selection functions (3.6);
- A definition of rationality for strategy profiles in two-player nondeterministic sequential games 3.7;
- A theorem stating that well-behaved selection functions compute the plays of rational strategy profiles (3.7.0.5);
- A generalisation of the above to n-round nondeterministic sequential games (3.9);
- A negative result stating that nondeterministic selection functions do not, in general, compute the set of subgame perfect plays (3.8.0.3);
- A proof that rational strategy profiles correspond to a sequential version of an already-known solution concept from game theory — the iterated removal of strictly dominated strategies (3.10).

In Part II, the main original contributions are the definition of a *context* for a generalised open game (6.6.0.2) and the proof that generalised open games form a symmetric monoidal category. The chronology of the definition of a context for an open game is reversed in this thesis: first it is presented in a more general form as a state

$$\mathbf{Lens}_{\mathbf{Lens}_\mathcal{C}}(I, (\Phi, \Psi)) = \int^{\Theta : \mathbf{Lens}_\mathcal{C}} \mathbf{Lens}_\mathcal{C}(I, \Theta \otimes \Phi) \times \mathbf{Lens}_\mathcal{C}(\Theta \otimes \Psi, I)$$

in a category of iterated lenses. We then show that contexts can equivalently be given (where $\Phi = (X, S)$ and $\Psi = (Y, R)$) as members of the coend

$$\int^{A:\mathcal{C}} \mathcal{C}(I, A \otimes X) \times \mathcal{C}(A \otimes Y, R)$$

in the case where the monoidal unit I of $\mathcal{C}$ is terminal. In reality, the latter form was discovered first by the author, resulting in the first definition of a category of generalised open games for which the monoidal product of the underlying category need not be cartesian (and, as a consequence, which could accommodate games involving Bayesian agents). The former form was discovered with the help of Guillaume Boisseau (in an informal conversation), who suggested taking a coend in $\mathbf{Lens}_\mathcal{C}$ in place of a coend in $\mathcal{C}$, meaning that the requirement that I must be terminal could be dropped. The realisation that the more general form corresponds to a state in a category of iterated lenses is due to the author.

1.6 Notational conventions

Given a function $f : X \to Y$ and $U \subseteq X$, we write $f(U)$ for $\{f(x) \mid x \in U\}$ (a standard abuse of notation). If $k : X \times Y \to Z$, we sometimes write $k(x, -)$ to denote $\lambda(y : Y).\, k(x, y)$ to make expressions more concise. The 'bang' operator ! is used to denote unique objects. For instance, $! : X \to \tau$ may denote the unique arrow to the terminal object τ in a category. Given a set X, $\mathrm{Rel}(X)$ denotes the set of relations on X. That is, $\mathrm{Rel}(X) = X \to \mathcal{P}(X)$. We also use double-headed arrows to denote relations. For example, $\beta : X \twoheadrightarrow Y$ denotes a relation β from X to Y. Then $x \overset{\beta}{\sim} y$ just means $y \in \beta(x)$. Given a category $\mathcal{C}$ with objects X, Y, we denote the hom-set of morphisms $X \to Y$ by $\mathcal{C}(X, Y)$. We adopt the following neat convention from the game theory literature. If $x = (x_1, \cdots, x_n) \in \prod_{i=1}^{n} X_i$, we write x_{-i} to denote the tuple x with the i^{th} component removed. That is $x_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots x_n)$. We also write (x_{-i}, x'_i) for the tuple obtained by replacing x_i with x'_i in x.

Part I

Sequential games

Chapter 2

Sequential games

Let's characterise an *agent* simply as something that makes a choice and then receives some outcome from its environment. This outcome may depend on the choice made by the agent and it may also depend on things external to the agent (perhaps the choices of other agents or random events). Let's also presume that our agent has some preference regarding the outcome it will receive and will make choices to induce satisfactory outcomes. If our agent makes a choice of type X and receives an outcome of type R, a function of type $X \to R$ specifies a way that choices are mapped to outcomes. We call such functions *contexts*, but they may also be understood as *outcome functions* or *utility functions*.

In what follows, we allow the collection of outcomes considered acceptable to an agent to depend on the agent's context. That is, a function which picks out a satisfactory outcome will have type $(X \to R) \to R$. We call such functions *quantifiers*[1]. As a paradigm example, when $R = \mathbb{R}$ and X is finite, there is the max quantifier given by $\max(k) = \max(k(X))$.

Similarly, we can allow the *choices* that an agent considers satisfactory to depend on a context. A function picking out a satisfactory choice given a context has type $(X \to R) \to X$ and is called a *selection function*. Analogous to the max quantifier, there are arg max selection functions. If $R = \mathbb{R}$ and X is finite as before, an arg max selection function is a selection function arg max satisfying, for all contexts k, $\arg\max(k) = x$ where $k(x) = \max(k(X))$[2]. We will see that quantifiers and selection functions admit of a product operation that can be used to describe backward induction in games involving sequential play.

Giving agents access to contexts is the technical innovation that gives the theory of sequential games its power. An analogy to computation is that we are doing game

[1]This name is justified in 2.3

[2]arg max selection functions are not unique, and we address this in detail in the next chapter

theory in *continuation passing style*. Functions in this style take a 'continuation' function as an extra parameter (where a continuation can be thought of as 'what is going to be done with the output of the function'). In game theoretic terms, this corresponds to giving the players of a game access to the function that will generate outcomes given their choices. That is, players get access to the *context* of the game.

Throughout this chapter we assume that we work over some underlying category $\mathcal{C}$ which is cartesian closed. In particular we use the fact that terms of the simply typed lambda calculus can be interpreted as morphisms in a cartesian closed category. We take a fairly relaxed approach to this, as we are predominantly interested in the case where the underlying category is **Set**. Nevertheless, the definitions in the following sections do hold for arbitrary cartesian closed categories.

2.1 Chapter overview

This chapter is mainly drawn from [EO10b] and [EO10a], the authoritative works on sequential games (in particular, the work in this chapter is not original). In 2.2 we cover various definitions concerning monads that will be needed throughout the rest of this thesis. 2.3 introduces *quantifiers*, functions with types of the form $(X \to R) \to R$; 2.4 introduces *selection functions*, functions with types of the form $(X \to R) \to X$; 2.5 introduces some concepts from classical game theory including the notion of a *solution concept*; 2.6 formally introduces *sequential games*.

The content of this chapter is included for completeness and ease of reference. Moreover, the material in this chapter is a hard prerequisite for the material in Chapter 3.

2.2 Monads

In this section we cover some basic monad definitions. We will be terse and direct readers to the book [BW85] as a reference.

Definition 2.2.0.1 (Monad)**.** A *monad* on a category $\mathcal{C}$ is a triple $(T, \eta, -^\dagger)$ where

1. An assignment $T : \mathcal{C} \to \mathcal{C}$ (this is a mapping on objects, *not* a functor);

2. η is a family of morphisms $\{\eta_A : A \to T(A) \mid A : \mathcal{C}\}$ where η is called the *unit* of T;

3. $-^\dagger$ is a family of **Set** functions $\left\{-^\dagger_{A,B} : \mathcal{C}(A, T(B)) \to \mathcal{C}(T(A), T(B)) \;\middle|\; A, B : \mathcal{C}\right\}$ where $f^\dagger$ is called the *(Kleisli) extension* of f;

such that whenever $f : A \to T(B)$ and $g : B \to T(C)$ the following axioms hold,

$$\eta_A^\dagger = \mathrm{id}_{T(A)} \qquad f^\dagger \circ \eta_A = f \qquad g^\dagger \circ f^\dagger = (g^\dagger \circ f)^\dagger.$$

We abuse notation slightly and refer to a monad by its associated mapping T.[3]

Definition 2.2.0.2 (Kleisli category)**.** The *Kleisli category*, $\mathbf{Kl}(T)$, of a monad T on a category $\mathcal{C}$ has the same objects as $\mathcal{C}$ and hom-sets $\mathbf{Kl}(T)(X, Y) = \mathcal{C}(X, T(Y))$. The identity morphism on an object X is the unit map $\eta_X : X \to T(X)$ and the composition of $f : X \to T(Y)$ and $g : Y \to T(Z)$ is $g^\dagger \circ f$.

A monad is *strong* if it comes with a *strength* natural transformation $\mathrm{st}_{A,B} : (A \times T(B)) \to T(A \times B)$ satisfying certain coherence axioms. If $\mathcal{C}$ is cartesian closed, T being strong is equivalent to T being $\mathcal{C}$-enriched [Koc70]. That is, the action of T on morphisms is given by a morphism $(A \to B) \to (T(A) \to T(B))$ of $\mathcal{C}$ rather than a **Set** function $\mathcal{C}(A, B) \to \mathcal{C}(T(A), T(B))$ (as a consequence, all monads over **Set** are strong).

Strong monads induce useful tensor operations that will correspond to *sequential play* in the theory of sequential games.

Definition 2.2.0.3 (Dependent tensor for strong monads)**.** A strong monad T over a cartesian closed category $\mathcal{C}$ induces a *dependent tensor* $\boxtimes_{A,B} : \big(T(A) \times (A \to T(B))\big) \to T(A \times B))$ given by

$$a \boxtimes f = \Big(\lambda(x : A).\big(\lambda(y : B).\eta_{X \times Y}(x, y)\big)^\dagger(f(x))\big)\Big)^\dagger(a).$$

Definition 2.2.0.4 (Independent tensor for strong monads)**.** A strong monad T over a cartesian closed category $\mathcal{C}$ induces an *independent tensor* $\otimes_{A,B} : T(A) \times T(B) \to T(A \times B)$ given by

$$a \otimes b = a \boxtimes (\lambda(x : A).\, b)$$

Example 2.2.0.5. In the case of the powerset monad $\mathcal{P}$ over **Set**, the independent tensor is the standard cartesian product

$$A \otimes_{\mathcal{P}} B = \big\{(a, b) \bigm| a \in A \text{ and } b \in B\big\}$$

and the dependent tensor is

$$A \boxtimes_{\mathcal{P}} f = \big\{(a, b) \bigm| a \in A \text{ and } b \in f(a)\big\}.$$

[3]This is the 'haskell style' definition and it is not standard, but is convenient for our purposes and can be shown to be equivalent to the standard definition by extending T to a functor and defining multiplication as $\mathrm{id}_{TA}^\dagger$.

Definition 2.2.0.6 (Algebra for a strong monad)**.** An *algebra*[4] for a strong monad T over $\mathcal{C}$ is an object R of $\mathcal{C}$ together with a family of maps $\{-^{\star}_{A} : (A \to R) \to (T(A) \to R) \mid A : \mathcal{C}\}$ such that, for all $f : A \to T(B)$ and $g : B \to R$,

1. $g^{\star} \circ \eta_B = g$, and
2. $(g^{\star} \circ f)^{\star} = g^{\star} \circ f^{\dagger}$.

2.3 Quantifiers

In this section we make formal the notion of a *quantifier*, a higher order function taking a function as argument and returning a value in the codomain of that argument.

Definition 2.3.0.1 (Quantifier)**.** A *quantifier* is function $\varphi : (X \to R) \to R$. The type $(X \to R) \to R$ is denoted by $\mathcal{K}_R(X)$.

A good first intuition of a quantifier is the higher-order function max.

Example 2.3.0.2. Given a finite nonempty set X, the quantifier $\max : (X \to \mathbb{R}) \to \mathbb{R}$ is given by $\max(k) = \max(k(X))$.

The max quantifier is thought of as modelling the utility maximising agents of classical game theory. Given a mapping from actions X to utility $\mathbb{R}$, max returns the optimal outcome.

The name 'quantifier' is not a misnomer. Functions of type $(X \to \mathbb{B}) \to \mathbb{B}$ were introduced in the paper [Mos79] as 'generalised quantifiers.' To see why, consider the following. A function $k : X \to \mathbb{B}$ can naturally be viewed as a unary predicate on X where $k(x)$ is true if and only if $k(x) = 1$. We recover the traditional quantifiers $\exists$ and $\forall$ as generalised quantifiers as follows.

Example 2.3.0.3. The quantifier $\exists : (X \to \mathbb{B}) \to \mathbb{B}$ is given by

$$\exists(k) = \begin{cases} 1, & \text{if there is } x \in X \text{ such that } k(x) = 1 \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, $\forall : (X \to \mathbb{B}) \to \mathbb{B}$ is given by

$$\forall(k) = \begin{cases} 1, & \text{if } k(x) = 1 \text{ for all } x \in X \\ 0, & \text{otherwise.} \end{cases}$$

[4] This definition, used also in [EO14], is not standard but is equivalent to the standard definition. We make use of it because it is convenient for our purposes and because our work is an extension of [EO14].

Note that $\exists$ and $\forall$ are, respectively, the max and min quantifiers for the poset $\{0 < 1\}$.

As the name suggests, generalised quantifiers are flexible enough to describe many interesting logical quantifiers not definable in terms of $\exists$ and $\forall$.

Example 2.3.0.4. Define **most** : $\mathcal{K}_{\mathbb{B}}(X)$ by $\mathbf{most}(k) = 1$ if and only if

$$|\{x \in X \mid k(x) = 1\}| > |\{x \in X \mid k(x) = 0\}|.$$

Example 2.3.0.5. Define **countable** : $\mathcal{K}_{\mathbb{B}}(X)$ by $\mathbf{countable}(k) = 1$ if and only if $|\{x \in X \mid k(x) = 1\}| \leq \aleph_0$.

Generalising further, we have many interesting example other than max where $R \neq \mathbb{B}$.

Example 2.3.0.6. Suppose now that we are working over a cartesian closed category of topological spaces[5] and that R is a topological space satisfying the *fixed point property*. That is, for every continuous function $f : R \to R$ there exists $r \in R$ such that $f(r) = r$. Then there exist quantifiers **fix** : $\mathcal{K}_R(R)$ satisfying $k(\mathbf{fix}(k)) = \mathbf{fix}(k)$ for all $k : R \to R$.

The mapping $X \mapsto \mathcal{K}_R(X)$ is the well-studied *continuation monad* (for discussion see, for example, Moggi's paper [Mog91]). Its unit is given by

$$\eta_X = \lambda(x : X).\, \lambda(k : X \to R).\, k(x)$$

and given a morphism $f : X \to \mathcal{K}_R(Y)$, its Kleisli extension $f^\dagger : \mathcal{K}_R(X) \to \mathcal{K}_R(Y)$ is given as follows. Given $\varphi : \mathcal{K}_R(X)$ and $k : Y \to R$, there is $\lambda(x : X).f(x)(k) : X \to R$. Then $f^\dagger(\varphi)(k)$ is given by $\varphi(\lambda(x : X).f(x)(k))$. That this forms a strong monad is shown by simple checks.

We explicitly describe the independent tensor induced by the strong monad structure.

Definition 2.3.0.7 (Tensor of quantifiers)**.** Given $\varphi : \mathcal{K}_R(X)$, $\psi : \mathcal{K}_R(Y)$, and $k : X \times Y \to R$, define

$$(\varphi \otimes \psi)(k) = \varphi\big(\lambda(x : X).\psi(\lambda(y : Y).k(x, y))\big).$$

[5]The category of topological spaces and continuous functions is not cartesian closed. Categories of topological spaces which are cartesian closed (and also satisfy certain other 'nice to have' properties) are often referred to as *convenient categories of topological spaces* (see the paper [S+67], for example), an example being the category of compactly generated spaces.

Example 2.3.0.8. The tensor in the case of $\exists$ is as one would expect.

$$(\exists^X \otimes \exists^Y)(k) = \begin{cases} 1 & \text{if there are } x \in X \text{ and } y \in Y \text{ with } k(x,y) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

The result of tensoring $\exists$ and $\forall$ is also not surprising.

Example 2.3.0.9. Given $\exists^X : \mathcal{K}_{\mathbb{B}}(X)$ and $\forall^Y : \mathcal{K}_{\mathbb{B}}(Y)$ their tensor is explicitly given by

$$(\exists^X \otimes \forall^Y)(k : X \times Y \to \mathbb{B}) = \begin{cases} 1 & \exists x : X \, \forall y : Y \; k(x,y) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Tensoring max quantifiers together simply yields a max operator on functions of two arguments.

Example 2.3.0.10. Given $\max^X : \mathcal{K}_{\mathbb{R}}(X)$ and $\max^Y : \mathcal{K}_{\mathbb{R}}(Y)$,

$$(\max^X \otimes \max^Y)(k : X \times Y \to \mathbb{R}) = \max(k(X \times Y))$$

2.4 Selection functions

If quantifiers return a satisfactory outcome, then *selection functions* return a choice that attains that satisfactory outcome. Accordingly, we think of selection functions as returning a *satisfactory choice.*

Definition 2.4.0.1 (Selection function)**.** A *selection function* is a function of the form $(X \to R) \to X$. The type $(X \to R) \to X$ is denoted $\mathcal{J}_R(X)$.

Example 2.4.0.2. Given a function $k : X \to \mathbb{R}$ on some finite set X, define $\arg\max(k)$ to be $\{x \in X \mid k(x) = \max(k(X))\}$. Then an $\arg\max$ selection function $\varepsilon_{\arg\max} : \mathcal{J}_{\mathbb{R}}(X)$ is a selection function satisfying $\varepsilon(k) \in \arg\max(k)$ for all contexts $k : X \to \mathbb{R}$.

Given a finite set X, the higher-order function $\arg\max : (X \to \mathbb{R}) \to \mathcal{P}(X)$ is naturally multivalued as a function may attain its maximum in multiple places. In the next chapter we look at *nondeterministic selection functions*, and study higher-order functions with this type structure in much more detail. For now, we work with *an* $\arg\max$ selection function from a family of $\arg\max$ selection functions (noting our reliance on the Axiom of Choice).

Selection functions have associated quantifiers. If a satisfactory play in a context k is x, then a satisfactory outcome is given by $k(x)$.

Definition 2.4.0.3. Given a selection function $\varepsilon : \mathcal{J}_R(X)$, the *quantifier for* ε is $\varphi_\varepsilon : \mathcal{K}_R(X)$ given by $\varphi_\varepsilon(k) = k(\varepsilon(k))$. A quantifier $\varphi : \mathcal{K}_R(X)$ is *attained* by a selection function $\varepsilon : \mathcal{J}_R(X)$ if φ is the quantifier for ε. We also say that φ is *attainable* if it is attained by some $\varepsilon : \mathcal{J}_R(X)$.

Note that a quantifier is considered not attainable precisely when there exists a context in which no argument attains the 'satisfactory' outcome as specified by the quantifier.

Example 2.4.0.4. Let $\varphi : \mathcal{K}_{\mathbb{R}}(X)$ be a quantifier such that $\varphi(c_0) = 1$ where $c_0 : X \to \mathbb{R}$ is the constant function $x \mapsto 0$. Then φ is not attainable.

The mapping $\mathcal{J}_R \to \mathcal{K}_R$ described above is, in fact, a monad morphism (as shown in [EO10a]).

Example 2.4.0.5. The existential quantifier $\exists : \mathcal{K}_{\mathbb{B}}(X)$ is attained by selection functions $\varepsilon_\exists : \mathcal{J}_{\mathbb{B}}(X)$ satisfying

$$\big(\exists x \in X.\, k(x) = 1\big) \implies k\big(\varepsilon_\exists(k)\big) = 1$$

for all contexts $k : X \to \mathbb{B}$.

The mapping $X \mapsto \mathcal{J}_R(X)$ has a strong monad structure, though far less is known about it than about the continuation monad.

Definition 2.4.0.6. The *selection monad* $\mathcal{J}_R$ is given by $\mathcal{J}_R(X) = (X \to R) \to X$. The unit η_X is

$$\lambda(x : X).\lambda(k : X \to R).x$$

and the Kleisli extension of a morphism $f : X \to \mathcal{J}_R(Y)$ is given by

$$f^\dagger(\varepsilon)(k) = f\Bigg(\varepsilon\big(\lambda(x : X).\, k(f(x)(k))\big)\Bigg)(k).$$

The proof that this structure is a strong monad can be found in [EO10a]. As with quantifiers, selection functions have dependent and independent tensors induced by the strong monad structure. We give an explicit formulation of the independent tensor for later use.

Definition 2.4.0.7 (Selection tensor)**.** Let $\varepsilon : \mathcal{J}_R(X)$ and $\delta : \mathcal{J}_R(Y)$. The tensor $\varepsilon \otimes \delta : \mathcal{J}_R(X \times Y)$ is given by

$$(\varepsilon \otimes \delta)(k : X \times Y \to R) = (a, f(a))$$

where

$$f = \lambda(x : X).\delta\Big(\lambda(y : Y).k(x, y)\Big)$$

and

$$a = \varepsilon\Big(\lambda(x : X).k(x, f(x))\Big).$$

Example 2.4.0.8. In the case $\varepsilon = \arg\max_{x \in X}$, $\delta = \arg\max_{y \in Y}$, $R = \mathbb{R}$, the tensor of selection functions is explicitly given by $(\arg\max_{x \in X} \otimes \arg\max_{y \in Y})(k : X \times Y \to \mathbb{R}) = (a, f(a))$ where

$$f(x) = \arg\max_{y \in Y}\Big(k(x, y)\Big)$$
$$a = \arg\max_{x \in X}\Big(k(x, \arg\max_{y \in Y}\big(k(x, y)\big))\Big).$$

In words, an x is chosen to maximise the function that will result after a y has been chosen to maximise k given x. In a sense, the selection functions are *cooperating* to maximise the value returned by k.

Example 2.4.0.9. If we change the above example such that $\delta = \arg\min$ rather than $\arg\max$, the selection functions can be seen to be *competing.* An $x \in X$ is chosen to maximise $k(x, y)$ where it is assumed that the y will be chosen such that it *minimises* $k(x, y)$. If one thinks of X and Y as the types of moves in a game, the connection to game theory is apparent.

2.5 Solution concepts

One of the concerns of classical game theory is that of providing *solution concepts.* A solution concept is usually a property of strategy profiles and strategy profiles satisfying a solution concept are considered 'optimal' in some sense. The most well-known solution concept in game theory, and the solution concept upon which many other solution concepts are based, is the *Nash equilibrium* (the modern usage of which can first be found in the seminal book [VNM44]). It is easiest to define Nash equilibria with respect to normal form games, which we give a definition of here.

Definition 2.5.0.1 (Normal form game)**.** A *normal form game* with n players is given by tuples $(S_1, \cdots, S_n)$ and $(q_1, \cdots, q_n)$ where

- S_i is the set of *strategies* for player i, and
- $q_i : \prod_{i=1}^{n} S_i \to \mathbb{R}$ is the *outcome function* for player i.

An element s_i of S_i is called a *strategy* (for player i) and a tuple $(s_1, \cdots, s_n) \in \prod_{i=1}^n S_i$ is called a *strategy profile.*

We are to think of the players $1, \cdots, n$ as each choosing some strategy $s_i \in S_i$ simultaneously. Each player i is then assigned the outcome $q_i(s_1, \cdots, s_n)$.

Example 2.5.0.2 (Prisoners' dilemma)**.** The prisoners' dilemma is a two person normal form game of perfect information where $S_1 = S_2 = \{\text{cooperate}, \text{defect}\}$. If both players choose 'cooperate,' they both receive utility -1; if both players choose 'defect', they both receive utility -2; if one player chooses 'cooperate' and the other chooses 'defect,' the player who cooperated receives utility -3 and the player who defected receives utility 0.

Given a strategy profile $s = (s_1, \cdots, s_n)$, we can think of the outcome that might occur if a particular player i were to unilaterally deviate from s.

Definition 2.5.0.3 (Unilateral deviation)**.** Given a normal form game $(S_i, q_i)_{i=1}^n$ we define the *unilateral deviation function* U_i^s of player i with respect to a strategy profile $s = (s_1, \cdots, s_n)$ by

$$U_i^s : S_i \to \mathbb{R}$$
$$U_i^s(s_i') = q_i(s_{-i}, s_i').$$

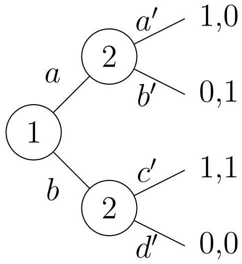
A Nash equilibrium is then defined as a strategy profile from which no player stands to gain by unilaterally deviating.

Definition 2.5.0.4 (Nash equilibrium)**.** A strategy profile $s = (s_1, \cdots, s_n)$ for a normal form game $(S_i, q_i)_{i=1}^n$ is a *Nash equilibrium* if for all $1 \leq i \leq n$ and all $s_i' \in S_i$ it holds that

$$q_i(s) \geq U_i^s(s_i').$$

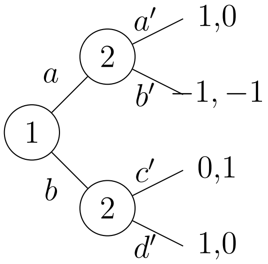
In the example of the prisoners' dilemma, the only Nash equilibrium is the strategy profile {defect, defect}. If one player cooperates, then the other player stands to gain by defecting.

Nash equilibria are 'optimal' in the sense that they are stable; no player has incentive to deviate. In games with sequential play, however, there exist Nash equilibria for which stability is implausible. A game of perfect information in *extensive form* can be described by a tree where each non-leaf node is labelled with the player whose turn it is to act; leaf nodes are labelled with outcomes for each player for that path through the tree; and edges are labelled with strategies. So, for instance, the tree



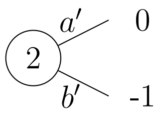
represents an extensive form game in which player 1 chooses a strategy from $\{a, b\}$, then player 2 makes a choice either from $\{a', b'\}$ (if player 1 picked a) or from $\{c', d'\}$ (if player 1 picked b). Both players then receive utility according to the appropriate leaf on the tree. A strategy profile for an extensive form game consists in a choice of strategy at each node. So, for instance, player 1 has two strategies in the above game; choose a or choose b. Player 2 has four strategies given by $(a', c'), (a', d'), (b', c')$, and (b', d'). These strategies for each player then form the obvious normal form game.

Consider the extensive form game described by



.

The strategy profile $(b, (b', c'))$ is a Nash equilibrium. Currently player 1 receives utility 0 and player 2 receives utility 1. If player 1 were to unilaterally deviate and play a, player 2 would play b' and both players would receive -1. This equilibrium is suspect; if player 1 actually played a, then player 2 would receive a better outcome by playing a'. The equilibrium $(b, (b', c')$ is a so-called *implausible threat*. A solution concept that does not classify such threats as 'solutions' is the *subgame perfect Nash equilibrium* (SPNE). Subgame perfect Nash equilibria can be defined in one of two equivalent ways. The first approach is to define an SPNE as a strategy profile which is a Nash equilibrium on all subtrees of the game tree (i.e. restricting the strategy profile to a subgame in the obvious way always yields a Nash equilibrium). By this definition, the strategy profile $(b, (b', c')$ fails to be a subgame perfect Nash equilibrium because the strategy profile b' is not a Nash equilibrium of the game



.

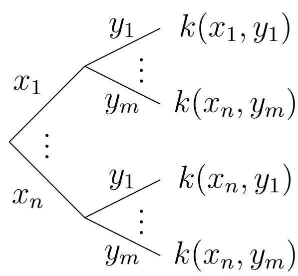
The second approach to refining the notion of 'Nash equilibrium', and the one that will be used for the remainder of this thesis, is the *one shot deviation principle*. The one shot deviation principle states that a strategy profile for an extensive form game is a subgame perfect Nash equilibrium if no player can profit by deviating from that strategy profile by a single decision in any subgame. Our example fails the one shot criterion as player 2 can profit by deviating from their decision in the same subgame given above.

2.6 Sequential games

Sequential games model a particular kind of strategic interaction. Suppose there are agents $\mathcal{A}_1, \ldots, \mathcal{A}_n$, and that each $\mathcal{A}_i$ has an associated type of *choices* X_i. The game plays out as follows: $\mathcal{A}_1$ makes a choice from X_1, $\mathcal{A}_2$ observes the choice made by $\mathcal{A}_1$ before making a choice from X_2, then $\mathcal{A}_3$ observes the choices made by $\mathcal{A}_1$ and $\mathcal{A}_2$ before making a choice from X_3, $\ldots$, then $\mathcal{A}_n$ observes the choices made by $\mathcal{A}_1, \cdots, \mathcal{A}_{n-1}$ before making a choice from X_n. Finally, an outcome is generated by an outcome function $q : \prod_{i=1}^n X_i \to R$.

Definition 2.6.0.1 (Sequential game)**.** An *n-round sequential game* is given by an n-tuple of selection functions $(\varepsilon_1, \cdots, \varepsilon_n)$ where $\varepsilon_i : \mathcal{J}_R(X_i)$ together with an *outcome function* $k : \prod_{i=1}^n X_i \to R$. The type X_i is thought of as the type of *choices* at round i. We denote a sequential game using the triple $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, k\big)$.

Sequential games can be seen as a special kind of extensive form game. Consider, for example, a 2 player sequential game $((X, Y), (\varepsilon, \delta), k : X \times Y \to R)$ where $X = \{x_1, \cdots, x_n\}$ and $Y = \{y_1, \cdots, y_n\}$. We can represent this game by the tree



2.6.1 Strategies and subgame perfection

A *strategy profile* for a sequential game consists in a choice for every node in the game tree. A node at the j^{th} stage of a game $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, q\big)$ is given by a tuple $(x_1, \cdots, x_{j-1}) \in \prod_{j=1}^{i-1} X_i$. We therefore make the following definition.

Definition 2.6.1.1 (Strategy)**.** A *strategy profile* for a sequential game $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, q\big)$ is a tuple $(\sigma_1, \cdots, \sigma_n)$ where $\sigma_i : \big(\prod_{j<i} X_j\big) \to X_i$. We refer to the individual σ_i as *strategies at round i*.

A strategy profile straightforwardly induces a playthrough of a sequential game. Similarly, a partial play of a sequential game can be extended to a full playthrough given a partial strategy profile for the rest of the game.

Definition 2.6.1.2 (Play of a strategy profile)**.** A *play* of a sequential game $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, k\big)$ is a tuple $x \in \prod_{i=1}^n X_i$. A *partial play* is a tuple $y \in \prod_{j=1}^i X_j$ for some $i < n$. The *strategic play* x^σ of a strategy profile $\sigma = (\sigma_1, \cdots, \sigma_n)$ is the play given recursively by

- $x_1^\sigma = \sigma_1$,
- $x_i^\sigma = \sigma_i(x_1^\sigma, \cdots, x_{i-1}^\sigma)$.

Given a partial play $x \in \prod_{j=1}^i X_j$ and a partial strategy profile $\sigma = (\sigma_{i+1}, \cdots, \sigma_n)$, the *strategic extension* x^σ of x is given by

$$x_j^\sigma = \begin{cases} x_i & \text{if } j \leq i \\ \sigma_j(x_1^\sigma, \cdots, x_{j-1}^\sigma) & \text{otherwise.} \end{cases}$$

In order to define SPNE for sequential games, we must first define *unilateral deviation* from a strategy profile.

Definition 2.6.1.3 (Unilateral deviation)**.** Let $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, q\big)$ be a sequential game. Let $x \in \prod_{j=1}^{i-1} X_j$, where $i \leq n$, be a partial play and let $\sigma = (\sigma_{i+1}, \cdots, \sigma_n)$ be a partial strategy profile. The *unilateral deviation function* $U_x^\sigma : X_i \to R$ for i at x is given by

$$U_x^\sigma(y_i) = q\big((x, y_i)^\sigma\big)$$

where $(x, y_i)^\sigma$ is the strategic extension of $(x_1, \cdots, x_{i-1}, y_i)$ by σ. The *best deviation* from σ at i for x is given by

$$\varepsilon_i\Big(\lambda(y_i : X_i).\, U_x^\sigma(y_i)\Big).$$

We think of a strategy profile σ as being 'acceptable' at round i if it agrees with the best deviation at i for every partial play of length $i - 1$ (i.e. the acceptable deviation is no deviation at all). That is, for all $x \in \prod_{j=1}^{i-1} X_j$,

$$\sigma_i(x) = \varepsilon_i\Big(\lambda(y_i : X_i).\, U_x^\sigma(y_i)\Big).$$

A subgame perfect Nash equilibrium for a sequential game is then just a strategy profile which is acceptable at every round.

Definition 2.6.1.4 (Subgame perfect Nash equilibrium (SPNE))**.** A strategy profile $\sigma = (\sigma_1, \cdots, \sigma_n)$ for a sequential game $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, q\big)$ is a *subgame perfect Nash equilibrium* (or an *SPNE*) if, for every $1 \leq i \leq n$ and $x \in \prod_{j=1}^{i-1} X_j$, it holds that

$$\sigma_i(x) = \varepsilon_i\Big(\lambda(y_i : X_i).\, U_x^{(\sigma_{i+1},\cdots,\sigma_n)}(y_i)\Big).$$

We note that there is precisely one SPNE for any sequential game because selection functions pick out only one acceptable move. In trying to find a subgame perfect strategy profile, the strategy in the final round is therefore fixed and, therefore the strategy in the penultimate round also fixed, and so on. Consequently, '*the* SPNE play' of a sequential game is well-defined.

We now note the significance of the independent tensor of selection functions. It computes the SPNE play of a sequential game as described in the following theorem.

Theorem 2.6.1.5. *Let* $\mathcal{G} = \big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, q\big)$ *be a sequential game. Then* $\Big(\bigotimes_{i=1}^n \varepsilon_i\Big)(q)$ *is the SPNE play of* $\mathcal{G}$. □

We can relate this result to the utility maximising agents of classical game theory as follows.

Example 2.6.1.6. For $1 \leq i \leq n$, suppose that $\varepsilon_i : \mathcal{J}_{\mathbb{R}^n}(X_i)$ is such that

$$\varepsilon_i(k : X_i \to \mathbb{R}^n) \in \arg\max(\pi_i \circ k)$$

for all $k : X_i \to \mathbb{R}^n$. Then, letting the i^{th} projection of k corresponds to the utility given to player i, the tensor of selection functions $\Big(\bigotimes_{i=1}^n \varepsilon_i\Big)\big(k : \prod_{i=1}^n X_i \to \mathbb{R}^n\big)$ returns the SPNE play of the sequential game $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, k\big)$ in the traditional game theoretic sense.

2.6.2 Limitations

The single-valued nature nature of the selection functions presented in this chapter restrict the theory to computing only one SPNE play, whilst more general extensive form games may have many SPNE strategy profiles and plays. This is the main motivation in considering the multi-valued generalisation of selection functions in the next chapter. An ideal situation would be a theory of generalised selection functions of the type $(X \to \mathbb{R}) \to \mathcal{P}(X)$ and a theorem showing that a tensor of such selection functions computes the set of SPNE plays of a sequential game. We shall see that the reality is not so neat, but that some interesting results can be salvaged.

Chapter 3

Generalised selection functions

In the previous chapter we modelled players in a sequential game with selection functions $\varepsilon : \mathcal{J}_R(X) = (X \to R) \to X$. In this chapter we consider *generalised selection functions* with type $(X \to R) \to TX$ where T is some strong monad over the underlying category and R is an algebra of T. In particular we are interested in the case where T is the non-empty finite powerset monad $\mathcal{P}_{\mathrm{f}}$ which we use to model nondeterminism. Working with generalised selection functions of the form $(X \to R) \to \mathcal{P}_{\mathrm{f}}(X)$ (which we call *multi-valued* or *nondeterministic* selection functions) allows us to take seriously the multi-valued-ness of arg max and, in doing so, model game theoretic agents which may be able to maximise their utility in multiple ways.

One might conjecture that, as single-valued selection functions compute a subgame perfect play, multi-valued selection functions compute the set of subgame perfect plays. This turns out to not be the case, and for fundamental reasons. On a technical level, this reason is that the algebra of the monad destroys much of the fine-grained information about a sequential game. From a game theoretic perspective the reason is that, if a player has multiple ways to maximise their outcome, then other players acting earlier in the game are acting under possibilistic uncertainty regarding how this later player will maximise. These players acting earlier do not know which plays they would be deviating from if they do choose to deviate from a strategy profile.

We will show that multi-valued selection functions do compute a kind of solution for sequential games, and that these solutions are those appropriate to games with possibilistic uncertainty.

Generalised selection functions have been studied in the context of proof theory in the paper [EO14], but the work in this chapter constitutes the first concerted attempt at an application of generalised selection functions to game theory.

3.1 Chapter overview

3.2 introduces *generalised selection functions* which have type $(X \to R) \to T(X)$ where T is a strong monad and R is a T-algebra; 3.3 is a discussion of how non-deterministic uncertainty can arise in deterministic games; 3.4 specialises generalised selection functions to the case of the finite non-empty powerset monad; 3.5 introduces *nondeterministic sequential games*; 3.6 defines two classes of particularly well-behaved generalised selection functions; 3.7 specifies an appropriate notion of rationality for two-player nondeterministic sequential games; 3.8 proves an important negative result about nondeterministic selection functions and subgame perfect Nash equilibria; 3.9 generalises rationality from two-player games to n-player games; and, finally, 3.10 provides a positive characterisation of the solution concept appropriate to nondeterministic sequential games.

3.2 Selection functions over a monad

Definition 3.2.0.1 (Generalised selection function)**.** Let $\mathcal{C}$ be a cartesian closed category. Suppose that $X : \mathcal{C}$ is an object, T is a strong monad over $\mathcal{C}$, and R is a T-algebra. A T-selection function is a morphism with type $(X \to R) \to TX$. The type $(X \to R) \to TX$ is denoted by $\mathcal{J}_R^T(X)$.

Recall that we defined algebras of a strong monad as an object R together with a family of morphisms $\{(X \to R) \to (TX \to R) \mid X \in \mathcal{C}\}$ in 2.2.0.6.

As with the selection functions of the previous chapter, generalised selection functions also form a strong monad.

Theorem 3.2.0.2 (Generalised selection functions form a strong monad [EO14])**.** *$\mathcal{J}_R^T$ is a strong monad with structural maps given as follows. The unit map is*

$$\eta_X : X \to \mathcal{J}_R^T(X) = \lambda(x : X).\, \lambda(k : X \to R).\, \eta_X^T(x).$$

Let $f : X \to \mathcal{J}_R^T(Y)$ and $k : Y \to R$. Define

$$\alpha^k : X \to TY$$
$$\alpha^k(x) = f(x)(k)$$

and, for $\varepsilon : \mathcal{J}_R^T(X)$, define

$$\beta^{\varepsilon,k} : TX$$
$$\beta^{\varepsilon,k} = \varepsilon(k^\star \circ \alpha^k).$$

The Kleisli extension $f^\dagger : \mathcal{J}_R^T(X) \to \mathcal{J}_R^T(Y)$ *is then given by*

$$f^\dagger(\varepsilon) = \lambda(k : Y \to R).\ (\alpha^k)^\dagger(\beta^{\varepsilon,k}).$$

$\square$

As $\mathcal{J}_R^T$ is a strong monad it has an associated dependent tensor product.

$$\boxtimes^{\mathcal{J}_R^T} : \mathcal{J}_R^T(X) \times (X \to \mathcal{J}_R^T(Y)) \to \mathcal{J}_R^T(X \times Y).$$

Explicitly this is given as follows. Fix $\varepsilon : \mathcal{J}_R^T(X)$, $\Delta : X \to \mathcal{J}_R^T(Y)$, and $k : X \times Y \to R$. Define

$$f : X \to TY$$
$$f(x) = \Delta(x)\big(k(x, -)\big)$$

and

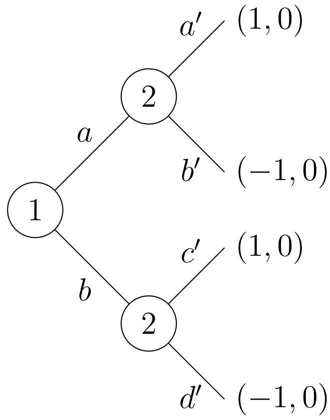
$$a : TX$$
$$a = \varepsilon\Big(\lambda(x : X).\big(k(x, -)^\star(f(x))\big)\Big).$$

Then $\varepsilon \boxtimes^{\mathcal{J}_R^T} \Delta = a \boxtimes^T f$.

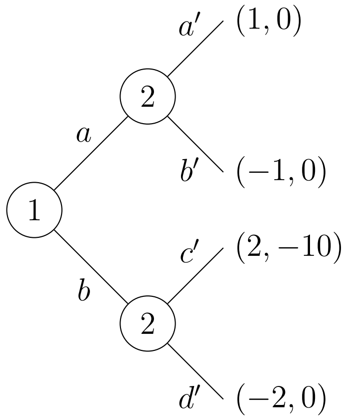
In practice we only need the simpler independent product $\otimes : \mathcal{J}_R^T(X) \times \mathcal{J}_R^T(Y) \to \mathcal{J}_R^T(X \times Y)$ given by taking the argument of type $X \to \mathcal{J}_R^T(Y)$ for $\boxtimes$ to be constant (note that, crucially, the independent product for $\mathcal{J}_R^T$ still makes use of the dependent product for T). We give an explicit definition of this independent product in the case where T is the finite nonempty powerset monad in 3.4.

3.3 Nondeterminism in games

'Nondeterministic sequential game' is a subtle misnomer for the type of game we are interested in. A more accurate name might be 'games with nondeterministic uncertainty,' but the former name is certainly snappier. We are interested in deterministic games during which nondeterministic *uncertainty* can arise, rather than games with nondeterministic components. Consider the following game of two players in extensive form.



Player 2 is entirely indifferent about the outcome of the game, but player 1 is not. Moreover, player 1 cares about which choice player 2 makes but has no way to know which choice he will actually make. Moreover, player 1 has no way of assigning probabilities to player 2's choices and so has no notion of *expected outcome*. Player 1 has to make her choice under profound possibilistic uncertainty about how player 2 will behave. An interesting problem, then, is finding a suitable definition of 'rational play' for player 1. We take the approach that a strategy is rational for player 1 if there is *at least one* way in which player 2 can maximise their outcome such that there is no incentive for player 1 to deviate from the strategy. Player 1 shrugs and guesses at how player 2 might behave. In the example above, both plays are rational for player 1. If player 2 would choose a' and d', then player 1 should choose a. If player 2 would choose b' and c', player 1 should choose b. If player 2 would choose b' and d', player 1's decision is irrelevant. In the game



player 1 should always choose a as player 2 will never choose c' under rational play.

3.4 Nondeterministic selection functions

In this section we specialise generalised selection functions to the finite nonempty powerset monad, which we use to model nondeterminism.

Definition 3.4.0.1 (Finite non-empty powerset monad)**.** The *finite nonempty powerset monad* is given by the following data. The underlying functor $\mathcal{P}_f : \mathbf{Set} \to \mathbf{Set}$ maps a set to the set of its finite, nonempty subsets:

$$\mathcal{P}_f(X) = \left\{ U \in \mathcal{P}(X) \;\middle|\; 0 < |X| < \aleph_0 \right\}.$$

The unit is given by $\eta_X(x) = \{x\}$; the Kleisli extension $f^\dagger : \mathcal{P}_f(X) \to \mathcal{P}_f(Y)$ of a function $f : X \to \mathcal{P}_f(Y)$ is given by $f^\dagger(U) = \bigcup_{x \in U} fx$.

Definition 3.4.0.2. The dependent product $\boxtimes : \mathcal{P}_\mathrm{f}(X) \times (X \to \mathcal{P}_\mathrm{f}(Y)) \to \mathcal{P}_\mathrm{f}(X \times Y)$ for $\mathcal{P}_\mathrm{f}$ is explicitly given by

$$U \boxtimes f = \left\{ (x, y) \in X \times Y \;\middle|\; x \in U, y \in f(x) \right\}$$

The algebras of the finite non-empty powerset monad are *join semilattices.*

Definition 3.4.0.3 (Join semilattice)**.** A *join semilattice* is a partial ordering $(R, \leq)$ such that every finite non-empty subset S of R has a least upper bound (or supremum) $\bigvee S$.

Given an assignment of joins to subsets, the partial ordering can be recovered by setting $r \leq s$ if and only if $\bigvee\{r, s\} = s$. Given a semilattice $(R, \bigvee)$, the associated algebra mapping of a function $f : X \to R$ is given by

$$\begin{aligned} f^\star : \mathcal{P}(X) &\to R \\ U &\mapsto \bigvee\{fx : x \in U\}. \end{aligned}$$

Definition 3.4.0.4. A *multi-valued* or *nondeterministic selection function* is a $\mathcal{P}_\mathrm{f}$-selection function.

We will be using the independent product of multi-valued selection functions extensively and so we explicitly unpack its definition here.

Definition 3.4.0.5 (Independent product of nondeterministic selection functions)**.** Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X), \delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$, and $k : X \times Y \to R$. Then

$$(\varepsilon \otimes \delta)(k) = \{(x, y) \in X \times Y : x \in a, y \in f(x)\}$$

where $f : X \to \mathcal{P}(Y)$ is given by

$$x \mapsto \delta\big(k(x, -)\big)$$

and $a : \mathcal{P}(X)$ is given by

$$a = \varepsilon\Big(\lambda(x : X). \bigvee_{y \in \delta(q(x,-))} q(x, y)\Big).$$

3.5 Nondeterministic sequential games

In this section we modify some of the definitions from chapter 2 so they make sense in a nondeterministic setting. This amounts, for the most part, to replacing the single-valued selection functions of the previous chapter with multi-valued selection functions. We abstain from defining nondeterministic subgame perfect Nash equilibria until 3.8 where it will be more relevant.

Definition 3.5.0.1 (Nondeterministic sequential game)**.** An n-round *nondeterministic sequential game* consists in a tuple of multi-valued selection functions $(\varepsilon_1, \cdots, \varepsilon_n)$ where $\varepsilon_i \in \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X_i)$ together with an *outcome function* $q : \prod_{i=1}^n X_i \to R$. We refer to a nondeterministic sequential game by the triple $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, q\big)$.

Throughout the rest of this chapter we will let $\mathcal{G}$ denote the arbitrary nondeterministic sequential game $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, q\big)$. We also, from hereon, often refer to nondeterministic sequential games simply as *sequential games*, or even just as *games*.

The definition of a *strategy* for a nondeterministic sequential game is entirely unchanged.

Definition 3.5.0.2 (Nondeterministic strategy)**.** Let $\mathcal{G}$ be a game. A *strategy* at round i is a function $\big(\sigma_i : \prod_{j<i} X_j\big) \to X_i$. A *strategy profile* for $\mathcal{G}$ is a tuple $\sigma = (\sigma_1, \cdots, \sigma_n)$ where each σ_i is a strategy for round i.

Just as single-valued selection functions were thought of as returning a satisfactory play given a context, so multi-valued selection functions are thought as providing the *set* of acceptable plays in a context.

3.6 Well-behaved selection functions

We will now specify 'niceness' constraints for multi-valued selection functions. The 'niceness' of a multivalued selection function relates to its interaction with the semilattice R. These constraints are chosen as they pick out a subclass of nondeterministic selection functions which are particularly relevant to sequential games.

Definition 3.6.0.1 (Witnessing selection function)**.** A multi-valued selection function $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ is *witnessing* if for all indexing functions $I : X \to \mathcal{P}_\mathrm{f}(X \to R)$ and all

$$x \in \varepsilon\Bigg(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\Bigg)$$

there exists a choice function $p_{\_} : X \to (X \to R)$ for I (so $p_{x'} \in I(x')$ for all x') such that

$$x \in \varepsilon(\lambda(x' : X).p_{x'}(x')).$$

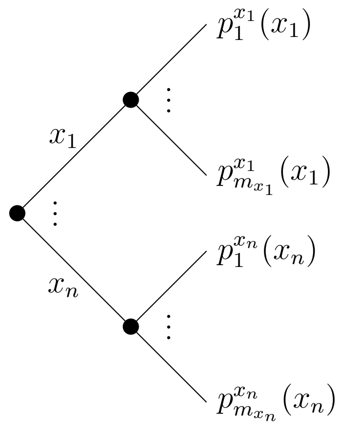
$I(x)$ is thought of as the set of contexts that might arise if x is chosen. The choice function $p_{\_}$ picks out a 'plausible scenario,' a possible context for each choice $x \in X$ that could be made. In game theoretic terms, a witnessing selection function represents a player that finds a move x acceptable to play only if there is some plausible hypothesis regarding how later players will behave under which x is an acceptable move.

Definition 3.6.0.2 (Upwards closed selection function)**.** A multi-valued selection function $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ is *upwards closed* if, whenever $p_{\_} : X \to (X \to R)$ is a choice function for some indexing function $I : X \to \mathcal{P}_f(X \to R)$ such that $x \in \varepsilon(\lambda x'.p_{x'}(x'))$, it holds that

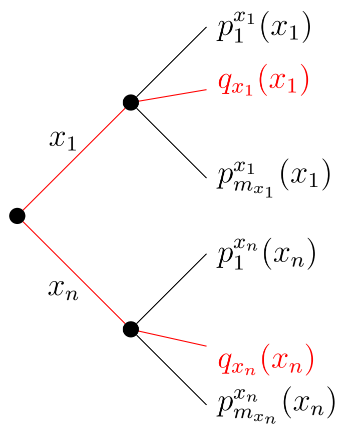
$$x \in \varepsilon\Bigg(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\Bigg).$$

Upwards closure is a converse notion to witnessing. If x is an acceptable choice, then x remains an acceptable choice in contexts where other possible contexts are added and then combined with the join operator (this notion is, admittedly, more game theoretically vague but its interpretation will become clearer in the case where $R = \mathcal{P}_\mathrm{f}(\mathbb{R})$ and the semilattice join is given by union).

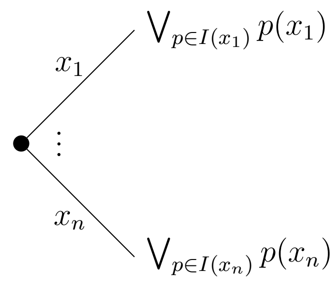
A good heuristic for thinking about witnessing and upwards closed selection functions is as follows. Suppose $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ and that $X = \{x_1, \cdots, x_n\}$ and let $I : X \to \mathcal{P}_\mathrm{f}(X \to R)$ with $I(x) = \{p_1^x \cdots, p_{m_x}^x\}$. We can organise this information in a game tree:



A choice function $q_- : X \to (X \to R)$ for I then corresponds to choosing a leaf of this tree for each $x \in X$. Visually (omitting dots for clarity),



The red subtree then corresponds to the context $\lambda(x : X).q_x x$. In contrast, the context $\lambda(x : X).\bigvee_{p \in I(x)} p(x)$ corresponds to the collapsed game tree



A witnessing selection function is a selection function where, if x is an acceptable play in the collapsed tree, there is some choice of leaves such that x is an acceptable play in the associated context. An upwards closed selection function has the converse property: if there is a choice of leaves under which x is an acceptable play, then x is an acceptable play in the collapsed tree.

Example 3.6.0.3. We will show that arg max is witnessing but not upwards closed. For a finite set X, define $\arg\max : (X \to \mathbb{R}) \to \mathcal{P}_{\mathrm{f}}(X)$ by

$$\arg\max(k) = \Big\{ x \in X \;\Big|\; \forall x' \in X \quad k(x) \geq k(x') \Big\}.$$

arg max is then a multi-valued selection function with the join operator on $\mathbb{R}$ given by max.

Claim 1: arg max is witnessing. *Proof:* Suppose

$$x \in \arg\max_{x' \in X} \Big(\max_{p \in I(x')} (p(x')) \Big)$$

for some $I : X \to \mathcal{P}_{\mathrm{f}}(X \to \mathbb{R})$. Then $\forall x' \in X$ it holds that

$$\max_{p \in I(x)} p(x) \geq \max_{p \in I(x')} p(x').$$

As $I(x')$ is finite, we can choose $p_{x'} \in I(x')$ such that $p_{x'}(x') = \max_{p \in I(x')} p(x')$. Then

$$x \in \arg\max \Big(\lambda(x' : X).p_{x'}(x') \Big).$$

Hence arg max is witnessing.

Claim 2: arg max is not upwards closed. *Proof:* Let $X = \{0, 1\}$ and let $c_i : X \to \mathbb{R}$ denote the constant function $x \mapsto i$. Define $I : X \to \mathcal{P}(X \to \mathbb{R})$ by

$$\begin{aligned} 0 &\mapsto \{c_0\} \\ 1 &\mapsto \{c_1, c_{-1}\} \end{aligned}$$

Note that the function $\lambda(x : X).\max_{p \in I(x)} p(x)$ is given by

$$\begin{aligned} 0 &\mapsto 0 \\ 1 &\mapsto 1 \end{aligned}$$

and, hence, $\arg\max_{x \in X} \big(\max_{p \in I(x)} p(x) \big) = \{1\}$. Define a choice function $p_{-} : X \to (X \to \mathbb{R})$ for I by $p_0 = c_0$ and $p_1 = c_{-1}$. Then $\arg\max_{x \in X} \big(p_x(x) \big) = \{0\}$, but $\{0\} \not\subseteq \{1\}$ and hence arg max is not upwards closed.

That arg max is witnessing follows from a more general result regarding multi-valued selection functions for which the semilattice R is total.

Proposition 3.6.0.4. *If the semilattice R is total, then for all sets X and all selection functions $\varepsilon : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(X)$, ε is witnessing.*

Proof. Suppose R is total and $I : X \to \mathcal{P}_{\mathrm{f}}(X \to R)$. Then for all $x' \in X$ there exists $p_{x'} \in I(x')$ such that

$$\bigvee_{p \in I(x')} p(x') = p_{x'}(x').$$

Then

$$x \in \varepsilon\Big(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\Big) \Longrightarrow x \in \varepsilon\big(\lambda(x' : X).p_{x'}(x')\big)$$

□

We now consider an example of a multi-valued selection function which is upwards closed but not witnessing.

Example 3.6.0.5. Let $X = \{0, \star\}$. We think of $\star$ as ε's 'favourite move' which is satisfactory in any context. Define a semilattice $R = \{\top, \bot_1, \bot_2\}$ where $\bot_1 \leq \top$, $\bot_2 \leq \top$, and $\bot_1$ and $\bot_2$ are not comparable. Define $\varepsilon : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(X)$ by

$$\varepsilon(p) = \{\star\} \cup \{x \in X \mid p(x) = \top\}.$$

Suppose that $x \in \varepsilon(\lambda x'.p_{x'}(x'))$ where $p_{x'} \in I(x')$ for some $I : X \to \mathcal{P}_{\mathrm{f}}(X \to R)$. Then either $x = \star$ or $p_x(x) = \top$. In either case, $x \in \varepsilon\Big(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\Big)$. Hence ε is upwards closed.

Conversely, define an indexing function given by $I(0) = I(\star) = \{c_{\bot_1}, c_{\bot_2}\}$ (where the c_i are constant functions as in the previous example). Then $0 \in \varepsilon\Big(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\Big) = X$, but there is no choice function $p_{-} : X \to (X \to R)$ for I such that $0 \in \varepsilon(\lambda(x' : X).p_{x'}(x'))$. Hence ε is not witnessing.

We will see an important example of a multi-valued selection function which is both witnessing and upwards closed in 3.10.

3.7 Rational strategies for nondeterministic games

In this section we define a notion of rationality for nondeterministic games. A strategy profile is *rational* precisely when there is some plausible hypothesis about how later players will behave under which that strategy profile is acceptable. As it sounds, this notion is closely linked to the properties 'witnessing' and 'upwards closed.' We show that witnessing and upwards closed selection functions compute precisely the plays of rational strategy profiles. We start by restricting ourselves to games of length 2, considering an arbitrary game given by

$$\mathcal{G}_2 = \Big((X, Y), \big(\varepsilon : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(X), \delta : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(Y)\big), k : X \times Y \to R\Big).$$

Definition 3.7.0.1 (Rational strategy profile)**.** Let $\mathcal{G}$ be the two-player game specified above. Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$, $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$, and $k : X \times Y \to R$. A strategy $\tau_2 : X \to Y$ is *rational at* round 2 (or *rational for* δ) if

$$\tau_2(x) \in \delta\big(k(x, -)\big)$$

for all $x \in X$. A strategy profile $(\sigma_1 : X, \sigma_2 : X \to Y)$ of $\mathcal{G}$ is *rational* if

1. σ_2 is rational at round 2; and
2. There is τ_2 rational at round 2 such that

$$\sigma_1 \in \varepsilon\Bigg(\lambda(x : X).k\big(x, \tau_2(x)\big)\Bigg).$$

The set of *rational plays* of $\mathcal{G}$ is given by

$$\mathrm{Rat}(\mathcal{G}) = \Big\{(x, y) \in X \times Y \;\Big|\; (x, y) = (\sigma_1, \sigma_2(\sigma_1)) \text{ for some rational } (\sigma_1, \sigma_2)\Big\}.$$

Remark 3.7.0.2. If $(x, y) \in (\varepsilon \otimes \delta)(k)$ where ε, δ, and k are as above, then the definition of the tensor of selection functions tells us that

$$x \in \varepsilon\big(\lambda(x' : X). \bigvee_{y' \in \delta(k(x', -))} k(x', y')\big).$$

In the definitions of witnessing and upwards closed selection functions we were concerned with expressions of the form

$$x \in \varepsilon\big(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\big).$$

It will be worthwhile to spend a few lines on the relationship between these two similar-looking expressions. Given $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$ we can define an indexing function $I_{\delta,k} : X \to \mathcal{P}_\mathrm{f}(X \to R)$ by

$$x' \mapsto \Big\{k(-, y') \;\Big|\; y' \in \delta(k(x', -))\Big\}$$

so that

$$\lambda(x' : X) \bigvee_{y' \in \delta(k(x', -))} k(x', y') \quad = \quad \lambda(x' : X) \bigvee_{p \in I_{\delta,k}(x')} p(x').$$

Moreover, a rational strategy $\sigma_2 : X \to Y$ for δ induces a choice function for $I_{\delta,k}$ via the mapping $x \mapsto k(-, \sigma_2(x))$. The converse does not quite hold. If $k(-, y) = k(-, y')$ but $y \neq y'$, then a choice function does not uniquely induce a rational strategy for δ.

The consequences of these remarks are worked out in the following lemma and theorem.

Definition 3.7.0.3. Let $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$ and $k : X \times Y \to R$. Define $I_{\delta,k} : X \to \mathcal{P}_\mathrm{f}(X \to R)$ by

$$x \mapsto \left\{ k(-,y) \;\middle|\; y \in \delta\big(k(x,-)\big) \right\}.$$

Lemma 3.7.0.4. *Let $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$ and $k : X \times Y \to R$. Then $p_- : X \to (X \to R)$ is a choice function for $I_{\delta,k}$ if and only if there exists $\sigma_2 : X \to Y$, rational for δ, such that*

$$\lambda(x : X).p_x(x) = \lambda(x : X).k(x, \sigma_2(x)).$$

□

Theorem 3.7.0.5. *Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ be a multi-valued selection function. The following equivalences hold.*

1. *ε is witnessing if and only if for any $k : X \times Y \to R$ and $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$ it holds that $(\varepsilon \otimes \delta)(k) \subseteq \mathrm{Rat}(\mathcal{G})$ where $\mathcal{G} = \big((X,Y),(\varepsilon,\delta),k\big)$.*
2. *ε is upwards closed if and only if for any $k : X \times Y \to R$ and $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$ it holds that $\mathrm{Rat}(\mathcal{G}) \subseteq (\varepsilon \otimes \delta)(k)$ (with $\mathcal{G}$ as above).*

Proof. We first prove the forward directions of both equivalences which follow quickly from the above lemma.

1. ⇒:

Suppose that ε is witnessing and $(x,y) \in (\varepsilon \otimes \delta)(k)$. Then

$$x \in \varepsilon\Bigg(\lambda(x' : X). \bigvee_{y' \in \delta(k(x',-))} k(x',y')\Bigg) = \varepsilon\Bigg(\lambda(x' : X) \bigvee_{p \in I_{\delta,k}(x')} p(x')\Bigg)$$

and

$$y \in \delta\big(k(x,-)\big).$$

By the previous lemma and the fact that ε is witnessing, there exists $\sigma_2 : X \to Y$, rational for δ, such that

$$x \in \varepsilon\big(\lambda(x' : X).k(x', \sigma_2(x'))\big).$$

Then the strategy profile (x, τ_2) where

$$\tau_2(x') = \begin{cases} y & \text{if } x' = x \\ \sigma_2(x') & \text{otherwise} \end{cases}$$

is rational with play (x,y).

2. $\Rightarrow$:

Suppose that ε is upwards closed and (σ_1, σ_2) is rational. Then $\sigma_2(x) \in \delta\big(k(x,-)\big)$ for all $x \in X$. In particular, $\sigma_2(\sigma_1) \in \delta\big(k(\sigma_1,-)\big)$.

By rationality there exists $\tau_2 : X \to Y$, rational for δ, such that $\sigma_1 \in \varepsilon\big(\lambda(x : X).k(x, \tau_2(x))\big)$. By 3.7.0.4 and the fact that ε is upwards closed,

$$x \in \varepsilon\left(\lambda(x' : X) \bigvee_{p \in I_{\delta,k}(x')} p(x')\right) = \varepsilon\left(\lambda(x' : X). \bigvee_{y' \in \delta(k(x',-))} k(x', y')\right).$$

Hence $(\sigma_1, \sigma_2(\sigma_1)) \in (\varepsilon \otimes \delta)(k)$.

For the converse directions of the two equivalences we construct a pathological counter example and prove the contrapositives. Define the context $k : X \times (X \to R) \to R$ to be function application $(x, p) \mapsto p(x)$. Given an indexing function $I : X \to \mathcal{P}_{\mathrm{f}}(X \to R)$ define $\delta_{I,k} : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(X \to R)$ by

$$\delta_{I,k}(p) = \begin{cases} I(x') & \text{if } p = k(x',-) \\ \text{arbitrary} & \text{otherwise.} \end{cases}$$

Note that $k(x,-) = k(x',-)$ if and only if $x = x'$ or $|R| < 2$. In the latter case the theorem holds vacuously for $R = \varnothing$ and, for $|R| = 1$, we have that $|X \to R| = 1$ and so $I(x) = I(x')$ for all $x, x' \in X$. Consequently, $\delta_{I,k}$ is well-defined.

1. $\Leftarrow$:

Suppose ε is not witnessing. Then there is some indexing function $I : X \to \mathcal{P}_{\mathrm{f}}(X \to R)$ and $x \in \varepsilon\big(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\big)$ such that there is no choice function $p_- : X \to (X \to R)$ for I such that $x \in \varepsilon(\lambda(x' : X).p_{x'}(x'))$. By construction,

$$\lambda(x' : X). \bigvee_{p \in I(x')} p(x') = \lambda(x' : X). \bigvee_{p \in \delta_I(k(x',-))} k(x, p).$$

Then, by the definition of the tensor of selection functions, $(x, p) \in (\varepsilon \otimes \delta_{I,k})(p)$ for any $p \in \delta_{I,k}(k(x,-))$. By hypothesis there is no choice function $p_- : X \to (X \to R)$ for I such that $x \in \varepsilon(\lambda(x' : X).p_{x'}(x'))$ and hence there are no rational strategy profiles with play (x, p). Hence $(\varepsilon \otimes \delta_{I,k})(p) \not\subseteq \mathrm{Rat}(\mathcal{G})$.

2. $\Leftarrow$:

Suppose ε is not upwards closed. Then there is $I : X \to \mathcal{P}_{\mathrm{f}}(X \to R)$ and a choice function $p_- : X \to (X \to R)$ for I such that there is $x \in \varepsilon(\lambda(x' : X).p_{x'}(x'))$ and

$$x \notin \varepsilon\left(\lambda(x' : X). \bigvee_{p \in I(x')} p(x')\right) = \varepsilon\left(\lambda(x' : X). \bigvee_{p \in \delta_{I,k}(k(x',-))} k(x', p)\right).$$

Define $\sigma_2 : X \to (X \to R)$ by $\sigma_2(x') = p_{x'}$. Then (x, σ_2) is rational but $(x, \sigma_2(x)) \notin (\varepsilon \otimes \delta_{I,k})(k)$. $\square$

This theorem has an easy corollary regarding selection functions which are both witnessing and upwards closed.

Corollary 3.7.0.6. *Suppose* $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ *is witnessing and upwards closed. Then for all* $k : X \times Y \to R$ *and all* $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$*,* $(\varepsilon \otimes \delta)(k) = \mathrm{Rat}(\mathcal{G})$*.* □

The property of being witnessing is not closed under the independent product of selection functions. In 3.10 we will see an example where ε and δ are both witnessing and upwards closed, but where $(\varepsilon\otimes\delta)$ is not witnessing. A heuristic for why witnessing fails is that it might be possible to choose witnesses for ε and δ, but be impossible to choose such witnesses simultaneously. The property of being upwards closed *is* closed under the independent product of selection functions. In order to show this, we first prove an easy lemma.

Lemma 3.7.0.7. *Suppose* $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ *is upwards closed and let* $k, k' : X \to R$ *be such that* $k(x) \geq k'(x)$ *for all* $x \in X$*. Then* $x \in \varepsilon(k') \Rightarrow x \in \varepsilon(k)$*.*

Proof. Define $I : X \to \mathcal{P}_\mathrm{f}(X \to R)$ to be the constant function $x \mapsto \{k, k'\}$. Define $p_- : X \to (X \to R)$ to be the constant function $p_x = k'$. Then p_- is a choice function for I. Also,

$$\lambda(x : X).\ \bigvee_{q \in I(x)} q(x) \quad = \quad \lambda(x : X).\ \bigvee \big\{k(x), k'(x)\big\} \quad = \quad k.$$

Hence, $x \in \varepsilon(k') \Rightarrow x \in \varepsilon(k)$. □

Proposition 3.7.0.8. *Suppose* $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ *and* $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$ *are upwards closed. Then* $(\varepsilon \otimes \delta)$ *is upwards closed.*

Proof. Let $p_- : (X \times Y) \to \big((X \times Y) \to R\big)$ be a choice function for $I : (X \times Y) \to \mathcal{P}_\mathrm{f}\big((X \times Y) \to R\big)$ and let $(x, y) \in (\varepsilon \otimes \delta)\Big(\lambda\big((x', y') : X \times Y\big).\, p_{(x',y')}(x', y')\Big)$. By the definition of the selection tensor, we know that

$$x \in \varepsilon\Big(\lambda(x' : X).\ \bigvee_{y' \in \delta(p_{(x',-)}(x',-))} p_{(x',y')}(x', y')\Big) \tag{$\star$}$$

$$y \in \delta\Big(\lambda(y' : Y).\, p_{(x,y')}(x, y')\Big). \tag{$\star\star$}$$

To show that $(\varepsilon \otimes \delta)$ is upwards closed, we need to show that

$$(x, y) \in (\varepsilon \otimes \delta)\Big(\lambda\big((x', y') : X \times Y\big).\ \bigvee_{k \in I(x',y')} k(x', y')\Big).$$

That is, we need to show

$$x \in \varepsilon\Big(\lambda(x' : X).\ \bigvee_{y' \in A(x')} \bigvee_{k \in I(x', y')} k(x', y')\Big) \tag{1}$$

where $A(x') = \delta\big(\lambda(y' : Y).\ \bigvee_{k \in I(x', y')} k(x', y')\big)$, and

$$y \in \delta\Big(\lambda(y' : Y).\ \bigvee_{k \in I(x, y')} k(x, y')\Big). \tag{2}$$

(2): Define $I_x : Y \to \mathcal{P}_{\mathrm{f}}(Y \to R)$ by $y' \mapsto \{k(x, -) \mid k \in I(x, y')\}$ and note that $y' \mapsto \lambda(y'' : Y).\ p_{(x,y')}(x, y'')$ is a choice function for I_x. Moreover, $I_x(y') = I(x, y')$. By ($\star\star$) and upwards closure of δ,

$$y \in \delta\Big(\lambda(y' : Y).\ \bigvee_{k \in I_x(y')} k(y')\Big) \quad = \quad \delta\Big(\lambda(y' : Y).\ \bigvee_{k \in I(x, y')} k(x, y')\Big).$$

(1): Using ($\star$) and 3.7.0.7, it suffices to show that

$$\bigvee_{y' \in \delta(p_{(x',-)}(x',-))} p_{(x',y')}(x', y') \quad \leq \quad \bigvee_{y' \in A(x')} \bigvee_{k \in I(x', y')} k(x', y')$$

for all $x' \in X$. It is therefore sufficient to prove that

$$\Big\{p_{(x',y')}(x', y') \ \Big|\ y' \in \delta\big(p_{(x',-)}(x', -)\big)\Big\} \subseteq \Big\{k(x', y') \ \Big|\ y' \in A(x'),\ k \in I(x', y')\Big\}$$

for all $x' \in X$. If $p_{(x',y')}(x', y')$ is an element of the left-hand side, then $y' \in A(x')$ by the upwards closure of δ. Also, as p_{-} is a choice function for I, we have that $p_{(x',y')} \in I(x', y')$. Hence $p_{(x',y')}(x', y')$ is also an element of the right-hand side. □

3.8 Relation to subgame perfect Nash equilibria

This section concerns a negative result: multi-valued selection functions cannot, in general, compute the set of plays of subgame perfect strategies. We can generalise the definition of *subgame perfect Nash equilibrium* from 2.6.1 as follows.

Definition 3.8.0.1 (Multi-valued subgame perfect strategy profile)**.** Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(X)$, $\delta : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(Y)$, and $k : X \times Y \to R$. A strategy profile $(\sigma_1 : X, \sigma_2 : X \to Y)$ for the game $\mathcal{G}_2 = \big((X, Y), (\varepsilon, \delta), k\big)$ is a *subgame perfect Nash equilibrium* (we also say that (σ_1, σ_2) is *subgame perfect*) if

1. $\sigma_1 \in \varepsilon\big(\lambda(x : X).\ k(x, \sigma_2(x))\big)$; and

2. for all $x \in X$, $\sigma_2(x) \in \delta\big(\lambda(y : Y).\, k(x, y)\big)$.

The set of *subgame perfect plays* of $\mathcal{G}$ is

$$\mathrm{SP}(\mathcal{G}_2) = \Big\{(\sigma_1, \sigma_2(\sigma_1)) \in X \times Y \;\Big|\; (\sigma_1, \sigma_2) \text{ is subgame perfect}\Big\}.$$

Remark 3.8.0.2. Note that subgame perfect strategy profiles are rational: subgame perfect strategy profiles are those where ε's guess about δ's future behaviour is correct.

Theorem 3.8.0.3. *Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$. If, for all sets Y, selection functions $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$, and functions $k : X \times Y \to R$ it holds that $(\varepsilon \otimes \delta)(k) = \mathrm{SP}(\mathcal{G}_{\delta,k})$, where $\mathcal{G}_{\delta,k} = \big((X, Y), (\varepsilon, \delta), k\big)$, then ε is constant.*

Proof. The proof proceeds by contradiction. Suppose that for all δ and k, $(\varepsilon \otimes \delta)(k) = \mathrm{SP}(\mathcal{G}_{\delta,k})$, and that ε is not constant. That is, there exist $p_1, p_2 : X \to R$ and $x \in X$ such that $x \in \varepsilon(p_1)$ and $x \notin \varepsilon(p_2)$. Let $q : X \times (X \to R) \to R$ be the function application operator, $(x, p) \mapsto px$. Define $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X \to R)$ by

$$\delta(p) = \begin{cases} \{p_1, p_2\} & p = q(x, -) \\ \{p_1\} & \text{otherwise.} \end{cases}$$

As $p_1 \neq p_2$, we have that $|R| > 1$. Consequently, $q(x, -) = q(x', -)$ if and only if $x = x'$. Moreover, $x' \neq x$ implies that $\delta(q(x', -)) = \{p_1\}$. Consider the play (x, p_2) of $\mathcal{G}_{\delta,q}$ noting that, by construction, (x, p_2) is not the play of any subgame perfect strategy profile. Define $p_- : X \to (X \to R)$ to be the constant mapping $p_{x'} = p_1$ so that $x \in \varepsilon(\lambda x'.\, p_{x'} x') = \varepsilon(p_1)$.

As all subgame perfect plays are rational, we have that ε is upwards closed by 3.7.0.5. Hence $(x, p_2) \in (\varepsilon \otimes \delta)(q)$, but we have already established that (x, p_2) is not a subgame perfect play.

□

This proof emphasizes the point that multi-valued selection functions fail to compute subgame perfect plays because players in a sequential game can be *indifferent* between two choices. In the case where x is played, δ is indifferent between playing p_1 or p_2 whilst ε is not. In games where there is no such conflicting indifference, witnessing and upwards closed selection functions *do* compute the set of subgame perfect plays.

Definition 3.8.0.4. Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$, $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$, and $q : X \times Y \to R$. We say that the game $\big((X, Y), (\varepsilon, \delta), k\big)$ has *coinciding indifference* if, for all $x \in X$ and $y, y' \in Y$,

$$y, y' \in \delta\big(k(x, -)\big) \Longrightarrow \varepsilon\big(k(-, y)\big) = \varepsilon\big(k(-, y')\big)$$

Proposition 3.8.0.5. *Suppose* $\big((X,Y),(\varepsilon,\delta),k\big)$ *has coinciding indifference and that* ε *is witnessing and upwards closed. Then* $(\varepsilon \otimes \delta)(k) = \mathrm{SP}(k,\varepsilon,\delta) = \mathrm{Rat}(k,\varepsilon,\delta)$.

Proof. Let $(x,y) \in (\varepsilon \otimes \delta)(k)$. By 3.7.0.5, (x,y) is the play of some rational strategy profile (σ_1, σ_2). Then there exists some function $y(-) : X \to Y$ where, for all $x' \in X$, $y(x') \in \delta\big(q(x',-)\big)$ and $\sigma_1 \in \varepsilon(\lambda x'.\, q(x', y(x')))\big)$. By coinciding indifference, $\sigma_1 \in \varepsilon\big(\lambda x'.q(x', \sigma_2 x')\big)$.

Conversely, subgame perfect plays are rational. Hence, if (x,y) is a subgame perfect play, then $(x,y) \in (\varepsilon \otimes \delta)(q)$ by 3.7.0.5. □

To summarize, in a two round game with first player ε, $(\varepsilon \otimes \delta)(k)$ computes subgame perfect plays for arbitrary second player δ and arbitrary context k if and only if ε is constant. $(\varepsilon \otimes \delta)(k)$ *does* compute subgame perfect plays in the special cases where ε is upwards closed and witnessing, and (q, ε, δ) has coinciding indifference.

3.9 Finite length nondeterministic sequential games

In this section we generalise the results concerning games of length 2 to games of arbitrary finite length. The main proof strategy is induction on the results of the previous sections.

Notation 3.9.0.1. Given $A \subseteq \bigcup_{i=1}^{n} X_i$, we use $A^{(j)}$ to denote $X_j \cap A$.

In particular, if Γ is a set of strategies for some sequential game $\mathcal{G}$, then $\Gamma^{(j)}$ denotes the set of strategies in Γ which are strategies for round j.

In the two player case, if a strategy profile (σ_1, σ_2) is rational with respect to $\big((X,Y),(\varepsilon,\delta),k\big)$, there is some rational strategy $\tau_2 : X \to Y$ for δ such that σ_1 is acceptable to ε if δ plays according to τ_2. To generalise to the n-round case, we can simply extend this heuristic as follows. Given a game $\big(\big(X_i\big)_{i=1}^{n}, \big(\varepsilon_i\big)_{i=1}^{n}, k\big)$, a strategy $\sigma_1 : X_1$ is rational if there are strategies $\sigma_2, \cdots, \sigma_n$, rational for $\varepsilon_2, \cdots, \varepsilon_n$ respectively, under which σ_1 is a good move. For players ε_i acting in the 'mid-game,' a strategy is rational if it is rational for all subgames given by partial plays $x \in \prod_{j=1}^{i-1} X_j$.

We define a more general notion of sets of strategies as *consistent* for a game $\mathcal{G}$. The set of rational strategy profiles will then be realised as the maximal consistent set of strategy profiles.

Definition 3.9.0.2. Let Γ be a set of strategies for a sequential game $\mathcal{G} = \big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, k\big)$. Γ is $\mathcal{G}$-*consistent* if for all $i < n$ and $\sigma_i \in \Gamma^{(i)}$, and all partial plays $x \in \prod_{j=1}^{i-1} X_j$, there exists $\sigma = (\sigma_{i+1}, \cdots, \sigma_n)$ where $\sigma_{i+1}, \cdots, \sigma_n \in \Gamma$ such that

$$\sigma_i(x) \in \varepsilon_i\Bigg(\lambda(y : X_i).k\big((x, y)^\sigma\big)\Bigg)$$

where $(x, y)^\sigma$ is the strategic extension of $(x_1, \cdots, x_{i-1}, y)$ by $(\sigma_{i+1}, \cdots, \sigma_n)$.

Note that if Γ is $\mathcal{G}$-consistent, the $\mathcal{G}$-consistency of $\Gamma \cup \{\sigma_i\}$ depends only on $\Gamma^{(j)}$ for $j > i$. With that in mind, we can define the maximal $\mathcal{G}$-consistent set of strategies, denoted by $\Sigma(\mathcal{G})$, as follows.

Definition 3.9.0.3. $\Sigma(\mathcal{G})$ is given by

$$\Sigma(\mathcal{G})^{(n)} = \Big\{\sigma_n : \prod_{i<n} X_i \to X_n \;\Big|\; \forall x \in \prod_{i=1}^{n-1} X_i.\ \sigma_n(x) \in \varepsilon_n\big(q(x, -)\big)\Big\}$$
$$\Sigma(\mathcal{G})^{(i)} = \Big\{\sigma_i : \prod_{j<i} X_j \to X_i \;\Big|\; \{\sigma_i\} \cup \bigcup_{j>i} \Sigma(\mathcal{G})^{(j)} \text{ is } \mathcal{G}\text{-consistent}\Big\}.$$

Definition 3.9.0.4. Let Γ be a set of strategies for a sequential game $\mathcal{G}$. A play $x \in \prod_{i=1}^n X_i$ is a Γ *play* if x is the strategic play of a strategy profile $\sigma = (\sigma_1, \cdots, \sigma_n)$ where $\sigma_i \in \Gamma^{(i)}$ for each $i \leq n$.

The following lemma and corollary show that adding ineffectual players to a game does not impact which choices are considered acceptable.

Lemma 3.9.0.5. *Let $k : X \to R$ and define $k' : X \times Y \to R$ by $k'(x, y) = k(x)$. Then, for all $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ and $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$,*

$$x \in \varepsilon(k) \Leftrightarrow \exists y \in Y \text{ such that } (x, y) \in (\varepsilon \otimes \delta)(k').$$

Proof. If $x \in \varepsilon(k)$ then, for all $y \in Y$,

$$x \in \varepsilon(k) = \varepsilon\Big(\lambda(x' : X). \bigvee_{y \in \delta(k(x'))} k(x')\Big) = \varepsilon\Big(\lambda(x' : X). \bigvee_{y \in \delta(k'(x', -))} k'(x', y)\Big).$$

Hence if $y \in \delta\big(k(x, -)\big)$ then $(x, y) \in (\varepsilon \otimes \delta)(k')$.

Conversely,

$$(x, y) \in (\varepsilon \otimes \delta)(k') \Rightarrow x \in \varepsilon\Big(\lambda(x' : X). \bigvee_{y' \in \delta(k'(x', -))} k'(x', y')\Big) = \varepsilon(k)$$

$\square$

Corollary 3.9.0.6. *Let $\big((X_i)_{i=1}^n, (\varepsilon_i)_{i=1}^n, k\big)$ be a sequential game. Suppose there exists $j < n$ and $k_j : X_i \times X_n \to R$ such that, for all $x \in \prod_{i=1}^n X_i$, $k(x) = k_j(x_j, x_n)$. Then $(x_j, x_n) \in (\varepsilon_j \otimes \varepsilon_n)(k_j)$ if and only if there exist $x_1, \cdots, x_{j-1}, x_{j+1}, \cdots, x_{n-1}$ with each $x_j \in X_j$ such that $(x_1, \cdots, x_n) \in \Big(\bigotimes_{i=1}^n \varepsilon_i\Big)(k)$.*

Proof. The proof proceeds by a routine induction on n, noting that the case $n = 2$ is trivial. When $j \neq 1$ the result follows easily by choosing

$$x_1 \in \varepsilon_1\Big(\lambda(y_1 : X_1). \bigvee_{(y_2,\cdots,y_n)\in A(y_1)} k(y_1, \cdots, y_n)\Big)$$

where $A(y_1) = (\bigotimes_{i=2}^m \varepsilon_i)(k(y_1, -))$ and applying the inductive hypothesis to the game $\big((X_i)_{i=2}^n, (\varepsilon_i)_{i=2}^n, k(x_1, -)\big)$. For the case $j = 1$, note that the value of the function

$$\lambda(y_1 : X_1)., \cdots, \lambda(y_n : X_n). \bigvee_{y_n \in \varepsilon_n(k(y_1,\cdots,y_{n-1},-))} k(y_1, \cdots, y_n)$$

depends only on y_1. Then, using 3.9.0.5,

$$\begin{aligned}
&(x_1, x_n) \in (\varepsilon_1 \otimes \varepsilon_n)(k_1)\\
\Leftrightarrow & x_1 \in \varepsilon_1\Big(\lambda(y_1 : X_1). \bigvee_{y_n \in \varepsilon_n(k_1(y_1,-))} k_1(y_1, y_n)\Big) \text{ and } x_n \in \varepsilon_n\big(k_1(x_1, -)\big)\\
\Leftrightarrow & \exists x \in \prod_{i=2}^{n-1} X_i.\ (x_1, x) \in \Big(\bigotimes_{i=1}^{n-1} \varepsilon_i\Big)\Big(\lambda y_1, y. \bigvee_{y_n \in \varepsilon_n(k(y_1,y,-))} k(y_1, y, y_n)\Big)\\
&\text{and } x_n \in \varepsilon_n\big(k(x_1, x, -)\big)\\
\Leftrightarrow & \exists x \in \prod_{i=1}^{n-1} X_i.\ (x_1, x, x_n) \in \Big(\bigotimes_{i=1}^{n} \varepsilon_i\Big)(k).
\end{aligned}$$

□

The following theorem generalises 3.7.0.5 to games of arbitrary finite length.

Theorem 3.9.0.7. *Let $\varepsilon_i : \mathcal{J}_R^{\mathcal{P}_f}(X_i)$ for $i < n$. For all sets X_n, selection functions $\varepsilon_n : \mathcal{J}_R^{\mathcal{P}_f}(X_n)$, and contexts $k : \prod_{i=1}^n X_i \to R$, the following equivalences hold.*

1. *ε_i is witnessing for each $i < n$ if and only if $\Big(\bigotimes_{i=1}^n \varepsilon_i\Big)(k)$ is a subset of the set of $\Sigma(\mathcal{G})$ plays.*

2. ε_i is upwards closed for each $i < n$ if and only if $\left(\bigotimes_{i=1}^n \varepsilon_i\right)(k)$ is a superset of the set of $\Sigma(\mathcal{G})$ plays.

Proof. We prove the forward directions of the two equivalences first. The proof proceeds by induction on n, noting that the cases $n = 1$ are trivial.

(1) : Suppose $x = (x_1, \cdots, x_n) \in \left(\bigotimes_{i=1}^n \varepsilon_i\right)(k)$. As ε_1 is witnessing, it is the play of some rational strategy profile $(x_1, f : X_1 \to \prod_{i=2}^n X_i)$ of the two round game $\big((X_1, \prod_{i=2}^n X_i), (\varepsilon_1, \bigotimes_{i=2}^n \varepsilon_i), k\big)$. By hypothesis we have that

$$\left(\bigotimes_{i=2}^n \varepsilon_i\right)(k(y_1, -))$$

is a subset of the set of $\Sigma(\mathcal{G}^{y_1})$ plays for all $y_1 \in X_1$ where

$$\mathcal{G}^{y_1} = \big((X_i)_{i=2}^n, (\varepsilon_i)_{i=2}^n, k(y_1, -)\big)$$

Hence $f(y_1)$ is the play of some $\Sigma(\mathcal{G}^{y_1})$-consistent strategy profile σ^{y_1} for all $y_1 \in X_1$. Then the strategy profile τ for $\mathcal{G}$ given by

$$\tau_1 = x_1$$
$$\tau_{i+1}(y_1, \cdots, y_i) = \sigma_{i+1}^{y_1}(y_2, \cdots, y_i)$$

is such that $\tau_i \in \Sigma(\mathcal{G})$ for all i and the play of τ is x.

(2) : Suppose that $x = (x_1, \cdots, x_n)$ is the $\Sigma(\mathcal{G})$ play of $(\sigma_1, \cdots, \sigma_n)$. A simple check demonstrates that for all $y_1 \in X_1$, we have that $\sigma_2, \cdots, \sigma_n \in \Sigma(\mathcal{G}^{y_1})$. By hypothesis, the strategic play $y_1^{\sigma_{-1}}$ of $(\sigma_2, \cdots, \sigma_n)$ for the game $\mathcal{G}^{y_1}$ is such that

$$y_1^{\sigma_{-1}} \in \left(\bigotimes_{i=2}^n \varepsilon_i\right)\big(k(y_1, -)\big).$$

In particular,

$$(x_2, \cdots, x_n) \in \left(\bigotimes_{i=2}^n \varepsilon_i\right)\big(k(x_1, -)\big). \qquad (\star)$$

As $x_1 = \sigma_1 \in \Sigma(\mathcal{G})$ there exists $\tau = (\tau_2, \cdots, \tau_n)$ with each $\tau_i \in \Sigma(\mathcal{G})$ such that

$$x_1 \in \varepsilon_1\big(\lambda(y_1 : X_1).k(y_1^\tau)\big)$$

and, for all $y_1 \in X_1$,

$$(y_1^\tau)_{-1} \in \left(\bigotimes_{i=2}^n \varepsilon_i\right)\big(k(y_1, -)\big).$$

As ε_1 is upwards closed,

$$x_1 \in \varepsilon_1\left(\lambda(y_1 : X_1). \bigvee_{z \in A(y_1)} k(x_1, z)\right)$$

where $A(y_1) = \left(\bigotimes_{i=2}^{n} \varepsilon_i\right)\big(k(y_1, -)\big)$. From this and $(\star)$ we conclude that

$$x \in \left(\bigotimes_{i=1}^{n} \varepsilon_i\right)(k).$$

As for the backward directions, for $i < n$ consider the construction $\delta_I^i : \mathcal{J}_R^{\mathcal{P}_{\mathrm{f}}}(X_i)$ as in the proof of 3.7.0.5 and let $k_i : \left(\prod_{j=1}^{n-1} X_j\right) \times (X_i \to R) \to R$ be given by $(x, p) \mapsto p(x_i)$. The converse directions are then a corollary of 3.9.0.6 and 3.7.0.5 by considering the game $((\varepsilon_1, \cdots, \varepsilon_{n-1}, \delta_I), k)$ for each i.

□

3.10 Dominating strategies

We have seen that multivalued selection functions do not, in general, compute subgame perfect plays. We have also characterised the plays nondeterministic selection functions *do* compute. In this section we make sense of this, relating it to a solution concept that is already well-known.

3.10.1 Dominance free strategy profiles

Consider a normal form game $\mathcal{N} = \big(S_i, q_i\big)_{i=1}^{n}$. For a given player i there may exist strategies $s_i, s_i' \in S_i$ such that, whatever choices the other players of the game make, choosing s_i results in a strictly better outcome than choosing s_i' for player i. In this instance, we say that the strategy s_i *strictly dominates* the strategy s_i'.

Definition 3.10.1.1 (Dominating strategy)**.** Let $\mathcal{N} = \big(S_i, q_i\big)_{i=1}^{n}$ be a normal form game, $1 \leq i \leq n$, and $s_i, s_i' \in S_i$. Then s_i *strictly dominates* s_i' if for all tuples

$$(t_1, \cdots, t_{i-1}, t_{i+1}, \cdots t_n) \in \prod_{\substack{1 \leq j \leq n \\ j \neq i}} S_j$$

it holds that

$$q_i(t_1, \cdots, s_i, \cdots, t_n) > q_i(t_1, \cdots, s_i', \cdots, t_n).$$

We also say that s_i' is *strictly dominated* by s_i.

Let's assume that strictly dominated strategies are never chosen in the course of rational play, as a player would be guaranteed a better outcome by choosing the corresponding dominating strategy. Given that such strategies will never be chosen, let's consider games that result from dropping the strictly dominated strategies. By removing a strictly dominated strategy from some normal form game $\mathcal{N} = \left(S_i, q_i\right)_{i=1}^n$ we obtain a new game $\mathcal{N}^1 = \left(S_i^1, q_i^1\right)_{i=1}^n$. Iterating, we arrive at some game $\mathcal{N}^2$, and then $\mathcal{N}^3$, and so on. If we iterate until $\mathcal{N}^j = \mathcal{N}^{j+1}$ (which will always happen strategy sets are finite) then we arrive at a game in which no strategy is strictly dominated. If a set of strategy profiles contains no strictly dominated strategy profiles, we say that it is *strict dominance free*.

Definition 3.10.1.2 (Dominance free (Normal form))**.** Let $\mathcal{N} = \left(S_i, q_i\right)_{i=1}^n$ be a normal form game. $S_i' \subseteq S_i$ is *strict dominance free* if S_i contains no strictly dominated strategies. We also say that $\mathcal{N}$ is *dominance free* if each S_i is.

The iteration of the removal of strictly dominated strategies therefore results in a dominance free game. There is then a result that states that the game that results from this process is independent of the order in which the strategies were removed (see, e.g. [MSZ13]).

Proposition 3.10.1.3. *Let $\mathcal{N}$ be a normal form game and suppose that $\mathcal{H}$ and $\mathcal{K}$ are games obtained by iteratively removing strictly dominated strategies from $\mathcal{N}$. Then $\mathcal{H} = \mathcal{K}$.*

A corollary of this proposition is that *the* largest dominance free subset of a set of strategies is well-defined.

3.10.2 Subgame perfect dominance

Strong domination is well-understood in the context of normal form games. There is little in the literature about dominant strategies in games with sequential play. Just as Nash equilibria generalise to subgame perfect Nash equilibria, dominance free strategies can be generalised to subgame perfect dominance free strategies.

3.10.3 Dominance selection functions

In this section we generalise the solution concept of the iterated removal of dominated strategies to sequential games.

In a normal form game we can associate to each strategy a set of real numbers that may result from playing that strategy. In this way we can see a corresponding notion of 'strict domination' amongst subsets of the real numbers.

Definition 3.10.3.1. Let $S, T \subseteq \mathbb{R}$ and suppose that $S, T \neq \varnothing$. S *strictly dominates* T if $\min(S) > \max(T)$. We write $S \succ_s T$.

We now define the *strict dominance selection functions* to be those that return the set of choices that are not mapped to strictly dominated subsets of the reals for a given context.

Definition 3.10.3.2. Let R be $\mathcal{P}_\mathrm{f}(\mathbb{R}^n)$ where the semilattice join is given by union (equivalently, the order structure is given by inclusion). Given $k : X_i \to \mathcal{P}_\mathrm{f}(\mathbb{R}^n)$, define $k^i : X_i \to \mathcal{P}_\mathrm{f}(\mathbb{R})$ to be $\big(\mathcal{P}_\mathrm{f}(\pi_i)\big) \circ = k$. Define the i^{th} *strict dominance* selection function, $\varepsilon_i^s : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X_i)$ by

$$\varepsilon_i^s(k : X_i \to \mathcal{P}_\mathrm{f}(\mathbb{R}^n)) = \left\{ x_i \in X_i \;\middle|\; \forall x_i' \in X_i,\;\; k^i(x_i) \not\prec_s k^i(x_i') \right\}.$$

The strict dominance selection functions are witnessing and upwards closed, demonstrating that they provide an appropriate solution concept for multi-valued selection functions.

Proposition 3.10.3.3. *ε_i^s is witnessing and upwards closed.*

Proof. Let $I : X_i \to \mathcal{P}_\mathrm{f}(X_i \to \mathcal{P}_\mathrm{f}(\mathbb{R}^n))$. Suppose first that

$$x \in \varepsilon_i^s(\lambda(x' : X). \bigcup_{k \in I(x')} k(x')).$$

That is, for all $x' \in X_i$,

$$\max(\bigcup_{k \in I(x)} k^i(x)) \geq \min(\bigcup_{k \in I(x')} k^i(x'))$$

Then, setting $p_x \in I(x)$ to be a function attaining the maximum of $\bigcup_{k \in I(x)} k^i(x)$ and, for $x' \neq x$, setting $p_{x'} \in I(x')$ to be a function attaining the minimum of $\bigcup_{k \in I(x')} k^i(x')$, we define a choice function $p_- : X_i \to (X_i \to \mathcal{P}_\mathrm{f}(\mathbb{R}^n))$ such that

$$x \in \varepsilon_i^s(\lambda(x' : X).p_{x'}(x')).$$

Hence ε_i^s is witnessing.

It is similarly easy to show that ε_i^s is upwards closed as

$$\max(p_x^i(x)) \geq \min(p_{x'}^i(x')) \Longrightarrow \max(\bigcup_{p \in I(x)} p^i(x)) \geq \min(\bigcup_{p \in I(x')} p^i(x')).$$

$\square$

The two strict dominance selection functions is not, in general, witnessing as shown by the below counterexample.

Proposition 3.10.3.4. *Let* $X_1 = X_2 = \{0,1\}$ *and let* $R = \mathcal{P}_{\mathrm{f}}(\mathbb{R}^2)$*.* $(\varepsilon_1^s \otimes \varepsilon_2^s)$ *is not witnessing.*

Proof. Define contexts $p_{\varepsilon_1^s}, p_{\varepsilon_2^s}, p_0 : X_1 \times X_2 \to \mathcal{P}_{\mathrm{f}}(\mathbb{R}^2)$ by

$$p_{\varepsilon_1^s}(x,x') = \begin{cases} \{(1,-1)\} & x = x' = 0 \\ \{(0,0)\} & \text{otherwise} \end{cases}$$

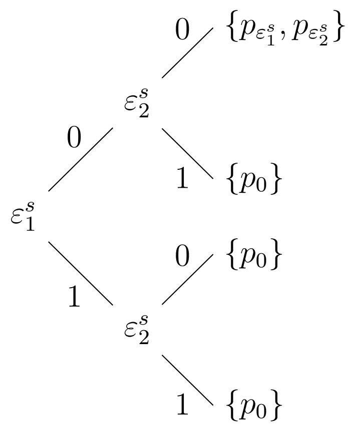
$$p_{\varepsilon_2^s}(x,x') = \begin{cases} \{(-1,1)\} & x = x' = 0 \\ \{(0,0)\} & \text{otherwise} \end{cases}$$

$$p_0(x,x') = \{(0,0)\}.$$

Define $I : X^2 \to \mathcal{P}_{\mathrm{f}}(X^2 \to \mathcal{P}_{\mathrm{f}}(\mathbb{R}^2))$ by

$$I(x,x') = \begin{cases} \{p_{\varepsilon_1^s}, p_{\varepsilon_2^s}\} & x = x' = 0 \\ \{p_0\} & \text{otherwise.} \end{cases}$$

We think of $\varepsilon_{\varepsilon_1^s}$ and $\delta_{\varepsilon_2^s}$ as playing the following game where the outcome function is chosen nondeterministically.



We will see that $(\varepsilon_1^s \otimes \varepsilon_2^s)$ fails to be witnessing as ε_1^s is satisfied with playing 0 in the case $(0,0)$ results in $p_{\varepsilon_1^s}$ and ε_2^s is satisfied playing 0 when $(0,0)$ results in $p_{\varepsilon_2^s}$, but there is no possible resulting context under which both ε_1^s and ε_2^s are happy to choose 0. Indeed, simple checks verify that

$$(0,0) \in (\varepsilon_1^s \otimes \varepsilon_2^s)\left(\lambda\big((x,y) : X_1 \times X_2\big).\bigcup_{k \in I(x,y)} k(x,y)\right)$$

but that there is no choice function $p_{:}(X^1 \times X_2) \to ((X_1 \times X_2) \to \mathcal{P}_{\mathrm{f}}(R^2))$ for I with $(0,0) \in (\varepsilon \otimes \delta)(\lambda(x,y).p_{(x,y)}(x,y))$. $\square$

Consider the game given by $\big((X_i)_{i=1}^n, (\varepsilon_i^s)_{i=1}^n, k\big)$. By 3.9.0.7, we know that $\big(\bigotimes_{i=1}^n \varepsilon_i\big)(k)$ is equal to the set of $\Sigma(\mathcal{G})$ plays. The set of strategies $\Sigma(\mathcal{G})$ is then the maximal set of strategies such that no strategy is strictly dominated in any subgame. When each X_i is finite, this simply means that $\big(\bigotimes_{i=1}^n \varepsilon_i^s\big)(k)$ computes the plays of strategies obtained via the iterated removal of strictly dominated strategies.

Chapter 4

Conclusions and further work

The selection monad and its generalised counterparts are not well-studied. This is perhaps not surprising as the first non-automated proof that selection functions form a monad appeared as recently as 2010 in [EO10a]. To the author's knowledge, the body of work making use of selection functions consists in a handful of papers on

- Synthetic topology [Esc04, Esc08, Esc07];
- Proof theory [EO14, OP15, EO15];
- Functional programming foundations [Hed14, Hed15]; and
- Game theory (these works being closely related to the work in this thesis) [EO10a, EO10b, HOS$^+$17a, HOS$^+$17b]

It is perhaps slightly concerning that work on selection functions is being conducted by so few authors, the potential reasons for this being that

1. The study of selection functions began only relatively recently and has not yet reached a wider audience;
2. The current uses of selection functions are somewhat esoteric and of less interest to the wider computer science community (unlike, say, the continuation monad);
3. The mathematics of selection functions is sometimes tedious and difficult to communicate (though this situation should improve as they become better understood).

In the rest of this chapter, we discuss some of the possible future work on sequential games and selection functions.

4.1 Dependent products

There is currently no clear game theoretic interpretation of the dependent product of selection functions. Recall that a strong monad T induces a dependent product

$$\boxtimes : (TX \times (X \to TY)) \to T(X \times Y)$$

as well as the simpler independent product

$$\otimes : (TX \times TY) \to T(X \times Y)$$

that we have been using. For the selection monad, the dependent product is an operation

$$\boxtimes : \Big(\mathcal{J}_R(X) \times \big(X \to \mathcal{J}_R(Y)\big)\Big) \to \mathcal{J}_R(X \times Y).$$

We can imagine that $\varepsilon \boxtimes \delta$ represents a sequential game where the moves δ considers optimal depends not merely on a context, but also on the move chosen by ε. Whether this brings new solutions concepts within the purview of selection functions is, as of yet, unclear.

4.2 Exotic monads

In 3.8 we showed that multi-valued selection functions fail, in general, to compute subgame perfect plays. *Prima facie*, this is because the algebras of the finite nonempty powerset monad destroy much of the fine-grained information about the structure of a sequential game. It may be the case that there exist monads that are better suited to this task and that do compute the set of subgame perfect plays.

Selection functions in arbitrary cartesian closed categories and over arbitrary monads are not well-understood. Moreover, the problem of generalising results regarding multi-valued selection functions to arbitrary T-selection functions seems very difficult.

Part II

Open games

Chapter 5

Concrete open games

Open games offer an approach to compositional game theory that is significantly more expressive than the theory of sequential games. Sequential games offer no account of *simultaneous play*, whilst open games support both sequential and parallel composition. There are instances of real-world games that include both sequential and simultaneous play and any satisfactory attempt at compositional game theory must be able to account for both types of play within the same formalism. For instance, in warfare we can imagine that opposing nations might simultaneously and privately decide where to deploy troops. Following this they must react to intelligence about where the other nation has deployed. Another example is 'hidden move Go' whereby two players each place a number of invisible (i.e. the other play does not know their location) stones on the Go board before play begins. The game then proceeds as usual and the location of the hidden stones is revealed as and when they would impact the game (for example, in the case of a capture or if a player tries to place a stone on top of an invisible stone).

In this chapter we cover concrete open games, which correspond to games of perfect information and which use **Set** as an ambient category. The key ideas in this chapter are not the author's (the best early work on open games is likely the PhD thesis [Hed16], and a more recent treatment is the paper [GKLF18]), but are presented both for completeness and because there were some minor gaps in the original works which I have since filled. In chapter 6 we will generalise concrete open games to ambient categories other than **Set** and, in doing so, we will greatly enhance the expressive power of the open games formalism.

5.1 Chapter overview

5.2 covers the basic definitions and results concerning *symmetric monoidal categories*, which form the foundation for the rest of the work in this thesis; 5.3 introduces *lenses*, a bidirectional mathematical structure which is used to describe the structure of an open game; 5.4 and 5.5 show how lenses form a symmetric monoidal category; 5.6 formally defines *concrete open games*, the structure of interest for this chapter, and shows how concrete open games form a symmetric monoidal category; 5.7 shows how the games of classical game theory can be modelled using concrete open games; 5.8 is a discussion of the possible benefits of a different definition of concrete open games.

5.2 Monoidal categories

The remainder of this thesis will be carried out in the context of *(symmetric) monoidal categories* which provide a natural setting for processes that can be combined in two dimensions (for instance, in *sequence* and in *parallel*). Monoidal categories admit of a powerful graphical calculus of *string diagrams* that we use extensively. This calculus makes readable many proofs and equations that would otherwise be obscure. In the past decade there has been an explosion of applications for monoidal category theory and its string diagrams in work on quantum mechanics [CK17, Bae06, Sel12], compositional models of natural language [BCG+17, KSPC13], chemical reaction networks [BP17], and probability theory [CS12] (to name a few). The standard reference for monoidal categories, and where a much more detailed discussion of the following definitions and results can be found, is [ML71].

Definition 5.2.0.1 (Monoidal category)**.** A *monoidal category* is a category $\mathcal{C}$ together with

- A bifunctor $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ called the *tensor product*;
- An object $I \in \mathcal{C}$ called the *monoidal unit*; and
- Natural isomorphisms $\alpha_{X,Y,Z} : (X \otimes Y) \otimes Z \to X \otimes (Y \otimes Z)$, $\lambda_X : I \otimes X \to X$, and $\rho_X : X \otimes I \to X$ called the *associator* and *left* and *right unitors* respectively.

It is moreover required that the following diagrams commute for all objects $X, Y, Z, W \in \mathcal{C}$.

$$
\begin{array}{ccccc}
((X \otimes Y) \otimes Z) \otimes W & \xrightarrow{\alpha_{X,Y,Z} \otimes \mathrm{id}_W} & (X \otimes (Y \otimes Z)) \otimes W & \xrightarrow{\alpha_{X,Y \otimes Z,W}} & X \otimes ((Y \otimes Z) \otimes W \\
\alpha_{X \otimes Y,Z,W} \downarrow & & & & \downarrow \mathrm{id}_X \otimes \alpha_{Y,Z,W} \\
(X \otimes Y) \otimes (Y \otimes Z) & & \xrightarrow[\alpha_{X,Y,Z \otimes W}]{\qquad\qquad\qquad} & & X \otimes (Y \otimes (Z \otimes W))
\end{array}
$$

$$
\begin{array}{ccc}
(X \otimes I) \otimes Y) & \xrightarrow{\alpha_{X,I,Y}} & X \otimes (I \otimes Y) \\
& \rho_X \otimes \mathrm{id}_Y \searrow & \downarrow \mathrm{id}_X \otimes \lambda_Y \\
& & X \otimes Y
\end{array}
$$

The above diagrams are sometimes referred to as the MacLane pentagon and MacLane triangle respectively.

The diagrams in the above definition are sufficient for establishing the following coherence theorem for monoidal categories.

Theorem 5.2.0.2 (Coherence for monoidal categories)**.** *Every formal diagram in a monoidal category constructed using the structural isomorphisms and monoidal unit commutes.* □

When the monoidal tensor $\otimes$ is 'sufficiently' commutative, we obtain a *symmetric monoidal category.*

Definition 5.2.0.3 (Symmetric monoidal category)**.** A *symmetric monoidal category* is a monoidal category $\mathcal{C}$ together with a *swap* natural isomorphism $s_{X,Y} : X \otimes Y \to Y \otimes X$ such that, for all $X, Y, Z \in \mathcal{C}$,

$$
\begin{array}{ccc}
X \otimes I & \xrightarrow{s_{X,I}} & I \otimes X \\
& \rho_X \searrow & \downarrow \lambda_X \\
& & A
\end{array}
$$

and

$$
\begin{array}{ccccc}
(X \otimes Y) \otimes Z & \xrightarrow{s_{X,Y} \otimes \mathrm{id}_Z} & (Y \otimes X) \otimes Z & \xrightarrow{\alpha_{Y,X,Z}} & Y \otimes (X \otimes Z) \\
\alpha_{X,Y,Z} \downarrow & & & & \downarrow \mathrm{id}_Y \otimes s_{X,Z} \\
X \otimes (Y \otimes Z) & \xrightarrow[s_{X,Y \otimes Z}]{} & (Y \otimes Z) \otimes X & \xrightarrow[\alpha_{Y,Z,X}]{} & Y \otimes (Z \otimes X)
\end{array}
$$

commute, and $s_{Y,X} \circ s_{X,Y} = \mathrm{id}_{X \otimes Y}$.

Example 5.2.0.4. **Set** is symmetric monoidal. The monoidal unit is the one element set $\{\star\}$; the monoidal tensor is the cartesian product; and the structural natural isomorphisms are the obvious bijections.

Example 5.2.0.5. The category $\mathbf{FVect}_{\mathbb{R}}$ of finite dimensional real vector spaces is symmetric monoidal. The monoidal unit is the one-dimensional vector space and if V and U are vector spaces with bases $\mathcal{B}_V = \{v_1, \cdots, v_n\}$ and $\mathcal{B}_U = \{u_1, \cdots, u_m\}$ respectively, then $V \otimes U$ is the vector space with basis $\mathcal{B}_V \times \mathcal{B}_U$.

A *monoidal functor* is a functor between monoidal categories that respects monoidal structure.

Definition 5.2.0.6. A *monoidal functor* $F : \mathcal{C} \to \mathcal{D}$ between monoidal categories $\mathcal{C}$ and $\mathcal{D}$ is a functor together with a natural transformation $\varphi_{A,B} : FA \otimes FB \to F(A \otimes B)$ and a morphism $\psi : I_{\mathcal{C}} \to FI_{\mathcal{C}}$ such that the following diagrams commute for all objects A, B, and C of $\mathcal{C}$.

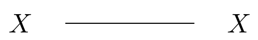
$$\begin{array}{ccccc}
(FA \otimes FB) \otimes FC & \xrightarrow{\alpha} & FA \otimes (FB \otimes FC) & \xrightarrow{\mathrm{id} \otimes \varphi_{B,C}} & FA \otimes F(B \otimes C) \\
\downarrow{\scriptstyle \varphi_{A,B} \otimes \mathrm{id}} & & & & \downarrow{\scriptstyle \varphi_{A,B\otimes C}} \\
F(A \otimes B) \otimes FC & \xrightarrow[\varphi_{A\otimes B,C}]{} & F((A \otimes B) \otimes C) & \xrightarrow[F\alpha]{} & F(A \otimes (B \otimes C))
\end{array}$$

$$\begin{array}{ccc}
FA \otimes I_{\mathcal{D}} & \xrightarrow{\mathrm{id} \otimes \psi} & FA \otimes FI_{\mathcal{C}} \\
\downarrow{\scriptstyle \rho} & & \downarrow{\scriptstyle \varphi_{A,I_{\mathcal{C}}}} \\
FA & \xleftarrow[F\rho]{} & F(A \otimes I_{\mathcal{C}})
\end{array}
\qquad
\begin{array}{ccc}
I_{\mathcal{D}} \otimes FB & \xrightarrow{\psi \otimes \mathrm{id}} & FI_{\mathcal{C}} \otimes FB \\
\downarrow{\scriptstyle \lambda} & & \downarrow{\scriptstyle \varphi_{I_{\mathcal{C}},B}} \\
FB & \xleftarrow[F\lambda]{} & F(I_{\mathcal{C}} \otimes B)
\end{array}$$

5.2.1 Diagrams for symmetric monoidal categories

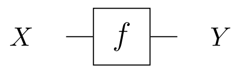
A thorough review of the many diagrammatic calculi associated with various types of monoidal categories is [Sel10]. There exist many extensions of the 'vanilla' calculus for monoidal categories that are able to visually represent additional structure that can exist in a monoidal category. The calculus we make use of is relatively modest, the only addition being a representation of the 'swap' morphisms that exist in symmetric monoidal categories. The string diagrams in this thesis should be read 'left to right,' but we note that there is little consensus on the appropriate orientation of string diagrams in the literature[1].

- An object X of $\mathcal{C}$ is denoted by a wire



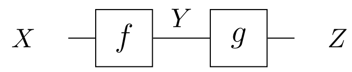
,

- A morphism $f : X \to Y$ is denoted by a box



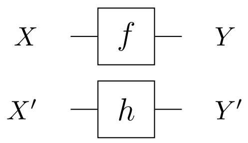
,

- Given another morphism $g : Y \to Z$, the sequential composite $g \circ f : X \to Z$ is



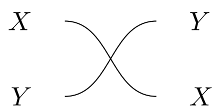
,

- Given $h : X' \to Y'$ then tensor $f \otimes h : X \otimes X' \to Y \otimes Y'$ is



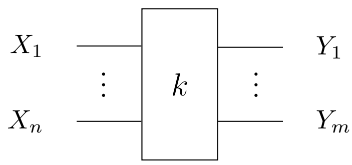
,

- The swap maps $s_{X,Y} : X \otimes Y \to Y \otimes X$ are



,

- In general, morphisms $k : X_1 \otimes \cdots \otimes X_n \to Y_1 \otimes \cdots \otimes Y_m$ are denoted by boxes with multiple inputs/outputs



,

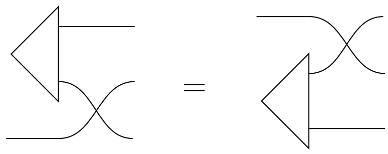
- morphisms $s : I \to X_1 \otimes \cdots \otimes X_n$ and $e : Y_1 \otimes \cdots \otimes Y_m \to I$ are called *states* and *effects*, and are respectively denoted by

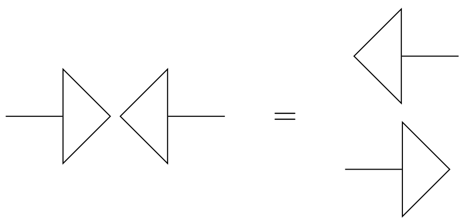
[1] Left-right is most common, but bottom-top is popular among quantum theorists. Top-bottom and right-left exist, but are rarely used.

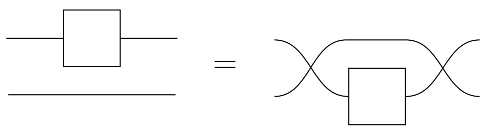
The remarkable theorem underlying the diagrammatic calculus is as follows.

Theorem 5.2.1.1 (Correctness of string diagrams for symmetric monoidal categories)**.** *A well-typed equation of morphisms in a formal symmetric monoidal category follows from the symmetric monoidal category axioms if and only if it holds in the graphical language up to four-dimensional isotopy.*

Intuitively, morphisms in a symmetric monoidal category are invariant under topological deformations of string diagrams which keep the inputs and outputs fixed. That these deformations are 'four dimensional' simply means that deformations where the diagram passes through itself are permitted. For instance, the following equalities hold in any symmetric monoidal category.





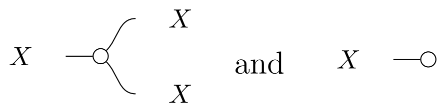


Reasoning using string diagrams is carried out by performing topological deformation and by using known equalities of the category in consideration to perform substitutions in a string diagram.

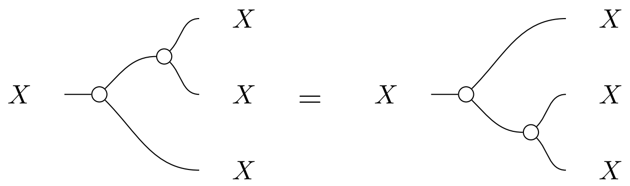
5.2.2 Comonoids

A *comonoid* in a symmetric monoidal category $\mathcal{C}$ is a 'copying-like' operation. A comonoid on an object $X : \mathcal{C}$ is given by a two maps $c : X \to X \otimes X$ and $e : X \to I$ where c is the 'copying' map and e is a 'deleting' map. These maps must be like copying in two ways. Firstly, it should be co-associative and, secondly, copying and then deleting one of the copies should be the same as doing nothing at all.

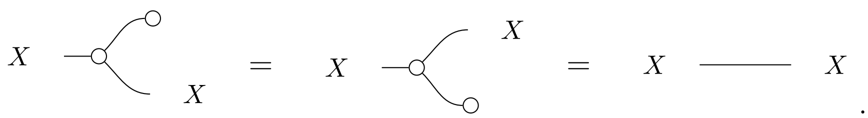
Definition 5.2.2.1 (Comonoid)**.** A *comonoid* in a monoidal category $\mathcal{C}$ on an object $X : \mathcal{C}$ is a pair of maps $c : X \to X \otimes X$, $e : X \to I$, denoted by



respectively, such that

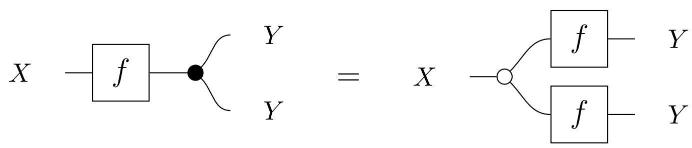


and

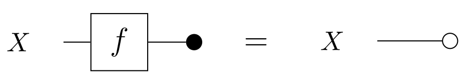


Example 5.2.2.2 (The copy/delete comonoid)**.** Let X be a set. The diagonal map $\Delta : X \to X \times X$ and deleting map $! : X \to \{\star\}$ form a comonoid in **Set**.

Definition 5.2.2.3 (Comonoid homomorphism)**.** Suppose we have comonoids on objects X and Y in a monoidal category denoted by white and black circles respectively. A *comonoid homomorphism* is a morphism $f : X \to Y$ such that



and



Example 5.2.2.4. Any function is a comonoid homomorphism for the copy/delete comonoid in **Set**.

5.3 Lenses

The lenses used in this thesis are direct descendants of the lenses of database theory (see the paper [BS81] for an embryonic account of lenses for databases). We use lenses to describe the flow of information through a game (the connection to databases will be explained shortly). A lens for a given game describes which players have access to what information when making a strategic decision, and also how information about players' strategic decisions is ultimately fed into the outcome function for the game. For example, it may specify an order of play, or whether two players are playing in parallel, or even whether some players are privy to certain information in the environment that other players are not.

In general, lenses can be thought of as processes that perform some computation and then propagate some resulting feedback from the environment backwards through a system of which they are a part. In particular, this means that lenses have both covariant and contravariant components. The covariant component carries out the initial computation and the contravariant component propagates the resulting feedback back through the system. Crucially, lenses are also *compositional* in the sense that they admit both sequential and parallel composition and, consequently, form a symmetric monoidal category.

Given a database x of type X we may want to view some subdatabase y of type Y. This is encapsulated by a *view function* $v : X \to Y$. From this 'close-up' view of the database we may want to edit the database by updating y. Given an update of the view y we then need to know how this update propagates to an update of the original superdatabase x. That is, given an original database x and an updated view $y' : Y$, we should specify some updated $x' : X$ given by some *update function* $u : X \times Y \to X$. The pair (v, u) is a *lens* with type $X \to Y$. The connection to our previous abstract definition of lenses is as follows:

- The covariant computation associated with the lens is the view function $v : X \to Y$,
- the resulting feedback from the environment is the update made to the subdatabase returned by the view function, and
- this feedback is propagated back to the whole database via the update function.

Abstracting away from databases, there is no reason to demand that the feedback generated by the environment will have the same type as the output of the lens computation. Similarly, we may be interested in cases where the update function is not-so-literally an 'update' function, but merely a function that propagates *some kind* of feedback back through the system. As such, the lenses we will be using will have types of the form $(X, S) \to (Y, R)$ where the covariant component of the lens is of type $X \to Y$ and the contravariant component is of type $X \times R \to S$.

In game theory, we can regard players as 'lenses that care about the feedback they receive from the environment.' In a game with sequential play, players make some play (computation), receive some utility (feedback) from the outcome function, and then pass some feedback to earlier players in the game (their outcome function given the moves that the later players chose). Moreover, given that lenses admit of parallel composition as well as sequential composition, we obtain a nuanced notion of information flow in a game.

In the next section we describe a symmetric monoidal category of concrete lenses. 'Concrete' here refers to the fact that the view and update functions are functions in **Set**. In chapter 6 we give a further generalisation of lenses, noting some of the surprising obstacles that arise.

5.4 The category of concrete lenses

Definition 5.4.0.1 (Concrete lens)**.** Let X, S, Y and R be sets. A *concrete lens* $l : (X, S) \to (Y, R)$ is a pair of functions $(l_v : X \to Y,\ l_u : X \times R \to S)$.

We offer game theoretic interpretations of the sets X, S, Y, and R in 5.6.

As a trivial first example, there is an obvious mapping that takes a morphism of $\mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ and returns a concrete lens.

Example 5.4.0.2. Let $f : X \to Y$ and $g : R \to S$. Define a concrete lens $\langle f, g\rangle : (X, S) \to (Y, R)$ by

$$\langle f, g\rangle_v = f$$
$$\langle f, g\rangle_u(x, r) = g(r)$$

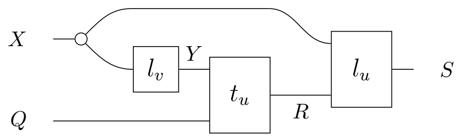
Definition 5.4.0.3 (Sequential composition of concrete lenses)**.** Let $l : (X, S) \to (Y, R)$ and $t : (Y, R) \to (Z, Q)$ be concrete lenses. The *sequential composite* $t \circ l : (X, S) \to (Z, Q)$ is given by $\big((t \circ l)_v : X \to Z, (t \circ l)_u : X \times Q \to S\big)$ where

$$(t \circ l)_v = t_v \circ l_v$$

and $(t \circ l)_u$ is given by

$$X \times Q \xrightarrow{\Delta_X \times \mathrm{id}_X} X \times X \times Q \xrightarrow{\mathrm{id}_X \times l_v \times \mathrm{id}_Q} X \times Y \times Q \xrightarrow{\mathrm{id}_X \times t_u} X \times R \xrightarrow{l_u} S.$$

As a string diagram $(t \circ l)_u$ is given by

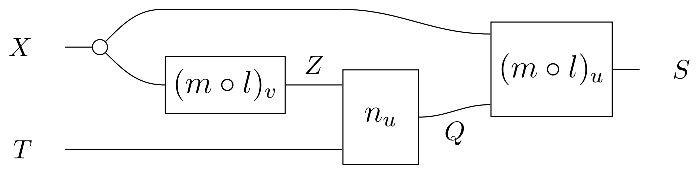


Lemma 5.4.0.4 (Sequential composition of concrete lenses is associative)**.** *Suppose we have concrete lenses*

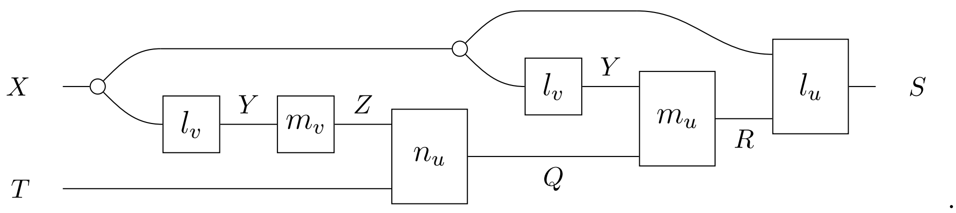
$$(X, S) \xrightarrow{l} (Y, R) \xrightarrow{m} (Z, Q) \xrightarrow{n} (W, T).$$

Then $n \circ (m \circ l) = (n \circ m) \circ l$*.*

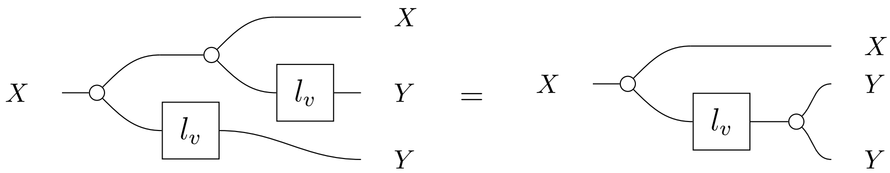
Proof. $\big(n \circ (m \circ l)\big)_v = \big((n \circ m) \circ l\big)_v$ by associativity of function composition. $\big(n \circ (m \circ l)\big)_u$ is given by



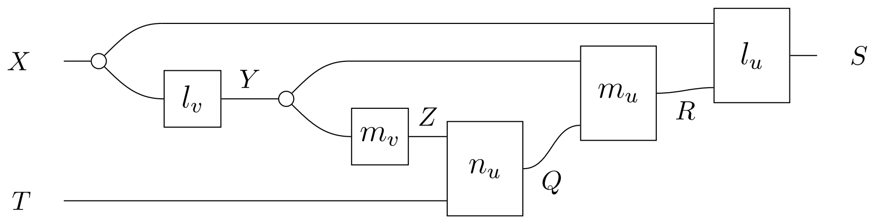
which, expanding, is



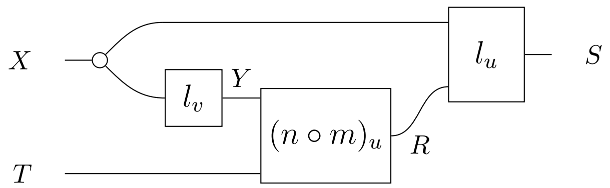
Using the comonoid equations for the copy/delete comonoid (5.2.2.1) and the fact that every function is a comonoid homomorphism (5.2.2.3) for the copy/delete comonoid, we see that



Substituting, we see that $\big(n \circ (m \circ l)\big)_u$ is



This, then, is just



which is $\big((n \circ m) \circ l\big)_u$ as required. ☐

Theorem 5.4.0.5 (Concrete lenses form a category)**.** *There is a category* **CL** *with object class* **Set** $\times$ **Set** *and concrete lenses as morphisms.*

Proof. It just remains to show that **CL** has identity morphisms $\mathrm{id}_{(X,S)} : (X, S) \to (X, S)$. Define $\mathrm{id}_{X_v}(x) = x$ and $\mathrm{id}_{X_u}(x, s) = s$. The result then follows by easy checks. ☐

5.5 The monoidal structure of concrete lenses

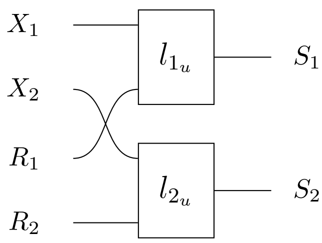
Definition 5.5.0.1 (Tensor composition of concrete lenses)**.** Let $l_1 : (X_1, S_1) \to (Y_1, R_1)$ and $l_2 : (X_2, S_2) \to (Y_2, R_2)$ be concrete lenses. The *tensor composition* $l_1 \otimes l_2 : (X_1 \times X_2, S_1 \times S_2) \to (Y_1 \times Y_2, R_1 \times R_2)$ is given by $\big((l_1 \otimes l_2)_v, (l_1 \otimes l_2)_u\big)$ where

$$(l_1 \otimes l_2)_v = l_{1_v} \times l_{2_v}$$

and $(l_1 \otimes l_2)_u$ is given by

$$X_1 \times X_2 \times R_1 \times R_2 \xrightarrow{\;\cong\;} X_1 \times R_1 \times X_2 \times R_2 \xrightarrow{\;l_{1_u} \times l_{2_u}\;} S_1 \times S_2$$

In a diagram, $(l_1 \otimes l_2)_u$ is



Lemma 5.5.0.2. $\otimes$ *is a functor.*

Proof. Suppose we have concrete lenses

$$(X_1, S_1) \xrightarrow{l^1} (Y_1, R_1) \xrightarrow{m^1} (Z_1, Q_1)$$

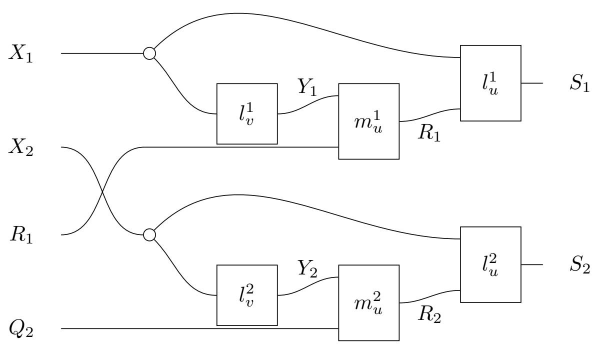
and

$$(X_2, S_2) \xrightarrow{l^2} (Y_2, R_2) \xrightarrow{m^2} (Z_2, Q_2) \quad .$$

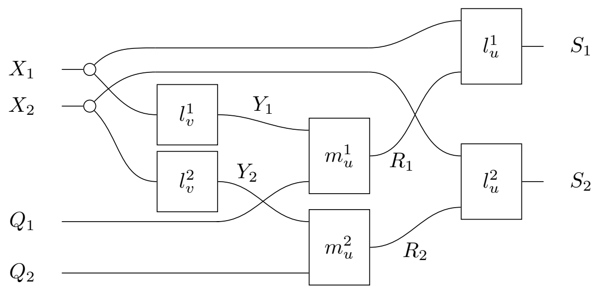
We note that

$$(m^1 \circ l^1) \otimes (m^2 \circ l^2) = (m^1 \otimes m^2) \circ (l^1 \otimes l^2).$$

is easy to verify for the view function and, concerning the update function, the corresponding string diagrams



and



are equal.

That $\mathrm{id}_{(X,S)} \otimes \mathrm{id}_{(X',S')} = \mathrm{id}_{(X \times X', S \times S')}$ follows easily from simple checks. $\square$

Theorem 5.5.0.3. *There is a symmetric monoidal category* **CL** *where the objects are pairs of sets and the morphisms are concrete lenses. Sequential composition and the monoidal tensor are as in the above definitions.*

Proof. The structural isomorphisms are inherited from **Set** as follows.

$$\begin{aligned}
\alpha_{(X_1,S_1),(X_2,S_2),(X_3,S_3)} &= \langle \alpha_{X_1,X_2,X_3}, \alpha^{-1}_{S_1,S_2,S_3} \rangle \\
\lambda_{(X,S)} &= \langle \lambda_X, \lambda_S^{-1} \rangle \\
\rho_{(X,S)} &= \langle \rho_X, \rho_S^{-1} \rangle \\
s_{(X,S),(X',S')} &= \langle s_{X,X'}, s^{-1}_{S,S'} \rangle
\end{aligned}$$

Where the α, ρ, λ, and s on the right-hand side are the obvious **Set** isomorphisms.

These maps satisfy the axioms for symmetric monoidal categories as α, ρ, λ, and s satisfy the axioms for **Set**. Naturality of these maps follows by easy checking. □

The following observations about states and effects in **CL** will be useful in the remainder of this chapter.

Lemma 5.5.0.4. $\mathbf{CL}\big(I, (X, S)\big) \cong X$.

Proof. This is easily seen as a state $l \in \mathbf{CL}\big((I, (X, S)\big)$ is given by a pair

$$\big(s : \{\star\} \to X, e : \{\star\} \times S \to \{\star\}\big).$$

□

Lemma 5.5.0.5. $\mathbf{CL}\big((Y, R), I\big) \cong (Y \to R)$

Proof. An effect $l \in \mathbf{CL}\big((Y, R), I\big)$ is given by a pair

$$\big(v : Y \to \{\star\}, u : Y \times \{\star\} \to R\big).$$

□

5.6 Concrete open games

In this section we give the definition of the central notion for this chapter, the *concrete open game*. A concrete open game consists in a set of strategy profiles; a family of concrete lenses indexed by the set of strategy profiles; and a best response function.

Definition 5.6.0.1 (Concrete open game)**.** Let X, S, Y, and R be sets. A *concrete open game* $\mathcal{G} : (X, S) \to (Y, R)$ is given by

1. A set of *strategy profiles* Σ;

2. A *play function* $\mathrm{P} : \Sigma \to \mathbf{CL}\big((X,S),(Y,R)\big)$; and

3. A *best response function* $\mathrm{B} : X \times (Y \to R) \to \mathrm{Rel}(\Sigma)$.

Recall that $\mathrm{Rel}(\Sigma)$ is the set of relations on Σ, given by $\Sigma \to \mathcal{P}(\Sigma)$. The type X is the type of *observations* made by the game; the type Y is the type of *actions* that can be chosen; the type R is the type of *outcomes*; and the type S is the type of *co-outcomes*. Of the four types associated with a concrete open game, the type S is the most mysterious. Succinctly, its purpose is to relay information about outcomes to games acting earlier. In a sequential composite $\mathcal{H} \circ \mathcal{G}$ of open games (we will define sequential composition of concrete open games shortly), the co-outcome type of $\mathcal{H}$ is also the outcome type $\mathcal{G}$. We think of $\mathcal{H}$ as receiving some outcome which is then acted upon by the contravariant component of a concrete lens given by $\mathcal{H}$'s play function before being passed back to $\mathcal{G}$ as $\mathcal{G}$'s outcome.

The best response function of an open game is an abstraction from the selection functions of the previous part and from the utility functions of classical game theory. Recall that a Nash equilibrium for a normal form game is a strategy profile in which no player has incentive to unilaterally deviate. We can instead think of a relation on the set of strategy profiles for a normal form game where strategy profiles σ and τ are related if τ is the result of players unilaterally deviating from σ to their most profitable unilateral deviation. Nash equilibria are then the fixed points of this relation. In the definition of a concrete open game, we work directly with a best response relation rather than preference relations.

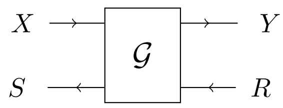
The play function takes a strategy as argument and returns a concrete lens that describes an *open play* of the game $\mathcal{G}$ ('open' here means 'lacking a particular observation and outcome function' and is explained in the next paragraph). To justify this interpretation, recall that a concrete lens $l : (X,S) \to (Y,R)$ consists in $v : X \to Y$ and $u : X \times R \to S$. The view function v describes how a game decides on an action given an observation (similar to how strategies for sequential games work). The update function u describes precisely how games relay information about outcomes to other games acting earlier.

As the name suggests, concrete open games are *open* to their environment. The appropriate notion of a *context* for a concrete open game is given in the following definition. A concrete open game together with a context can be thought of as a full description of a game.

Definition 5.6.0.2. Let $\mathcal{G} : (X, S) \to (Y, R)$. A *history* for $\mathcal{G}$ is an element x of X, an *outcome function* for $\mathcal{G}$ is a function $k : Y \to R$, and a *context* for $\mathcal{G}$ is a pair $(x, k) : X \times (Y \to R)$.

We are now in a position to justify the type of the best response function. The best response function takes a context as argument, and a context is precisely the information required for resolving the 'openness' of a concrete open game. Given a context, the best response function then returns the set of best deviations from a strategy profile σ.

We represent a concrete open game $\mathcal{G} : (X, S) \to (Y, R)$ using the following diagram.



This diagrammatic notation emphasizes the point that information flows both covariantly through $\mathcal{G}$ from observations to actions, and contravariantly through $\mathcal{G}$ from outcomes to co-outcomes. These diagrams constitute a *bona fide* diagrammatic calculus for the category of concrete open games defined in the remainder of this chapter, as detailed in [Hed17].

Notation 5.6.0.3. String diagrams in the category of open games will always be drawn with arrowheads on wires, whilst string diagrams in the ambient category will always be drawn without arrowheads.

Atomic concrete open games are an important class of concrete open games, and are the basic components out of which more complex games are constructed. Whilst concrete open games can, in general, represent aggregates of agents responding to each other (in a way that will be made precise in 5.6.3 and 5.6.4), atomic concrete open games describe games in which there is no strategic interaction. Examples are simple computations in which no decisions are made whatsoever and single agents sensitive only to a given context.

Definition 5.6.0.4 (Atomic concrete open game)**.** A concrete open game $a : (X, S) \to (Y, R)$ is *atomic* if

1. $\Sigma_a \subseteq \mathbf{CL}\big((X, S), (Y, R)\big)$;

2. For all $l \in \Sigma_a$, $\mathrm{P}_a(l) = l$; and

3. For all contexts $c : X \times (Y \to R)$, $\mathrm{B}_a(c)$ is constant.

We sometimes refer to an atomic concrete open game simply as an *atom*.

Note that an atom $a : (X, S) \to (Y, R)$ is fully determined by a subset $\Sigma_a \subseteq \mathbf{CL}\big((X, S), (Y, R)\big)$ and a *preference function* $\varepsilon_a : X \times (Y \to R) \to \mathcal{P}(\Sigma_a)$. The play function is given by the identity on Σ_a, and the best response function $\mathrm{B}_a : X \times (Y \to R) \to \mathrm{Rel}(\Sigma_a)$ is given by $\mathrm{B}_a(x, k)(\sigma) = \varepsilon_a(x, k)$.

Given $f : X \to Y$ and $g : R \to S$, as in **CL**, the pair $(f, g) \in \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}}$ can be represented as a concrete open game. We refer to such games as *computations* as no strategic choice is being made.

Example 5.6.0.5 (Computation)**.** Let $f : X \to Y$ and $g : R \to S$. The atom $\langle f, g\rangle : (X, S) \to (Y, R)$ is given by

1. $\Sigma = \{\langle f, g\rangle\}$; and

2. For all $c : X \times (Y \to R)$, $\varepsilon(c) = \Sigma$.

Similar to **CL**, the following computations will turn out to be the underlying structural maps for the symmetric monoidal category of concrete open games.

Definition 5.6.0.6 (Structural computations)**.** Define identity, associator, swaps, and left/right unitor computations to be the atomic concrete open games given by

$$\begin{aligned}
\mathrm{id}_{(X,S)} &= \langle \mathrm{id}_X, \mathrm{id}_S\rangle \\
\alpha_{X\otimes(Y\otimes Z), A\otimes(B\otimes C)} &= \langle \alpha_{X,Y,Z}, \alpha^{-1}_{X,Y,Z}\rangle \\
s_{(X,A),(Y,B)} &= \langle s_{(X,Y)}, s^{-1}_{(A,B)}\rangle \\
\rho_{(X,Y)} &= \langle \rho_X, \rho^{-1}_Y\rangle \\
\lambda_{(X,Y)} &= \langle \lambda_X, \lambda^{-1}_Y\rangle
\end{aligned}$$

where the **Set** functions on the right-hand side of the equalities are the obvious **Set** isomorphisms.

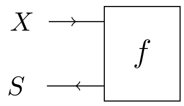
Counit games are an interesting class of atoms that reverse the direction of information flow in a concrete open game.

Definition 5.6.0.7 (Counit)**.** Let $f : X \to S$. Define an atomic concrete open game $c_f : (X, S) \to (\{\star\}, \{\star\})$ by

1. $\Sigma_{c_f} = \{\langle !, f\rangle\}$; and

2. For all $c : X \times (\{\star\} \to \{\star\})$, $\varepsilon(c) = \Sigma_{c_f}$.

We are being slightly relaxed with notation here as the update function for c_f has type $X \times \{\star\} \to S$ while f has type $X \to S$. We represent c_f as follows.

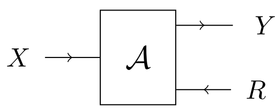


5.6.1 Agents

So far we have only seen open games for which the set of strategies is a singleton, describing games with no strategic decisions. Our first examples of a concrete open game with non-trivial strategy set are *agents*. These can be used to represent the utility-maximising agents of traditional game theory or, more generally, to represent players trying to influence the outcome of a game.

Definition 5.6.1.1 (Agent)**.** An *agent* $\mathcal{A} : (X, \{\star\}) \to (Y, R)$ is an atom such that $\Sigma = \mathbf{CL}\big((X, \{\star\}), (Y, R)\big)$.

Recall that a concrete lens $l : \mathbf{CL}\big((X, \{\star\}), (Y, R)\big)$ is a pair $(v : X \to Y, u : X \times R \to \{\star\})$ and, hence, is uniquely determined by a function of type $X \to Y$. Consequently, a *strategy* for an agent specifies how an agent map chooses an action of type Y given an observation of type X. Given a context $c : X \times (Y \to R)$, $\mathrm{B}_{\mathcal{A}}(c)$ picks out the set of strategies $\mathcal{A}$ considers acceptable in the context c.. Agents are represented diagrammatically by



We can specialise the definition above to model the utility-maximising agents of traditional game theory.

Example 5.6.1.2 (Utility-maximising agent)**.** The *utility-maximising agent* $\mathcal{A} : (X, \{\star\}) \to (Y, \mathbb{R})$ is given by

$$\varepsilon(x, k) = \big\{\sigma : X \to Y \bigm| \sigma(x) \in \arg\max(k)\big\}.$$

5.6.2 Best response with concrete lenses

Recall from 5.5.0.4 and 5.5.0.5 that $\mathbf{CL}\big(I,(X,S)\big) \cong X$ and that $\mathbf{CL}\big((Y,R),I\big) \cong Y \to R$. Using these facts we can rephrase the type of best response for a concrete open game $\mathcal{G} : (X,S) \to (Y,R)$ as

$$\mathrm{B}_{\mathcal{G}} : \mathbf{CL}\big(I,(X,S)\big) \times \mathbf{CL}\big((Y,R),I\big) \to \mathrm{Rel}(\Sigma_{\mathcal{G}}).$$

This formulation allows for a concise and natural definition of *sequential composition* for concrete open games where it would otherwise seem *ad hoc*. To make matters clear, we write x when talking about elements of X and $x^\star$ when talking about corresponding states in $\mathbf{CL}\big(I,(X,S)\big)$. Similarly, we write $k : Y \to R$ when talking about functions in **Set** and we write $k_\star$ when talking about corresponding effects in $\mathbf{CL}\big((Y,R),I\big)$.

5.6.3 Sequential composition of concrete open games

In this section we specify how to define the *sequential composite* $\mathcal{H} \circ \mathcal{G} : (X,S) \to (Z,Q)$ of two concrete open games $\mathcal{G} : (X,S) \to (Y,R)$ and $\mathcal{H} : (Y,R) \to (Z,Q)$.

We imagine that this composition *really is* sequential in a straightforward way. $\mathcal{G}$ is 'played out' according to some strategy $\sigma \in \Sigma_{\mathcal{G}}$ and then $\mathcal{H}$ is 'played out' according to some $\tau \in \Sigma_{\mathcal{H}}$. A choice of $(\sigma,\tau) \in \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$ therefore determines an open play of $\mathcal{G}$ and $\mathcal{H}$ played in sequence, and so we take $\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$ to be the strategy profile set of $\mathcal{H} \circ \mathcal{G}$.

The play function of the sequential composite is defined straightforwardly using the sequential composition of concrete lenses defined in 5.4.0.3.

Defining best response for a sequential composite is a bit more delicate and, for explanatory purposes, we make use of the informal notion of a *local context for a subgame*. Given a context $c = (x : X, k : Z \to Q)$ and a strategy (σ,τ) for $\mathcal{H} \circ \mathcal{G}$, the best response relation of $\mathcal{H} \circ \mathcal{G}$ is specified by calling the best response function of $\mathcal{G}$ with a modified context corresponding to how c 'appears' to $\mathcal{G}$ when $\mathcal{H}$ plays according to τ and, similarly, calling the best response function of $\mathcal{H}$ with a modified context corresponding to how c 'appears' to $\mathcal{H}$ when $\mathcal{G}$ plays according to σ. In practice we define these 'local contexts' in the obvious way that type checks, but this is because the work has already been done in carefully choosing the correct definitions.

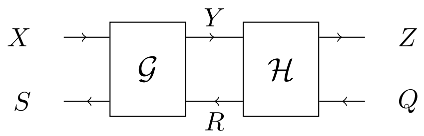
Definition 5.6.3.1 (Sequential composition for concrete open games)**.** Let $\mathcal{G} = (\Sigma_{\mathcal{G}}, \mathrm{P}_{\mathcal{G}}, \mathrm{B}_{\mathcal{G}}) : (X,S) \to (Y,R)$ and $\mathcal{H} = (\Sigma_{\mathcal{H}}, \mathrm{P}_{\mathcal{H}}, \mathrm{B}_{\mathcal{H}}) : (Y,R) \to (Z,Q)$ be concrete open games. Define

1. $\Sigma_{\mathcal{H}\circ\mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$,

2. $\mathrm{P}_{\mathcal{H}\circ\mathcal{G}}(\sigma,\tau) = \mathrm{P}_{\mathcal{H}}(\tau) \circ \mathrm{P}_{\mathcal{G}}(\sigma)$ (where $\circ$ composition is in **CL**), and

3. $\mathrm{B}_{\mathcal{H}\circ\mathcal{G}}(x^\star, k_\star)(\sigma,\tau) = \mathrm{B}_{\mathcal{G}}(x^\star, k_\star \circ \mathrm{P}_{\mathcal{H}}(\tau))\sigma \times \mathrm{B}_{\mathcal{H}}(\mathrm{P}_{\mathcal{G}}(\sigma) \circ x^\star, k_\star)(\tau)$.

We represent $\mathcal{H} \circ \mathcal{G}$ with the diagram



.

5.6.4 Tensor composition for concrete open games

The tensor composition of open games represents simultaneous play. Given concrete open games $\mathcal{G} : (X_1, S_1) \to (Y_1, R_1)$ and $\mathcal{H} : (X_2, S_2) \to (Y_2, R_2)$, the strategy set for $\mathcal{G} \otimes \mathcal{H}$ is $\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$; we make use of the tensor composition in **CL** in defining the play function; and the best response function is given by modifying the context c to give local contexts for $\mathcal{G}$ and $\mathcal{H}$.

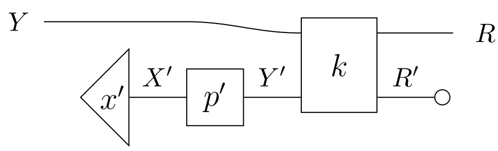
Definition 5.6.4.1 (Local contexts for tensor composition)**.** Define the *left local tensor context operator*

$$\mathcal{L} : \Big(X' \times (X' \to Y') \times (Y \times Y' \to R \times R')\Big) \to (Y \to R)$$

by

$$\mathcal{L}(x', p', k)(y) = \pi_1 \circ k(y, p'(x')).$$

As a diagram, $\mathcal{L}(x', p', l)$ is the function



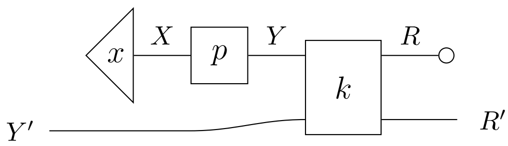
Similarly, define the *right local tensor context operator*

$$\mathcal{R} : \Big(X \times (X \to Y) \times (Y \times Y' \to R \times R')\Big) \to (Y' \to R')$$

by

$$\mathcal{R}(x, p, k)(y') = \pi_2 \circ k(p(x), y).$$

As a diagram,



Suppose we have concrete open games $\mathcal{G} : (X, S) \to (Y, R)$ and $\mathcal{H} : (X', S') \to (Y', R')$ and we wish to combine them to create some game $\mathcal{G} \otimes \mathcal{H} : (X \times X', S \times S') \to (Y \times Y', R \times R')$. Consider the left context operator $\mathcal{L}$ acting on some triple (x', p', k). If k is an outcome function for the game $\mathcal{G} \otimes \mathcal{H}$ and $\mathcal{H}$ observes x' and plays according to the function p', then $\mathcal{L}(x', p', k)$ is the 'apparent' outcome function for $\mathcal{G}$. Similarly, $\mathcal{R}(x, p, k)$ is the 'apparent' outcome function for $\mathcal{H}$ when $\mathcal{G}$ observes x and plays according to p. With this in mind, we define tensor composition for concrete open games as follows.

Definition 5.6.4.2 (Tensor composition of concrete open games)**.** Let $\mathcal{G} : (X, S) \to (Y, R)$ and $\mathcal{H} : (X', S') \to (Y', R')$ be concrete open games. Define

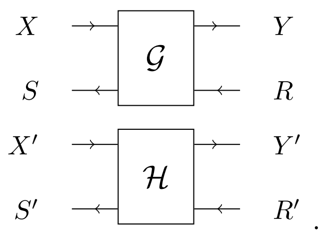
$$\mathcal{G} \otimes \mathcal{H} : (X \times X', S \times S') \to (Y \times Y', R \times R')$$

by

1. $\Sigma_{\mathcal{G}\otimes\mathcal{H}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$,
2. $\mathrm{P}_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau) = \mathrm{P}_{\mathcal{G}}(\sigma) \otimes \mathrm{P}_{\mathcal{H}}(\tau)$ (in **CL**),
3. $\mathrm{B}_{\mathcal{G}\otimes\mathcal{H}} : \Big((X \times X') \times (Y \times Y' \to R \times R')\Big) \to \mathrm{Rel}(\Sigma_{\mathcal{G}\otimes\mathcal{H}})$ is given by

$$\begin{aligned}\mathrm{B}_{\mathcal{G}\otimes\mathcal{H}}\big((x, x')^\star, k_\star\big)(\sigma, \tau) = \mathrm{B}_{\mathcal{G}}&\big(x^\star, \mathcal{L}(x', (\mathrm{P}_{\mathcal{H}}(\tau))_v, k)_\star\big)(\sigma) \\ \times \mathrm{B}_{\mathcal{H}}&\big(x'^\star, \mathcal{R}(x, (\mathrm{P}_{\mathcal{G}}(\sigma))_v, k)_\star\big)(\tau)\end{aligned}$$

$\mathcal{G} \otimes \mathcal{H}$ is represented by the diagram



5.6.5 Equivalence of open games

One subtlety remains before we can define the category of concrete open games. We aim to define a category with object class **Set** × **Set** and morphisms given by concrete open games. If carried out naïvely, this runs into the problem that strategy sets which should be identical are merely isomorphic. For instance, the strategy set of $(\mathcal{G} \circ \mathcal{H}) \circ \mathcal{K}$ is $(\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times \Sigma_{\mathcal{K}}$ whilst the strategy set of $\mathcal{G} \circ (\mathcal{H} \circ \mathcal{K})$ is $\Sigma_{\mathcal{G}} \times (\Sigma_H \times \Sigma_{\mathcal{K}})$. In order for concrete open games to form a category, we must first take an appropriate quotient. We do this by defining a notion of *simulation* between concrete open games of the same type.

Definition 5.6.5.1. A relation $\alpha : A \twoheadrightarrow B$ is *serial*[2] if for all $a \in A$ there exists some $b \in B$ such that $\alpha(a, b)$.

Definition 5.6.5.2. Let $\mathcal{G}, \mathcal{H} : (X, S) \to (Y, R)$ be concrete open games. A *simulation* $\alpha : \mathcal{G} \to \mathcal{H}$ is given by a serial relation $\alpha : \Sigma_{\mathcal{G}} \twoheadrightarrow \Sigma_{\mathcal{H}}$ such that, for all $\sigma, \sigma' \in \Sigma_{\mathcal{G}}, \tau \in \Sigma_{\mathcal{H}}$, and $c \in X \times (Y \to R)$, if $\alpha(\sigma, \tau)$ then

1. $\mathrm{P}_{\mathcal{G}}(\sigma) = \mathrm{P}_{\mathcal{H}}(\tau)$; and
2. $\sigma' \in \mathrm{B}_{\mathcal{G}}(c)(\sigma) \implies \exists \tau' \in \Sigma_{\mathcal{H}}$ such that $\alpha(\sigma', \tau')$ and $\tau' \in \mathrm{B}_{\mathcal{H}}(c)(\tau)$.

Definition 5.6.5.3. Let $\mathcal{G}, \mathcal{H} : (X, S) \to (Y, R)$ be concrete open games. $\mathcal{G}$ and $\mathcal{H}$ are *equivalent*, written $\mathcal{G} \sim \mathcal{H}$, if there exists a simulation $\alpha : \mathcal{G} \to \mathcal{H}$ where the converse relation $\alpha^c : \mathcal{H} \to \mathcal{G}$ is also a simulation. In this instance we say that α is a *bisimulation* of open games. We write $[\mathcal{G}]$ for the equivalence class of $\mathcal{G}$ under this relation. We also say that the relation α *witnesses* the equivalence between $\mathcal{G}$ and $\mathcal{H}$ and write $\mathcal{G} \overset{\alpha}{\sim} \mathcal{H}$.

We note that this approach is not standard in the open games literature, and that there are multiple sensible definitions for 'maps' between open games. In the paper [Hed18a], Jules Hedges defines *contravariant lens morphisms* between open games as follows.

Definition 5.6.5.4 (Contravariant lens morphism)**.** Let $\mathcal{G}_1 : (X_1, S_1) \to (Y_1, R_1)$ and $\mathcal{G}_2 : (X_2, S_2) \to (Y_2, R_2)$ be open games. A *contravariant lens morphism* $\alpha : \mathcal{G} \to \mathcal{G}'$ consists in

- Concrete lenses $s_\alpha : (X_2, S_2) \to (X_1, S_1)$ and $t_\alpha : (Y_2, R_2) \to (Y_1, R_1)$;

[2]In the literature, serial relations are sometimes called *total* or *entire*.

- and a function $\Sigma_\alpha : \Sigma_{\mathcal{G}_1} \to \Sigma_{\mathcal{G}_2}$

such that, for all $\sigma \in \Sigma_{\mathcal{G}_1}$, the square

$$\begin{array}{ccc} (X_1, S_1) & \xrightarrow{\mathrm{P}_{\mathcal{G}_1}(\sigma)} & (Y_1, R_1) \\ {\scriptstyle s_\alpha}\big\uparrow & & \big\uparrow{\scriptstyle t_\alpha} \\ (X_2, S_2) & \xrightarrow[\mathrm{P}_{\mathcal{G}_2}(\Sigma_\alpha(\sigma))]{} & (Y_2, R_2) \end{array}$$

commutes and, for all $(x_2, k_1) \in X_2 \times (Y_1 \to R_1)$ and $\sigma_1, \sigma_1' \in \Sigma_{\mathcal{G}_1}$,

$$\sigma_1' \in \mathrm{B}_{\mathcal{G}_1}(s_\alpha \circ x_2^\star, k_1^\star)(\sigma_1) \quad \Longrightarrow \quad \Sigma_\alpha(\sigma_1') \in \mathrm{B}_{\mathcal{G}_2}(x_2^\star, t_\alpha \circ k_1^\star)\big(\Sigma_\alpha(\sigma_1)\big)$$

Using these morphisms Hedges constructs a symmetric monoidal pseudo double category of concrete open games — a level of generality where one can work without needing to take a quotient. Moreover, if one views open games as the *objects* of a double category (rather than as morphisms as we have been doing in this chapter), one can talk about open games specified by *universal properties.*

One other approach to defining maps between open games is given by Neil Ghani et al. in the paper [GKLF18], though this approach is a means to a different end. Namely, providing a definition of *infinitely repeated games* as final coalgebras.

The following results demonstrate that sequential and tensor composition of concrete open games respects equivalence of concrete open games.

Lemma 5.6.5.5. *Let $\mathcal{G}, \mathcal{G}' : (X, S) \to (Y, R)$ and $\mathcal{H}, \mathcal{H}' : (Y, R) \to (Z, Q)$ be concrete open games. If $\mathcal{G} \sim \mathcal{G}'$ and $\mathcal{H} \sim \mathcal{H}'$, then $\mathcal{G} \circ \mathcal{H} \sim \mathcal{G}' \circ \mathcal{H}'$.*

Proof. Suppose $\mathcal{G} \overset{\alpha}{\sim} \mathcal{H}$ and $\mathcal{G}' \overset{\beta}{\sim} \mathcal{H}'$. Then $\alpha \times \beta : \Sigma_\mathcal{G} \times \Sigma_\mathcal{G} \twoheadrightarrow \Sigma_{\mathcal{G}'} \times \Sigma_{\mathcal{H}'}$ given by

$$(\alpha \times \beta)\big((\sigma, \tau), (\sigma', \tau') \Leftrightarrow \alpha(\sigma, \sigma') \text{ and } \beta(\tau, \tau')$$

is such that $\mathcal{G} \otimes \mathcal{H} \overset{\alpha\times\beta}{\sim} \mathcal{G}' \otimes \mathcal{H}'$. □

Lemma 5.6.5.6. *Let $\mathcal{G}, \mathcal{H} : (X, S) \to (Y, R)$ and $\mathcal{G}', \mathcal{H}' : (X', S') \to (Y', R')$ be concrete open games. If $\mathcal{G} \sim \mathcal{H}$ and $\mathcal{G}' \sim \mathcal{H}'$, then $\mathcal{G} \otimes \mathcal{G}' \sim \mathcal{H} \otimes \mathcal{H}'$.*

Proof. If $\mathcal{G} \overset{\alpha}{\sim} \mathcal{H}$ and $\mathcal{G}' \overset{\beta}{\sim} \mathcal{H}'$, then $\mathcal{G} \otimes \mathcal{G}' \overset{\alpha\times\beta}{\sim} \mathcal{H} \otimes \mathcal{H}'$ as in the previous lemma. □

In addition to its utility for defining a category of concrete open games, quotienting by equivalence rules out some pathological behaviour. Given a game $\mathcal{G} : (X, S) \to (Y, R)$, we can define $\mathcal{G}' : (X, S) \to (Y, R)$ where $\Sigma_{\mathcal{G}'} = \Sigma_{\mathcal{G}} + \Sigma_{\mathcal{G}}$ and where the behaviour of $\mathcal{G}'$ is the same on both copies of $\Sigma_{\mathcal{G}}$. $\mathcal{G}$ and $\mathcal{G}'$ are distinct, but equivalent concrete open games.

Example 5.6.5.7. Let $i_l, i_r : A \to A+A$ be the *left* and *right* injections respectively. If $a \in i_l(A)$, then a is a member of the 'left' copy of A in $A + A$. If $a \in i_r(A)$, a is a member of the 'right' copy. Let $\mathcal{G} : (X, S) \to (Y, R)$ be a concrete open game. Define $\mathcal{G}' : (X, S) \to (Y, R)$ by

1. $\Sigma_{\mathcal{G}'} = \Sigma_{\mathcal{G}} + \Sigma_{\mathcal{G}}$;

2. $\mathrm{P}_{\mathcal{G}'}(\sigma) = \mathrm{P}_{\mathcal{G}}(\sigma)$; and

3. For all $c \in X \times (Y \to R)$, $x \in \{l, r\}$, and $\sigma \in i_x$,

$$\mathrm{B}_{\mathcal{G}'}(c)(\sigma) = \left\{ \tau \in i_x(A) \;\middle|\; \tau \in \mathrm{B}_{\mathcal{G}}(c)(\sigma) \right\}.$$

Let $\beta : \Sigma_{\mathcal{G}} \twoheadrightarrow \Sigma_{\mathcal{G}} + \Sigma_{\mathcal{G}}$ be the relation given by $\beta(\sigma) = \{i_l(\sigma), i_r(\sigma)\}$. Then β witnesses the equivalence between $\mathcal{G}$ and $\mathcal{G}'$.

We will have more to say about the meaning of this quotient in 5.8, but for now we will simply show that it allows us to form a symmetric monoidal category of concrete open games.

5.6.6 The category of concrete open games

We are now finally in a position to show that concrete open games form a symmetric monoidal category.

Notation 5.6.6.1. In string diagrams we refer to a play function applied to a strategy simply by the strategy. For example, σ may refer to $\mathrm{P}_{\mathcal{G}}(\sigma)$. In practice this does not lead to ambiguity because proofs and definitions proceed by assigning fixed strategies to particular open games. This notational convention allows for less cluttered string diagrams.

Lemma 5.6.6.2. *Sequential composition of concrete open game equivalence classes is associative.*

Proof. Suppose we have concrete open games

$$(X,S) \xrightarrow{\mathcal{G}} (Y,R) \xrightarrow{\mathcal{H}} (Z,Q) \xrightarrow{\mathcal{K}} (W,T)\ .$$

Define a relation $\beta : (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times \Sigma_{\mathcal{K}} \twoheadrightarrow \Sigma_{\mathcal{G}} \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{K}})$ by $\beta\Big(\big((\sigma,\tau),\mu\big),\big(\sigma,(\tau,\mu)\big)\Big)$. $\mathrm{P}_{\mathcal{K}\circ(\mathcal{H}\circ\mathcal{G})}((\sigma,\tau),\mu) = \mathrm{P}_{(\mathcal{K}\circ\mathcal{H})\circ\mathcal{G}}(\sigma,(\tau,\mu))$ as composition in **CL** is associative.

Also,

$$\begin{aligned}
&\mathrm{B}_{\mathcal{K}\circ(\mathcal{H}\circ\mathcal{G})}(x^\star,k_\star)((\sigma,\tau),\mu)\\
=&\mathrm{B}_{\mathcal{H}\circ\mathcal{G}}\big(x^\star,k_\star\circ\mathrm{P}_{\mathcal{K}}(\mu)\big)(\sigma,\tau)\times\mathrm{B}_{\mathcal{K}}\big(\mathrm{P}_{\mathcal{H}\circ\mathcal{G}}(\sigma,\tau)\circ x^\star,k_\star\big)(\mu)\\
=&\Big(\mathrm{B}_{\mathcal{G}}\big(x^\star,k_\star\circ\mathrm{P}_{\mathcal{K}}(\mu)\circ\mathrm{P}_{\mathcal{H}}(\tau)\big)(\sigma)\times\mathrm{B}_{\mathcal{H}}\big(\mathrm{P}_{\mathcal{G}}(\sigma)\circ x^\star,k_\star\circ\mathrm{P}_{\mathcal{K}}(\mu)\big)(\tau)\\
&\times\mathrm{B}_{\mathcal{K}}\big(\mathrm{P}_{\mathcal{H}\circ\mathcal{G}}(\sigma,\tau)\circ x^\star,k_\star\big)(\mu)\Big)\\
\overset{\beta}{\sim}&\mathrm{B}_{\mathcal{G}}\big(x^\star,k_\star\circ\mathrm{P}_{\mathcal{K}\circ\mathcal{H}}(\tau,\mu)\big)(\sigma)\times\Big(\mathrm{B}_{\mathcal{H}}\big(\mathrm{P}_{\mathcal{G}}(\sigma)\circ x^\star,k_\star\circ\mathrm{P}_{\mathcal{K}}(\mu)\big)(\tau)\\
&\times\mathrm{B}_{\mathcal{K}}\big(\mathrm{P}_{\mathcal{K}}(\mu)\circ\mathrm{P}_{\mathcal{H}}(\tau)\circ x^\star,k_\star\big)(\mu)\Big)\\
=&\mathrm{B}_{\mathcal{G}}\big(x^\star,k_\star\circ\mathrm{P}_{\mathcal{K}\circ\mathcal{H}}(\tau,\mu)\big)(\sigma)\times\mathrm{B}_{\mathcal{K}\circ\mathcal{H}}\big(\mathrm{P}_{\mathcal{G}}(\sigma)\circ x^\star,k_\star\circ\mathrm{P}_{\mathcal{K}}(\mu)\big)(\tau,\mu)\\
=&\mathrm{B}_{(\mathcal{K}\circ\mathcal{H})\circ\mathcal{G}}(x^\star,k_\star)(\sigma,(\tau,\mu))
\end{aligned}$$

Basic checks verify that β does witness an equivalence between $\mathcal{K}\circ(\mathcal{H}\circ\mathcal{G})$ and $(\mathcal{K}\circ\mathcal{H})\circ\mathcal{G}$. □

Remark 5.6.6.3. In the above proof we handwave slightly in the best response calculation where we use the relation β. Really, we are manipulating the best response function until it is obvious why β witnesses an equivalence between the two open games. The gory details are easy, but also tedious and not illuminating.

The identity morphism $(X,S)\to(X,S)$ is given by the computation $\langle \mathrm{id}_X, \mathrm{id}_S\rangle$.

Lemma 5.6.6.4. *Let* $\mathcal{G} : (X,S) \to (Y,R)$*. Then* $[\mathcal{G}] = [\mathcal{G}\circ\langle \mathrm{id}_X,\mathrm{id}_S\rangle] = [\langle \mathrm{id}_Y,\mathrm{id}_R\rangle\circ\mathcal{G}]$.

Proof. Define a relation $\beta : \Sigma_{\mathcal{G}} \twoheadrightarrow \Sigma_{\mathcal{G}} \times \{\star\}$ by $\beta(\sigma,(\star,\sigma))$. Then $\mathcal{G} \overset{\beta}{\sim} (\langle \mathrm{id}_Y,\mathrm{id}_R\rangle\circ\mathcal{G})$. A similar argument works for $\mathcal{G}\circ\langle \mathrm{id}_X,\mathrm{id}_S\rangle$. □

Corollary 5.6.6.5. *There is a category* **ConGame** *with object class* **Set** $\times$ **Set** *and equivalence classes of concrete open games as morphisms.* □

We now move on to proving that **ConGame** is symmetric monoidal.

Lemma 5.6.6.6. $\otimes$: **ConGame** $\times$ **ConGame** $\to$ **ConGame** *is a functor.*

Proof. Suppose we have concrete open games

$$(X,S) \xrightarrow{\mathcal{G}} (Y,R) \xrightarrow{\mathcal{H}} (Z,Q)$$

$$(X',S') \xrightarrow[\mathcal{G}']{} (Y',R') \xrightarrow[\mathcal{H}']{} (Z',Q')$$
.

Let $\sigma \in \Sigma_{\mathcal{G}}, \tau \in \Sigma_{\mathcal{H}}, \sigma' \in \Sigma_{\mathcal{G}'}$, and $\tau' \in \Sigma_{\mathcal{H}'}$. Let $\beta : (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times (\Sigma_{\mathcal{G}'} \times \Sigma_{\mathcal{H}'}) \twoheadrightarrow (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{G}'}) \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{H}'})$ be the relation generated by

$$\big((\sigma,\tau),(\sigma',\tau')\big) \overset{\beta}{\sim} \big((\sigma,\sigma'),(\tau,\tau')\big).$$

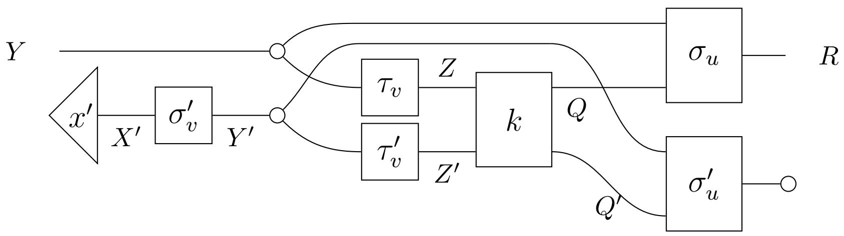
As **CL** is symmetric monoidal,

$$\mathrm{P}_{(\mathcal{H}\circ\mathcal{G})\otimes(\mathcal{H}'\circ\mathcal{G}')}\big((\sigma,\tau),(\sigma',\tau')\big) = \mathrm{P}_{(\mathcal{H}\otimes\mathcal{H}')\circ(\mathcal{G}\otimes\mathcal{G}')}\big((\sigma,\sigma'),(\tau,\tau')\big).$$

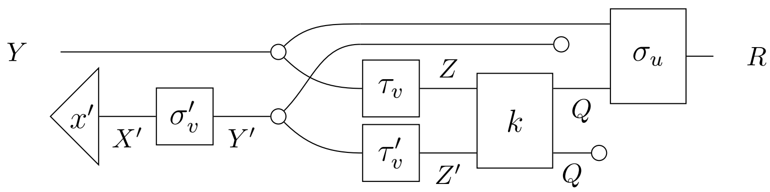
We now proceed by showing that the games $\mathcal{G}, \mathcal{G}', \mathcal{H}$, and $\mathcal{H}'$ have the same local contexts in both $(\mathcal{H}\circ\mathcal{G})\otimes(\mathcal{H}'\circ\mathcal{G}')$ and $(\mathcal{H}\otimes\mathcal{H}')\circ(\mathcal{G}\otimes\mathcal{G}')$. Let $x \in X, x' \in X'$, and $k : Z \times Z' \to Q \times Q'$. For the local outcome function for $\mathcal{G}$, unpacking definitions reveals that

$$k_{\mathcal{G}} := \mathcal{L}\Big(x', \mathrm{P}_{\mathcal{G}}(\sigma'), \big(k_\star \circ \mathrm{P}_{\mathcal{H}\otimes\mathcal{H}'}(\tau,\tau')\big)\Big)$$

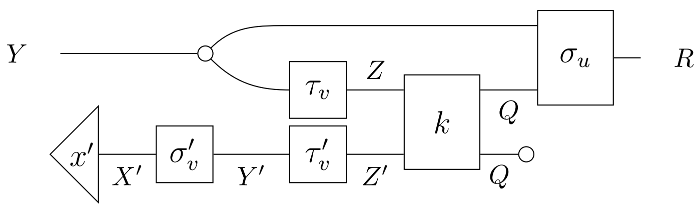
$=$



$=$



$=$



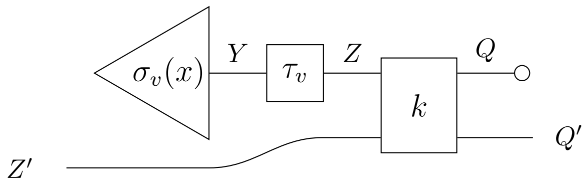
.

$$= \mathcal{L}\Big(x', \mathrm{P}_{\mathcal{H}'\circ\mathcal{G}'}(\sigma', \tau'), k\Big)_{\star} \circ \mathrm{P}_{\mathcal{H}}(\tau)$$

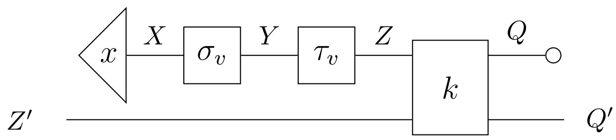
For $\mathcal{H}'$ the local outcome function is given by

$$k_{\mathcal{H}'} := \mathcal{R}\Big(\mathrm{P}_{\mathcal{G}}(\sigma)_v(x), \mathrm{P}_{\mathcal{H}}(\tau_v), k\Big)$$

$$=$$



$$=$$



$$= \mathcal{R}\Big(x, \mathrm{P}_{\mathcal{H}}(\tau)_v \circ \mathrm{P}_{\mathcal{G}}(\sigma)_v, k\Big)$$

The following equalities for the local outcome functions of $\mathcal{G}'$ and $\mathcal{H}$ respectively follow by similar checks.

$$k_{\mathcal{G}'} := \mathcal{R}\Big(x, \mathrm{P}_{\mathcal{G}}(\sigma), k_{\star} \circ \mathrm{P}_{\mathcal{H}\otimes\mathcal{H}'}(\tau, \tau')\Big) = \mathcal{R}\Big(x, \mathrm{P}_{\mathcal{H}\circ\mathcal{G}}(\sigma, \tau), k\Big)_{\star} \circ \mathrm{P}_{\mathcal{H}'}(\tau')$$

$$k_{\mathcal{H}} := \mathcal{L}\Big(\mathrm{P}_{\mathcal{G}'}(\sigma')_v(x'), \mathrm{P}_{\mathcal{H}'}(\tau')_v, k\Big) = \mathcal{L}\Big(x', \mathrm{P}_{\mathcal{H}'}(\tau')_v \circ \mathrm{P}_{\mathcal{G}'}(\sigma')_v, k\Big)$$

Then

$$\mathrm{B}_{(\mathcal{H}\otimes\mathcal{H}')\circ(\mathcal{G}\otimes\mathcal{G}')}\big((x, x')^{\star}, k_{\star}\big)\big((\sigma, \sigma'), (\tau, \tau')\big)$$

$$= \Big(\mathrm{B}_{\mathcal{G}}\big(x^{\star}, k_{\mathcal{G}_{\star}}\big)(\sigma) \times \mathrm{B}_{\mathcal{G}'}\big(x'^{\star}, k_{\mathcal{G}'_{\star}}\big)(\sigma')\Big) \times \Big(\mathrm{B}_{\mathcal{H}}\big(\mathrm{P}_{\mathcal{G}}(\sigma)_v(x)^{\star}, k_{\mathcal{H}_{\star}}\big)\tau \times \mathrm{B}_{\mathcal{H}'}\big(\mathrm{P}_{\mathcal{G}'}(\sigma')_v(x')^{\star}, k_{\mathcal{H}'_{\star}}\big)\Big)$$

$$\overset{\beta}{\sim} \Big(\mathrm{B}_{\mathcal{G}}\big(x^{\star}, k_{\mathcal{G}_{\star}}\big)(\sigma) \times \mathrm{B}_{\mathcal{H}}\big(\mathrm{P}_{\mathcal{G}}(\sigma)_v(x)^{\star}, k_{\mathcal{H}}\big)(\tau)\Big) \times \Big(\mathrm{B}_{\mathcal{G}'}\big(x', k_{\mathcal{G}'}\big)(\sigma') \times \mathrm{B}_{\mathcal{H}'}\big(\mathrm{P}_{\mathcal{G}'}(\sigma')_v(x')^{\star}, k_{\mathcal{H}'}\big)\Big)$$

$$= \mathrm{B}_{(\mathcal{H}\circ\mathcal{G})\otimes(\mathcal{H}'\circ\mathcal{G}')}\big((x, x'), k\big)\big((\sigma, \tau), (\sigma', \tau')\big)$$

All that remains is to show that $\mathrm{id}_{(X,S)} \otimes \mathrm{id}_{(X',S')} \sim \mathrm{id}_{(X\times X', S\times S')}$. This equivalence is witnessed by the relation $\gamma : \{\star\} \times \{\star\} \twoheadrightarrow \{\star\}$ given by the total relation $\gamma\big((\star, \star), \star\big)$.

□

Lemma 5.6.6.7. *The associator in* **ConGame** *is natural.*

Proof. Let $\mathcal{G}_i : (X_i, S_i) \to (Y_i, R_i)$ be open games where $i \in \{1, 2, 3\}$. We need to show that $\alpha \circ (\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3)) \sim ((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3) \circ \alpha$. Define $\beta : (\Sigma_{\mathcal{G}_1} \times (\Sigma_{\mathcal{G}_2} \times \Sigma_{\mathcal{G}_3})) \times \{\star\} \twoheadrightarrow \{\star\} \times ((\Sigma_{\mathcal{G}_1} \times \Sigma_{\mathcal{G}_2}) \times \Sigma_{\mathcal{G}_3})$ by $\big((\sigma, (\tau, \mu)), \star\big) \overset{\beta}{\sim} \big(\star, ((\sigma, \tau), \mu)\big)$.

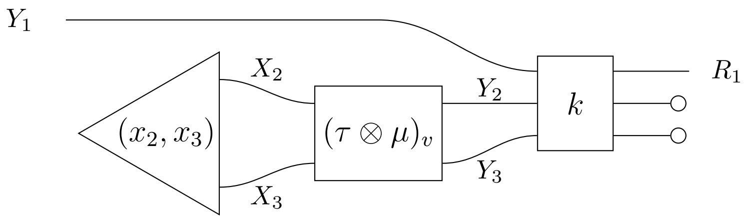
As **CL** is symmetric monoidal, we have that

$$\alpha \circ \mathrm{P}_{\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3)}(\sigma, (\tau, \mu)) = \mathrm{P}_{(\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_2}((\sigma, \tau), \mu) \circ \alpha.$$

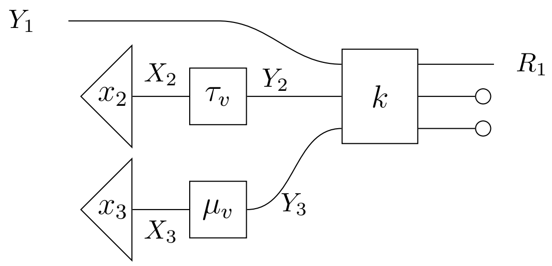
Let $x_i \in X_i$ and $k : (Y_1 \times Y_2 \times Y_3) \to (R_1 \times R_2 \times R_3)$. We will show that the local contexts for $\mathcal{G}_1, \mathcal{G}_2$, and $\mathcal{G}_3$ are the same in both $\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_2)$ and $(\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3$. First we consider $\mathcal{G}_1$. Let $k' := \mathcal{L}(x_3, \mathrm{P}_{\mathcal{G}_3}\mu, k)$. Then,

$$k_{\mathcal{G}_1} := \mathcal{L}\big((x_2, x_3), \mathrm{P}_{\mathcal{G}_2 \otimes \mathcal{G}_3}(\tau, \mu), k\big)$$

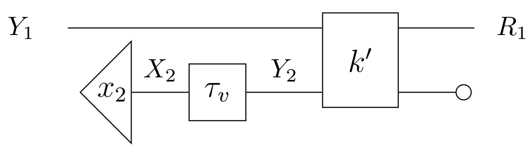
$=$



$=$



$=$



$$= \mathcal{L}\Big(x_2, \mathrm{P}_{\mathcal{G}_2}(\tau), \mathcal{L}\big(x_3, \mathrm{P}_{\mathcal{G}_3}\mu, k\big)\Big)$$

Similar arguments hold for $\mathcal{G}_2$ and $\mathcal{G}_3$, showing that

$$k_{\mathcal{G}_2} := \mathcal{R}\Big(x_1, \mathrm{P}_{\mathcal{G}_1}(\sigma), \mathcal{L}\big(x_3, \mathrm{P}_{\mathcal{G}_3}(\mu), k\big)\Big) = \mathcal{L}\Big(x_3, \mathrm{P}_{\mathcal{G}_3}(\mu), \mathcal{R}\big(x_1, \mathrm{P}_{\mathcal{G}_1}(\sigma), k\big)\Big)$$

and

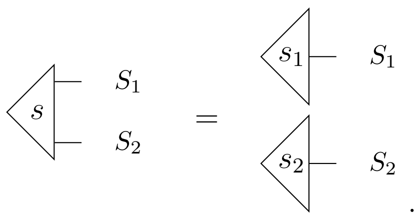
$$k_{\mathcal{G}_3} := \mathcal{R}\Big((x_1, x_2), \mathrm{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma, \tau), k\Big) = \mathcal{R}\Big(x_2, \mathrm{P}_{\mathcal{G}_2}(\tau), \mathcal{R}\big(x_1, \mathrm{P}_{\mathcal{G}_1}(\sigma), k\big)\Big).$$

Then

$$\begin{aligned}
&\mathrm{B}_{\alpha \circ (\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3))}\big((x_1, (x_2, x_3)), k\big)\big(\star, (\sigma, (\tau, \mu))\big) \\
=&\mathrm{B}_\star \times \Bigg(\mathrm{B}_{\mathcal{G}_1}(x_1, k_{\mathcal{G}_1})(\sigma) \times \Big(\mathrm{B}_{\mathcal{G}_2}(x_2, k_{\mathcal{G}_2})(\tau) \times \mathrm{B}_{\mathcal{G}_3}(x_3, k_{\mathcal{G}_3})(\mu)\Big)\Bigg) \\
\overset{\beta}{\sim}&\Bigg(\Big(\mathrm{B}_{\mathcal{G}_1}(x_1, k_{\mathcal{G}_1})(\sigma) \times \mathrm{B}_{\mathcal{G}_2}(x_2, k_{\mathcal{G}_2})(\tau)\Big) \times \mathrm{B}_{\mathcal{G}_3}(x_3, k_{\mathcal{G}_3})(\mu)\Bigg) \times \mathrm{B}_\star \\
=&\mathrm{B}_{((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3) \circ \alpha}\big(((x_1, x_2), x_3), k\big)\big(((\sigma, \tau), \mu), \star\big)
\end{aligned}$$

where $\mathrm{B}_\star$ is the total best response relation of α. □

The above lemma relies on the fact that the monoidal tensor in **Set** is cartesian. In particular we needed that bipartite states $s : I \to S_1 \otimes S_2$ in **Set** (i.e. elements of $S_1 \times S_2$) correspond to pairs of states $(s_1 : I \to S_1, s_2 : I \to S_2)$. In an arbitrary monoidal category, it need not be the case that for all states $s : I \to S_1 \otimes S_2$ there exist states $s_1 : I \to S_1$ and $s_2 : I \to S_2$ such that



This poses a significant barrier to generalising concrete open games to monoidal categories where the monoidal tensor is not cartesian, and chapter 6 addresses this problem.

Lemma 5.6.6.8. *The structural computations λ, ρ, and s are natural in* **ConGame**.

Proof. Easy checks. □

Theorem 5.6.6.9. ConGame *is symmetric monoidal.*

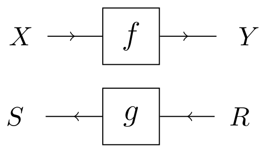
Proof. All that remains to be checked is that the structural maps satisfy the MacLane pentagon and triangle identities, but this follows easily from the symmetric monoidal structure of **Set**. □

5.6.7 Encoding functions as games

Recall that, given functions $f : X \to Y$ and $g : R \to S$, there is a computation concrete open game $\langle f, g\rangle : (X, S) \to (Y, R)$. In fact, this operation is functorial.

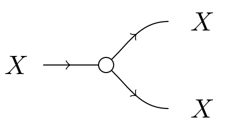
Lemma 5.6.7.1 ([Hed16]). *Define* $F : \mathbf{Set} \times \mathbf{Set}^{\mathrm{op}} \to \mathbf{ConGame}$ *by* $F(X, S) = (X, S)$ *and* $F(f : X \to Y, g : R \to S) = \langle f, g\rangle$. *Then* F *is a faithful monoidal functor.* □

We also incorporate computations directly into the diagrammatic calculus for concrete open games, representing the computation $\langle f, g\rangle : (X, S) \to (Y, R)$ by

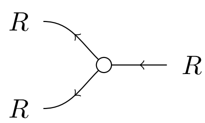


.

Two particularly useful examples of this notation are the covariant and contravariant copying computations $\langle \Delta_X, \mathrm{id}\rangle : (X, I) \to (X \times X, I)$ and $\langle \mathrm{id}, \Delta_R\rangle : (I, R \times R) \to (I, R)$ which are represented by



and



respectively.

5.7 Game theory with concrete open games

In this section we give some examples of games modeled using concrete open games. We will be light on details, aiming to simply demonstrate some of the expressive power of concrete open games. We direct the reader to [Hed16] for more details.

5.7.1 Bimatrix games

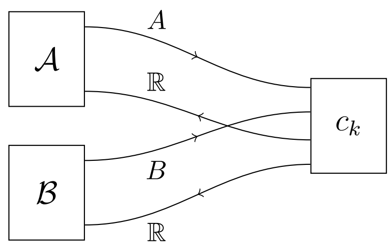
Bimatrix games are simply two player normal form games, the most well-known example of which is likely the prisoners' dilemma, discussed in 2.5.0.2. We assume for simplicity that the set of actions available to each player is finite.

Definition 5.7.1.1. A *bimatrix game* consists in

1. Finite set of actions A and B; and

2. An outcome function $k : A \times B \to \mathbb{R}^2$.

A bimatrix game $\mathcal{G} = (A, B, k)$ is represented by the concrete open game



where $\mathcal{A}$ and $\mathcal{B}$ are utility-maximising agents and c_k is the counit game associated with k. In diagrammatic form, the structure of the game is made clear. Players $\mathcal{A}$ and $\mathcal{B}$ make independent choices from A and B respectively which are then used to generate two real numbers as outcomes. Bimatrix games may not have a Nash equilibrium in pure strategies, but in cases that do have Nash equilibria, they appear as fixed points of the best response function of the above concrete open game.

5.7.2 Deterministic sequential games as open games

There is an obvious mapping from selection functions to concrete open games.

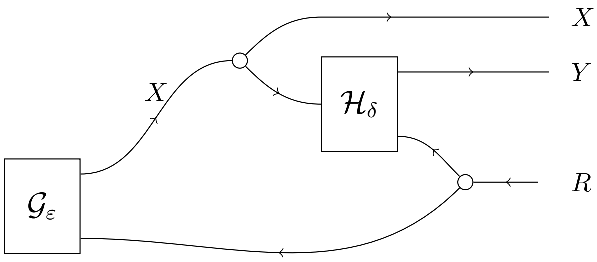
Example 5.7.2.1. Let $\varepsilon : \mathcal{J}_R(X)$ be a single-valued selection function. Define a concrete open game $\mathcal{G}_\varepsilon : I \to (X, R)$ to be the agent with preference function[3]

$$\rho_{\mathcal{G}_\varepsilon}(\star, k : X \to R) = \{\varepsilon(k)\}$$

Given in this form it is not clear how we should compose these open games that correspond to selection functions. Given G_ε and $\mathcal{G}_\delta$, where $\varepsilon : J_R(X)$ and $\delta : J_R(Y)$ there is no obvious operation that forms $\mathcal{G}_{\varepsilon\otimes\delta}$ due to the fact that $\mathcal{G}_\delta$ does not have an observation type of X. This asymmetry between sequential composition of selection functions and sequential composition of open games is not surprising: open games have a notion of *history*, whilst sequential games do not. Nevertheless, given a pair of selection functions, we can still define an open game related to their tensor.

Example 5.7.2.2. Let $\varepsilon : \mathcal{J}_R(X)$ and $\delta : \mathcal{J}_R(Y)$ be single-valued selection functions. Define $\mathcal{G}_{\varepsilon,\delta} : I \to (Y, R)$ to be the concrete open game given by

[3] ε was chosen as the notation for preference functions due to their similarity to selection functions. In this section, we wish to carefully distinguish between the preference functions of open games and the selection functions of sequential games, so we will be using ρ instead.



where $\mathcal{G}_\varepsilon$ is as in the previous example and where $\mathcal{H}_\delta$ is the agent with preference function

$$\rho_{\mathcal{H}_\delta}(x, k : X \times Y \to R) = \{\delta(k(x, -))\}.$$

The obvious question to ask of $\mathcal{G}_{\varepsilon,\delta}$ is whether the fixed points of its best response function given an outcome function k correspond to the value $(\varepsilon \otimes \delta)(k)$. The answer is that, in general, the fixed points do not correspond to the play computed by the tensor of selection functions. This is because selection functions compute subgame perfect plays, whilst the fixed points of the best response function of an open game are Nash equilibria. That is, the fixed points of $\mathrm{B}_{\mathcal{G}_{\varepsilon,\delta}}(\star, k)$ are the *Nash equilibria* for the two-round sequential game $\big((X, Y), (\varepsilon, \delta), k\big)$, which are strategy profiles $(\sigma_1 : X, \sigma_2 : X \to Y$ such that

1. $\sigma_1 = \varepsilon\big(\lambda(x : X).k(x, \sigma_2(x))\big)$; and

2. $\sigma_2(\sigma_1) = \delta\big(\lambda(y : Y).k(\sigma_1, y)\big)$.

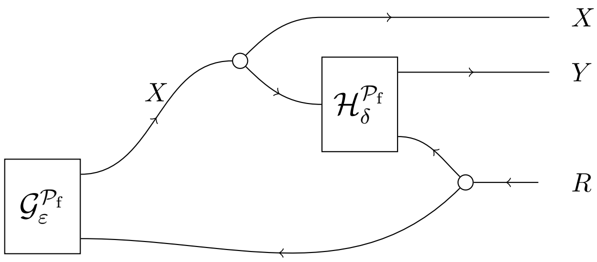
5.7.3 Nondeterministic sequential games as open games

As with deterministic sequential games, there is an obvious mapping from nondeterministic selection functions to concrete open games.

Example 5.7.3.1. Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ be a nondeterministic selection function. Define $\mathcal{G}_\varepsilon^{\mathcal{P}_\mathrm{f}} : I \to (X, R)$ to be the agent with preference function $\rho_{\mathcal{G}_\varepsilon^{\mathcal{P}_\mathrm{f}}}(\star, k : X \to R) = \varepsilon(k)$.

Also, as in the previous subsection, given another nondeterministic selection function δ, we can define an open game related to a nondeterministic sequential game involving ε and δ.

Example 5.7.3.2. Let $\varepsilon : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(X)$ and $\delta : \mathcal{J}_R^{\mathcal{P}_\mathrm{f}}(Y)$ be nondeterministic selection functions. Define $\mathcal{G}_{\varepsilon,\delta}^{\mathcal{P}_\mathrm{f}} : I \to (Y, R)$ to be the concrete open game given by



where $\mathcal{G}_\varepsilon^{\mathcal{P}_\mathrm{f}}$ is as in the previous example and where $\mathcal{H}_\delta^{\mathcal{P}_\mathrm{f}}$ is the agent with preference function

$$\rho_{\mathcal{H}_\delta}(x, k : (X \times Y) \to R) = \delta\big(k(x, -)\big).$$

Whilst, superficially, the structure of the open game above seems similar to the structure of a nondeterministic sequential game $\big((X, Y), (\varepsilon, \delta), k\big)$, they in fact have quite different associated solution concepts. As we saw in Chapter 3, the tensor of nondeterministic selection functions computes the plays of *rational strategies* (when the selection functions are sufficiently well-behaved). Open games, in contrast, compute Nash equilibria which, for a nondeterministic sequential game $\big((X, Y), (\varepsilon, \delta), k\big)$, are strategy profiles $(\sigma_1 : X, \sigma_2 : X \to Y)$ such that

1. $\sigma_1 \in \varepsilon\big(\lambda(x : X).k(x, \sigma_2(x))\big)$; and
2. $\sigma_2(\sigma_1) \in \delta\big(\lambda(y : Y).k(\sigma_1, y)\big)$.

5.7.4 Normal form games

Let $\big((S_i)_{i=1}^n, (k_i)_{i=1}^n\big)$ be a normal form game for n players. Define $k : \prod_{i=1}^n S_i \to \mathbb{R}^n$ by $s = (s_1, \cdots, s_n) \mapsto (q_1(s), \cdots, q_n(s))$. We can model this normal form game using the concrete open game

$$c_k \circ \Big(\bigotimes_{i=1}^n \mathcal{A}_i\Big)$$

where $\mathcal{A}_i : I \to (S_i, \mathbb{R})$ is the utility-maximising agent. The fixed points of this game are then the Nash equilibria of the normal form game.

5.8 Problems with open games

The category **ConGame** defined in this chapter has some less-than-ideal properties. Namely,

- Morphisms (open games) are specified, in part, by an arbitrary set of strategy profiles. This seems unnecessary and also means that the category of concrete open games is not locally small.

- The need to quotient by bisimulation adds an additional layer of complexity. A category of open game where no such quotient is necessary would be preferable.

These two problems are related and, also, quite persistent. We will discuss a potential solution to these problems and, in doing so, illustrate why these problems are difficult to resolve.

A natural approach is to define concrete open games in the way atomic concrete open games are specified. That is, a concrete open game $\mathcal{G} : (X, S) \to (Y, R)$ would be given by a set of strategy profiles $\Sigma_{\mathcal{G}} \subseteq \mathbf{CL}\big((X, S), (Y, R)\big)$ and a preference function $\varepsilon_{\mathcal{G}} : \big(X \times (Y \to R)\big) \to \mathrm{Rel}(\Sigma)$. Given another concrete open game $\mathcal{H} : (Y, R) \to (Z, Q)$ given by $(\Sigma_{\mathcal{H}}, \varepsilon_{\mathcal{H}})$, we could then easily define $\Sigma_{\mathcal{H}\circ\mathcal{G}} = \big\{m \circ l \bigm| l \in \Sigma_{\mathcal{G}},\ m \in \Sigma_{\mathcal{H}}\big\}$. This operation is clearly associative, obviating the need to take a quotient. Defining the set of strategy profiles for tensor composition is similarly not problematic. Moreover, bounding sets of strategy profiles by hom-sets in **CL** means that the resulting category would be locally small. Problems begin to arise when one attempts to define sequential composition of preference functions. Consider $\mathcal{G}$ and $\mathcal{H}$ as above. We would expect $\varepsilon_{\mathcal{H}\circ\mathcal{G}}$ to have type $X \times (Z \to Q) \to \mathrm{Rel}(\Sigma_{\mathcal{H}\circ\mathcal{G}})$. If we fix $n \in \Sigma_{\mathcal{H}\circ\mathcal{G}}$, there may be no canonical way of factoring n as $m \circ l$ where $l \in \Sigma_{\mathcal{G}}$ and $m \in \Sigma_{\mathcal{H}}$. Consequently, there may be no canonical way to call $\varepsilon_{\mathcal{G}}$ and $\varepsilon_{\mathcal{H}}$ given n and a context $(x, k) \in X \times (Z \to Q)$.

Abstracting, we can explain why this problem arises by pointing to the two purposes of the set of strategy profiles in our original definition of concrete open games.

1. Concrete lenses describing the structure of a concrete open game are indexed by strategy profiles; and

2. The structure of the set of strategy profiles describes which subgames can unilaterally deviate from the rest of the game. That is, the set of strategy profiles carves up a concrete open game into components over which different agents have control.

In our attempted redefining of concrete open games above, we do not keep enough fine-grained structure about strategies to fulfill the second item on this list.

Given these observations, designing a category of concrete open games where no quotient needs to be taken seems like a difficult problem. We can fare better in making the category of concrete open games locally small. We cannot define open games just as we specify atoms, but we *can* limit ourselves to only those open games that are generated by the atoms and sequential and tensor composition. This results in a locally small category of concrete open games.

Chapter 6

General open games

This chapter constitutes the beating heart of this thesis. We define a notion of 'open game' over arbitrary symmetric monoidal categories, clearing the path for a vastly more expressive theory of open games. In the next chapter we show that this generalisation provides a model of various classes of *Bayesian games*. We will find that there are two primary obstacles to overcome.

1. Concrete lenses do not straightforwardly generalise to arbitrary symmetric monoidal categories; and

2. The notion of best response needs to be suitably modified in order to account for Bayesian updating.

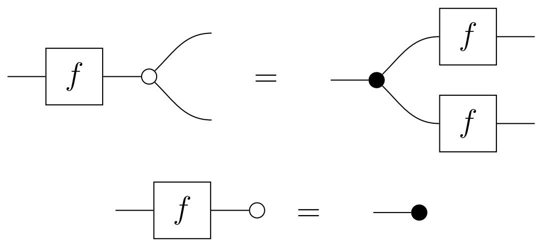
Fortunately, there exists mathematical machinery that will help solve both of these problems: *coends*.

6.1 Chapter overview

In 6.3 we introduce *coends*. In 6.4 we show how coends can be used to generalise concrete lenses. 6.5 discusses the problem of generalising concrete open games to more general monoidal categories. In 6.6, we develop the notion of a *context* for general open games.

6.2 Generalising concrete lenses

In the proof of 5.4.0.4 we made use of the fact that every **Set** function is a comonoid homomorphism for the copy/delete comonoid. Recall that a morphism is a comonoid homomorphism if it can be 'moved through' the comonoid structure.



If **Set** is replaced with some arbitrary symmetric monoidal category $\mathcal{C}$ and the copy/delete comonoid is replaced with some arbitrary comonoid in $\mathcal{C}$, sequential composition of lenses, as defined in 5.4.0.3, may not be associative. This presents a substantive problem — there exist categories relevant to game theory in which sequential composition of concrete lenses is not associative. Of particular interest is the Kleisli category of the finitary distribution monad, $\mathbf{Kl}(D)$, which we will need in order to model Bayesian games (discussed in chapter 7). $\mathbf{Kl}(D)$ inherits a copy/delete comonoid from **Set**, but its comonoid homomorphism are the deterministic maps (i.e. precisely the non-probabilistic maps).

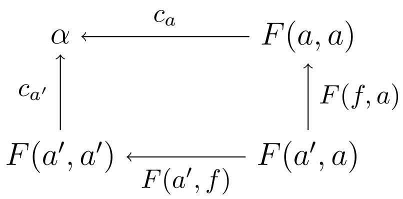
In the next section we introduce some standard categorical machinery — *co-wedges* and *coends*. These will allow us to generalise concrete lenses to categories other than **Set**. We call these generalised lenses *coend lenses* or, simply, *lenses*. The introduction of coend lenses proceeds similarly to the introduction of symmetric monoidal categories (5.2) in that we first introduce the technical notion before describing a graphical notation allowing for more readable proofs.

6.3 Co-wedges and Coends

Co-wedges[1] are a variant of natural transformations applying to functors that act both covariantly and contravariantly on an argument. In 5.3 we noted that lenses have both covariant and contravariant components. We will see that this behaviour can be described by coends, which are initial co-wedges. For extra motivation, discussion, and examples, we refer the reader to [Lor15].

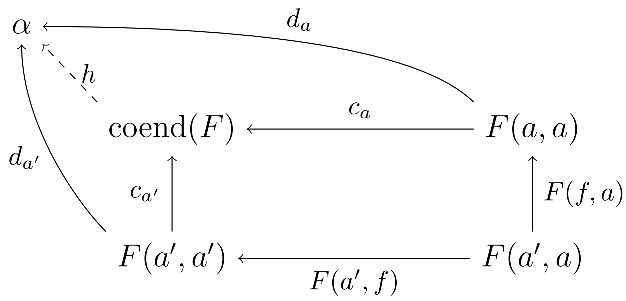
Definition 6.3.0.1 (Co-wedge)**.** Let $F : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$ be a functor. A *co-wedge* $c : F \to \alpha$ is an object $\alpha : \mathcal{D}$ together with maps $\left\{ c_a : F(a,a) \to \alpha \;\middle|\; a : \mathcal{C} \right\}$ such that, for any morphism $f : a' \to a$, the diagram

[1]'Co-wedges' is hyphenated because it was once remarked to me that 'cowedge' could be read 'cow edge' and some things cannot be unseen.



commutes.

Definition 6.3.0.2 (Coend)**.** A *coend* is a couniversal co-wedge. Diagrammatically, the coend of a functor $F : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to D$ is a co-wedge $\{c_a : F(a,a) \to \mathrm{coend}(F) \mid a : \mathcal{C}\}$ such that for any other co-wedge $\{d_a : F(a,a) \to \alpha \mid a : \mathcal{C}\}$ and morphism $f : a' \to a$ the diagram



commutes for a unique morphism $h : \mathrm{coend}(F) \to \alpha$.

We adopt the integral notation for co-ends, writing

$$\int^{a:\mathcal{C}} F(a,a)$$

for coend(F). We will make use of the fact that coends can be characterised by the following coequaliser.

Lemma 6.3.0.3. *Let $F : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$. If $\mathcal{D}$ is cocomplete and $\mathcal{C}$ is small, the coend $\int^{a:\mathcal{C}} F(a,a)$ is given by the coequaliser of the following pair of arrows.*

$$\coprod_{\substack{a,a':\mathcal{C} \\ f:a'\to a}} F(a,a') \underset{F_2}{\overset{F_1}{\rightrightarrows}} \coprod_{a:\mathcal{C}} F(a,a)$$

Where the $f : a' \to a$ components of F_1 and F_2 are $F(f,a')$ and $F(a,f)$ respectively.

□

When $\mathcal{C}$ is *not* small (as is usually the case), we need to show directly that coends exist.

6.4 Coend lenses

The content of this section is due to Mitchell Riley's work in [Ril18], which contains a far more detailed treatment and which serves as a good standard reference for coend lenses. We first give an abstract definition of coend lenses, then provide some justification.

Definition 6.4.0.1 (Coend lens)**.** Let X, S, Y, and R be objects in a symmetric monoidal category $(\mathcal{C}, \otimes)$. A *coend lens* $l : (X, S) \to (Y, R)$ is an element of the set

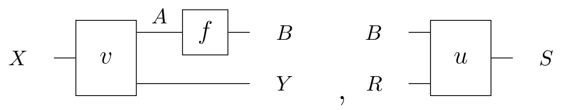
$$\int^{A:\mathcal{C}} \mathcal{C}(X, A \otimes Y) \times \mathcal{C}(A \otimes R, S)$$

We think of the coend in the above definition as acting as a kind of existential quantifier over the type variable A, followed by a quotient (to be described) over the resulting structure. That is, a coend lens $l : (X, S) \to (Y, R)$ consists in an equivalence relation over triples comprised of a choice of type A, a morphism $v : X \to A \otimes Y$, and another morphism $u : A \otimes R \to S$.

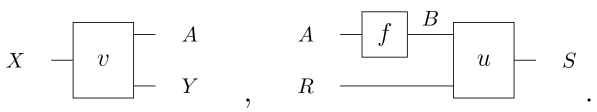
By 6.3.0.3 we can characterise coend lenses $(X, S) \to (Y, R)$ as the elements of a particular coequaliser. Moreover, coequalisers in **Set** are given by quotients. Unpacking the coequaliser explicitly, coend lenses $(X, S) \to (Y, R)$ are given by the set of triples of the form described above, quotiented by the equivalence relation generated by

$$\big((f \otimes \mathrm{id}_Y) \circ v, u\big) \sim \big(v, u \circ (f \otimes \mathrm{id}_R)\big)$$

for all $A, B : \mathcal{C}$ and $f : A \to B$. In diagrammatic form, the pair

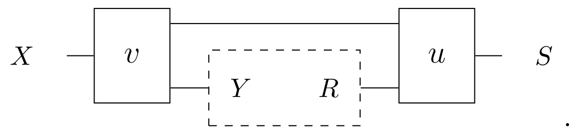


is related to the pair

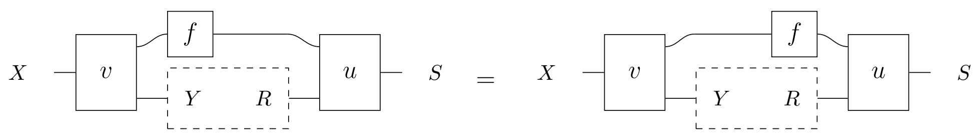


We refer to the types A and B as *bound types* (B is bound in the first diagram, A in the second). In chapter 7, we will see that this bound type keeps track of *correlations* between random variables in the Kleisli category of the distribution monad.

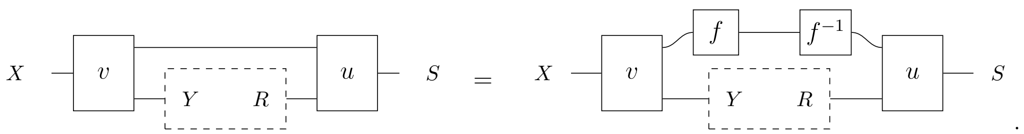
In vague terms, two pairs of morphisms are related if one can get from one to the other by 'sliding' a morphism off the bound type of one morphism on to the bound type of the other. Given a pair of morphisms $(v : X \to A \otimes Y, u : A \otimes R \to S)$, we write $[v, u]$ for their equivalence class. When we need to talk explicitly about the bound type of $[v, u]$ we write $[A, v, u]$ to specify that the pair (v, u) has bound type A. We also adopt the convention that $l = [A_l, l_v, l_u]$ where, as with concrete lenses, we say that l_v is the *view morphism* and l_u is the *update morphism*. We follow [Ril18], taking the hint from the diagrammatic representation of the equivalence relation by representing a coend lens $[v, u] : (X, S) \to (Y, R)$ as



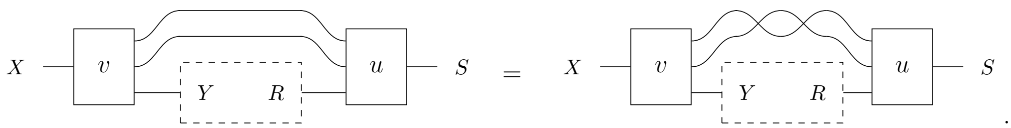
We usually omit the bound type in diagrams for clarity. The equivalence relation is then simply



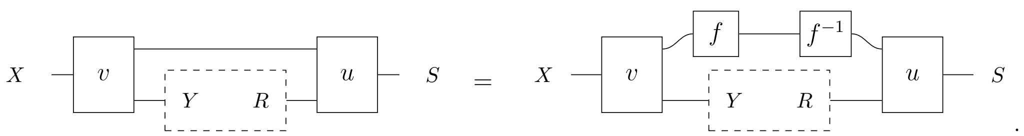
The equivalence relation permits the cancelling of isomorphisms.



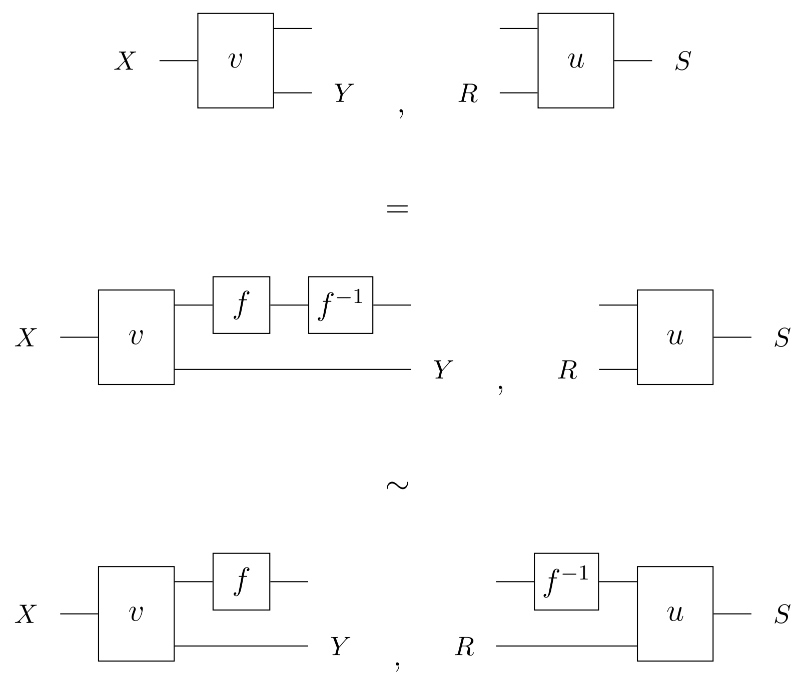
Many proofs in this chapter proceed by allowing symmetric monoidal structure to interact with coend structure as, for example, in the following diagram.



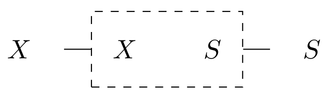
We suspect that this notation corresponds to a graphical calculus on a par with the various calculi for monoidal categories, but verifying this falls outside the scope of this thesis. We consider these diagrams as mere short-hand. The cases relevant to us in this work are certainly not problematic. For instance, the above equation



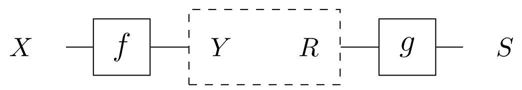
is a shorthand for the following inference:



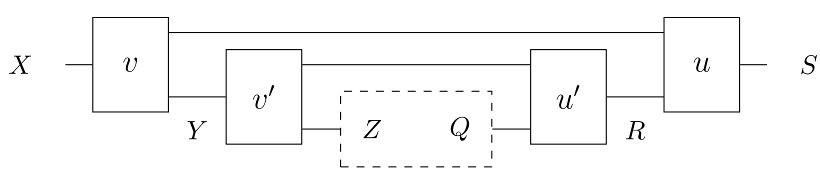
Example 6.4.0.2 (Identity lens)**.** The *identity lens* $\mathrm{id}_{(X,S)} : (X, S) \to (X, S)$ is given by $[\mathrm{id}_X : X \to X, \mathrm{id}_S : S \to S]$. Diagrammatically,



Example 6.4.0.3. A pair of morphisms $(f : X \to Y, g : R \to S)$ is encoded by the coend lens $[I, f, g] : (X, S) \to (Y, R)$:



Definition 6.4.0.4 (Sequential composition of coend lenses)**.** Let $[v, u] : (X, S) \to (Y, R)$ and $[v', u'] : (Y, R) \to (Z, Q)$ be coend lenses. Define $[v', u'] \circ [v, u] : (X, S) \to (Z, Q)$ to be



.

Explicitly,

$$[v', u', A'] \circ [v, u, A] = [(v' \otimes \mathrm{id}_A) \circ v, u \circ (\mathrm{id}_A \otimes u'), A \otimes A'].$$

Theorem 6.4.0.5 (Coend lenses form a category)**.** *Suppose $\mathcal{C}$ is a category such that, for all objects $X, S, Y, R \in \mathcal{C}$,*

$$\int^{A:\mathcal{C}} \mathcal{C}(X, A \otimes Y) \times \mathcal{C}(A \otimes R, S)$$

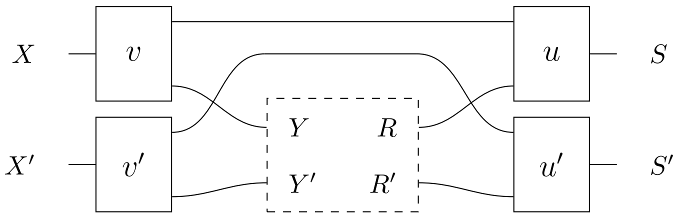
exists. Then there is a category $\mathbf{Lens}_{\mathcal{C}}$ with object class $\mathcal{C} \times \mathcal{C}$ and where

$$\mathbf{Lens}_{\mathcal{C}}\big((X,S),(Y,R)\big) = \int^{A:\mathcal{C}} \mathcal{C}(X, A \otimes Y) \times \mathcal{C}(A \otimes R, S)$$

. □

When $\mathcal{C}$ is small, the existence of the coend lens types are guaranteed by the cocompleteness of **Set**. When $\mathcal{C}$ is not small, and the lens types correspond to coends indexed by a large category, we must verify that these types exist by some other means (by, for example, giving a **Set** isomorphism). Fortunately, this is not difficult for the categories of interest in this work.

Definition 6.4.0.6 (Tensor composition of coend lenses)**.** Let $[v, u] : (X, S) \to (Y, R)$ and $[v', u'] : (X', S') \to (Y', R')$ be coend lenses. Define $[v, u] \otimes [v', u'] : (X \otimes X', S \otimes S') \to (Y \otimes Y', R \otimes R')$ to be



.

Explicitly, $[v, u, A] \otimes [v', u', A']$ is given by

$$\Big[(\mathrm{id}_A \otimes s_{Y,A'} \otimes \mathrm{id}_{Y'}) \circ (v \otimes v'), (u \otimes u') \circ (\mathrm{id}_A \otimes s_{A',R} \otimes \mathrm{id}_{R'}), A \otimes A'\Big]$$

Theorem 6.4.0.7 ($\mathbf{Lens}_{\mathcal{C}}$ is symmetric monoidal)**.** *The category $\mathbf{Lens}_{\mathcal{C}}$ is symmetric monoidal with the tensor given in 6.4.0.6, monoidal unit $I = (I_{\mathcal{C}}, I_{\mathcal{C}})$, and with structural morphisms inherited from $\mathcal{C}$ given by*

$$\begin{aligned}
\alpha_{(X,A),(Y,B),(Z,C)} &= [\alpha_{X,Y,Z}, \alpha^{-1}_{A,B,C}] \\
\lambda_{(X,A)} &= [\lambda_X, \lambda^{-1}_A] \\
\rho_{(X,A)} &= [\rho_X, \rho^{-1}_A] \\
s_{(X,A),(Y,B)} &= [s_{X,Y}, s_{B,A}]
\end{aligned}$$

Lemma 6.4.0.8. *When $\otimes$ is cartesian, $\mathbf{Lens}_{\mathcal{C}}$ is isomorphic to $\mathbf{CL}_{\mathcal{C}}$.* □

6.5 Generalising open games

We could, at this point, attempt to define a (generalised) open game $\mathcal{G} : (X, S) \to (Y, R)$ over a symmetric monoidal category $\mathcal{C}$ as

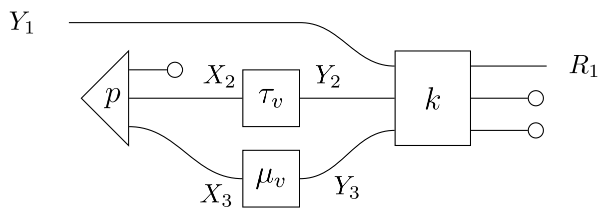
1. A set Σ of strategies;
2. A play function $\mathrm{P} : \Sigma \to \mathbf{Lens}\big((X, S), (Y, R)\big)$; and
3. A best response function $\mathrm{B} : \mathcal{C}(I, X) \times \mathcal{C}(Y, R) \to \mathrm{Rel}(\Sigma)$.

Call such generalised open games *interim open games* (for they will not live long). Sequential composition and tensor composition of interim open games could be defined much as we did for concrete open games. The problems begin to arise when one attempts to prove that this definition results in a symmetric monoidal category.

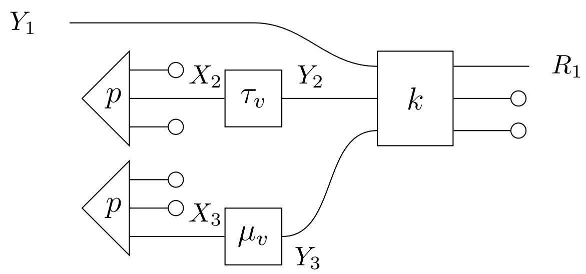
In proving that the associator was natural in **CL**, we used the fact that the monoidal tensor in **Set** is cartesian. If the tensor of $\mathcal{C}$ is *not* cartesian, the local context of $\mathcal{G}$ in $\mathcal{G} \otimes (\mathcal{H} \otimes \mathcal{K})$ is different to the local context of $\mathcal{G}$ in $(\mathcal{G} \otimes \mathcal{H}) \otimes \mathcal{K}$. Let

$$\begin{aligned} \mathcal{G} &: (X_1, S_1) \to (Y_1, R_1) \\ \mathcal{H} &: (X_2, S_2) \to (Y_2, R_2) \\ \mathcal{K} &: (X_3, S_3) \to (Y_3, R_3) \end{aligned}$$

be interim open games, $p \in \mathcal{C}(I, X_1 \otimes X_2 \otimes X_3)$, $k \in \mathcal{C}(Y_1 \otimes Y_2 \otimes Y_3, R_1 \otimes R_2 \otimes R_3)$, and $(\sigma, \tau, \mu) \in \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{K}}$. The local context of $\mathcal{G}$ in $\mathcal{G} \otimes (\mathcal{H} \otimes \mathcal{K})$ is given by



whilst the local context of $\mathcal{G}$ in $(\mathcal{G} \otimes \mathcal{H}) \otimes \mathcal{K}$ is given by



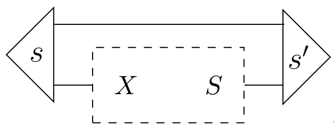
.

In general, these morphisms are *not* the same. In the case where $\mathcal{C}$ is the Kleisli category of the distribution monad (to be defined in chapter 6), the first morphism contains information about correlations between the types X_2 and X_3 whilst the second morphism does not. Consequently, the distinction between these two local contexts for $\mathcal{G}$ is substantive. Fortunately, coend lenses also provide a solution to this problem.

The high level approach for defining a category of generalised open games is to use as few 'deleting' maps as possible. We do this by 'hiding' information in the bound variable of a coend lens whenever we would otherwise delete it. A consequence of this approach is that the correct definition of a 'context' for generalised open games is quite abstract, but we will see this abstractness allows for more elegant proofs and, in any case, disappears when dealing with the categories we are actually interested in.

6.6 States, continuations, and contexts

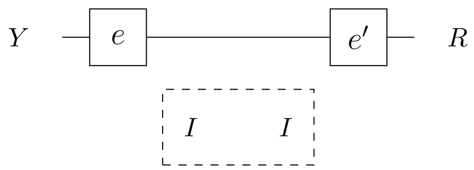
In this section we define a generalised notion of *context* for open games. Observe that a state $[s, s'] \in \mathbf{Lens}_{\mathcal{C}}(I, (X, S))$ has the form



.

More verbosely, a state $s \in \mathbf{Lens}_{\mathcal{C}}(I, (X, S))$ is the equivalence class of a choice of type $A : \mathcal{C}$ together with a state $s : \mathcal{C}(I, A \otimes X)$ in $\mathcal{C}$ and an effect $s' : \mathcal{C}(A \otimes S, I)$ in $\mathcal{C}$. We think of the state s as a *history* and we call the effect s' a *cohistory*[2].

An effect $[e, e'] \in \mathbf{Lens}_{\mathcal{C}}((Y, R), I)$ has the form



.

Concerning effects, we have the following result.

Lemma 6.6.0.1. $\mathcal{C}(Y, R) \cong \mathbf{Lens}_{\mathcal{C}}((Y, R), I)$

[2]Cohistories are not yet well understood. They make proofs easier, but vanish in categories which make game theoretic sense.

Proof. The isomorphism $i : \mathcal{C}(Y, R) \to \mathbf{Lens}_{\mathcal{C}}((X, S), (Y, R))$ is given by

$$i(f : Y \to R) = [R, f, \mathrm{id}_R] = [Y, \mathrm{id}_Y, f]$$

□

This result captures the idea that 'effects in $\mathbf{Lens}_{\mathcal{C}}$ are outcome functions in $\mathcal{C}$.'

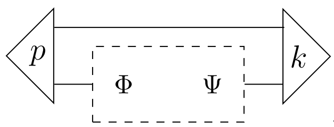
We can now define *(generalised) contexts* which consist in a coend over a state in $\mathbf{Lens}_{\mathcal{C}}$ (a history/cohistory pair) and an effect in $\mathbf{Lens}_{\mathcal{C}}$ (an outcome function). Contexts are therefore members of a *iterated coend.* This iterated coend turns out to be a state in the iterated lens category $\mathbf{Lens}_{\mathbf{Lens}_{\mathcal{C}}}$. From a purely technical standpoint, using iterated lenses allows for elegant proofs. From a heuristics perspective, we will see that the extra bound variable the iterated lens affords us enables us, in the case $\mathcal{C} = \mathbf{Kl}(D)$, to store information about correlations between variables where we would otherwise have to take marginals.

Definition 6.6.0.2 (Context functor)**.** [3] The *context functor* $\mathbb{C} : \mathbf{Lens}_{\mathcal{C}} \times \mathbf{Lens}_{\mathcal{C}}^{\mathrm{op}} \to \mathbf{Set}$ is given by

$$\begin{aligned} \mathbb{C}(\Phi, \Psi) &= \int^{\Theta : \mathbf{Lens}_{\mathcal{C}}} \mathbf{Lens}_{\mathcal{C}}(I, \Theta \otimes \Phi) \times \mathbf{Lens}_{\mathcal{C}}(\Theta \otimes \Psi, I) \\ &= \mathbf{Lens}_{\mathbf{Lens}_{\mathcal{C}}}(I, (\Phi, \Psi)) \end{aligned}$$

Elements of $\mathbb{C}(\Phi, \Psi)$ are called *contexts.*

As a context $[p, k] \in \mathbb{C}(\Phi, \Psi)$ is just a state in $\mathbf{Lens}_{\mathbf{Lens}_{\mathcal{C}}}$, it admits a graphical representation as



.

This is neat, and means many of the proofs in the rest of this chapter can be carried out graphically.

[3]This particular form of the context functor is due to Guillaume Boisseau, and was communicated to the author in person. It is a refinement of the original definition of a context for an open game, which can be seen in 6.11, which is due to the author.

6.7 General open games

We have now arrived at a level of generality where we can define generalised open games in a way that is obviously analogous to concrete open games. Given $\Phi, \Psi \in \mathbf{Lens}_{\mathcal{C}}$, an open game consists in a set of strategy profiles, a family of lenses indexed by the set of strategy profiles, and a best response function which takes a context as input and returns a relation on strategy profiles.

Definition 6.7.0.1 (Open game). Let $\Phi, \Psi \in \mathbf{Lens}_{\mathcal{C}}$. An *open game* $\mathcal{G} : \Phi \to \Psi$ consists in

1. A set of *strategy profiles* Σ;
2. A *play function* $\mathrm{P} : \Sigma \to \mathbf{Lens}_{\mathcal{C}}(\Phi, \Psi)$; and
3. A *best response function* $\mathrm{B} : \mathbb{C}(\Phi, \Psi) \to \mathrm{Rel}(\Sigma)$.

The rationale here is much the same as it is with concrete open games. The play function takes a strategy profile as input and returns a lens describing an open play of the game. Best response takes a context as argument that provides the information necessary for the game to make informed strategic decisions, and returns a relation on strategies.

As with concrete open games, we define a notion of *atomic open game*.

Definition 6.7.0.2. An *atomic open game* $a : \Phi \to \Psi$ is an open game such that

1. $\Sigma_a \subseteq \mathbf{Lens}_{\mathcal{C}}(\Phi, \Psi)$;
2. For all $l \in \Sigma_a$, $\mathrm{P}_a(l) = l$; and
3. For all contexts $c \in \mathbb{C}(\Phi, \Psi)$, $\mathrm{B}_a(c)$ is constant.

Atomic open games are uniquely specified by a subset $\Sigma \subseteq \mathbf{Lens}_{\mathcal{C}}(\Phi, \Psi)$ and a preference function $\varepsilon : \mathbb{C}(\Phi, \Psi) \to \mathcal{P}(\Sigma)$. We refer to atomic open games simply as *atoms*.

Example 6.7.0.3. The *identity atom* $\mathrm{id}_\Phi : \Phi \to \Phi$ is given by $\Sigma = \{\mathrm{id}_\Phi\}$, $\varepsilon(c) = \{\mathrm{id}_\Phi\}$ for all $c \in \mathbb{C}(\Phi, \Psi)$.

Example 6.7.0.4 (Computation). Let $f : \mathcal{C}(X, Y)$ and $g : \mathcal{C}(R, S)$ be morphisms in $\mathcal{C}$. Define the atom $\langle f, g \rangle : (X, S) \to (Y, R)$ by

1. $\Sigma_{\langle f,g \rangle} = \{[f, g]\}$; and
2. $\mathrm{B}_{\langle f,g \rangle}(c) = \{[f, g]\}$ for all $c \in \mathbb{C}\big((X, S), (Y, R)\big)$.

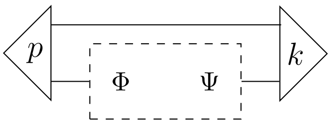
6.7.1 Composing open games

The heuristic for sequential composition of general open games is much the same as for concrete open games in 5.6.3. The only difference is that we are now using coend lenses rather than concrete lenses, and contexts also are slightly different. Best response of a sequential composite $\mathcal{H} \circ \mathcal{G}$ is still defined by forming local contexts for $\mathcal{G}$ and $\mathcal{H}$.

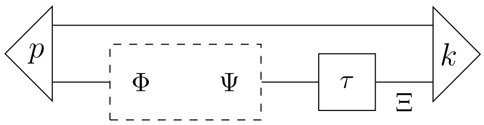
Definition 6.7.1.1 (Sequential composition of open games)**.** Let $\mathcal{G} : \Phi \to \Psi$ and $\mathcal{H} : \Psi \to \Xi$ be open games. Define $\mathcal{H} \circ \mathcal{G} : \Phi \to \Xi$ by

1. $\Sigma_{\mathcal{H}\circ\mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$,
2. $\mathrm{P}_{\mathcal{H}\circ\mathcal{G}}(\sigma, \tau) = \mathrm{P}_{\mathcal{H}}(\tau) \circ \mathrm{P}_{\mathcal{G}}(\sigma)$,
3. $\mathrm{B}_{\mathcal{H}\circ\mathcal{G}}[p, k](\sigma, \tau) = \mathrm{B}_{\mathcal{G}}[p, k \circ \mathrm{P}_{\mathcal{H}}(\tau)](\sigma) \times \mathrm{B}_{\mathcal{H}}[\mathrm{P}_{\mathcal{G}}(\sigma) \circ p, k](\tau)$.

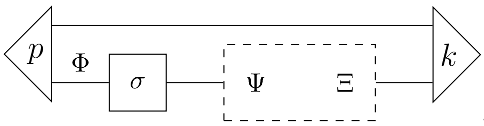
Given a context $[p, k] \in \mathbb{C}(\Phi, \Xi)$ represented by the diagram



the local context for $\mathcal{G}$ given a strategy $\tau \in \Sigma_{\mathcal{H}}$ is given by



and given a strategy $\sigma \in \Sigma_{\mathcal{G}}$ the local context for $\mathcal{H}$ is given by



.

In this representation the process of taking a local context is non-arbitrary, and obviously associative.

6.7.2 The tensor of open games

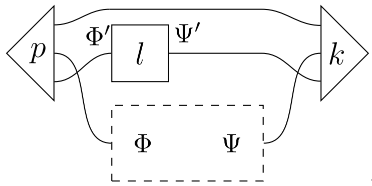
Again, the heuristic for defining the tensor of open games is much as it was for concrete open games. We will first formalise the notion of 'local context' for tensored general open games.

Definition 6.7.2.1 (Local contexts for tensor composition)**.** Define the *left local context function*

$$\mathcal{L}_{\Phi,\Phi',\Psi,\Psi'} : \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') \times \mathbf{Lens}_{\mathcal{C}}(\Phi', \Psi') \to \mathbb{C}(\Phi, \Psi)$$

by

$\mathcal{L}([p, k], l) =$



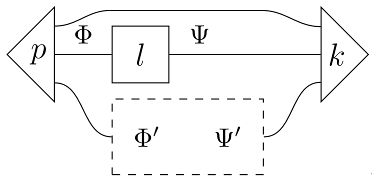
.

Define the *right local context function*

$$\mathcal{R}_{\Phi,\Phi',\Psi,\Psi'} : \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') \times \mathbf{Lens}_{\mathcal{C}}(\Phi, \Psi) \to \mathbb{C}(\Phi', \Psi')$$

by

$\mathcal{R}([p, k], l) =$



.

We will usually suppress the subscripts of $\mathcal{L}$ and $\mathcal{R}$ as the types can be inferred from context.

Definition 6.7.2.2 (Tensor composition of open games)**.** Let $\mathcal{G} : \Phi \to \Psi$ and $\mathcal{H} : \Phi' \to \Psi'$ be open games. Define $\mathcal{G} \otimes \mathcal{H} : \Phi \otimes \Phi' \to \Psi \otimes \Psi'$ by

- $\Sigma_{\mathcal{G}\otimes\mathcal{H}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$;
- $\mathrm{P}_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau) = \mathrm{P}_{\mathcal{G}}(\sigma) \otimes \mathrm{P}_{\mathcal{H}}(\tau)$ (in $\mathbf{Lens}_{\mathcal{C}}$);
- Define $\mathrm{B}_{\mathcal{G}\otimes\mathcal{H}} : \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') \to \mathrm{Rel}(\Sigma_{\mathcal{G}\otimes\mathcal{H}})$ by

$$\mathrm{B}_{\mathcal{G}\otimes\mathcal{H}}(c)(\sigma, \tau) = \mathrm{B}_{\mathcal{G}}(\mathcal{L}(c, \mathrm{P}_{\mathcal{H}}(\tau)))(\sigma) \times \mathrm{B}_{\mathcal{H}}(\mathcal{R}(c, \mathrm{P}_{\mathcal{G}}(\sigma)))(\tau)$$

6.8 Equivalence of open games

As in 5.6.5, we need to quotient open games in order to obtain a category.

Definition 6.8.0.1 (Simulation of open games)**.** Let $\mathcal{G}, \mathcal{H} : \Phi \to \Psi$ be open games. A *simulation* of open games $\alpha : \mathcal{G} \to \mathcal{H}$ is a serial relation $\alpha : \Sigma_{\mathcal{G}} \to \Sigma_{\mathcal{H}}$ such that, for all $\sigma, \sigma' \in \Sigma_{\mathcal{G}}$, $\tau \in \Sigma_{\mathcal{H}}$, and $c : \mathbb{C}(\Phi, \Psi)$, $\alpha(\sigma, \tau)$ implies that

1. $\mathrm{P}_{\mathcal{G}}(\sigma) = \mathrm{P}_{\mathcal{H}}(\tau)$; and

2. $\sigma' \in \mathrm{B}_{\mathcal{G}}(c)(\sigma) \Rightarrow \exists \tau' \in \Sigma_{\mathcal{H}}$ such that $\alpha(\sigma', \tau')$ and $\tau' \in \mathrm{B}_{\mathcal{H}}(c)(\tau)$.

Definition 6.8.0.2 (Equivalence of open games)**.** Let $\mathcal{G}, \mathcal{H} : \Phi \to \Psi$ be open games. $\mathcal{G}$ and $\mathcal{H}$ are *equivalent*, written $\mathcal{G} \sim \mathcal{H}$, if there is a simulation $\alpha : \mathcal{G} \to \mathcal{H}$ such that the converse relation $\alpha^c : \mathcal{H} \to \mathcal{G}$ is also a simulation of open games. We say that α is a *bisimulation of open games* and write $[\mathcal{G}]$ for the equivalence class of $\mathcal{G}$ under this relation.

Lemma 6.8.0.3. *Let $\mathcal{G}, \mathcal{G}' : \Phi \to \Psi$, $\mathcal{H}, \mathcal{H}' : \Psi \to \Xi$, and $\mathcal{K}, \mathcal{K}' : \Phi' \to \Psi'$ be open games. Then*

1. *If $\mathcal{G} \sim \mathcal{G}'$ and $\mathcal{H} \sim \mathcal{H}'$, then $\mathcal{H} \circ \mathcal{G} \sim \mathcal{H}' \circ \mathcal{G}'$; and*

2. *If $\mathcal{G} \sim \mathcal{G}'$ and $\mathcal{K} \sim \mathcal{K}'$, then $\mathcal{G} \otimes \mathcal{K} \sim \mathcal{G}' \otimes \mathcal{K}'$.*

□

Demonstrating equivalence in the cases of interest will always be trivial, and so we simply specify the witnessing relation between strategy sets.

6.9 The category of open games

That equivalences classes of open games form a category follows easily from the fact that coend lenses form a category.

Lemma 6.9.0.1. *Sequential composition of equivalence classes of open games is associative.*

Proof. Suppose we have open games

$$\Phi \xrightarrow{\mathcal{G}} \Psi \xrightarrow{\mathcal{H}} \Xi \xrightarrow{\mathcal{K}} \Upsilon.$$

The equivalence between $(\mathcal{K} \circ \mathcal{H}) \circ \mathcal{G}$ and $\mathcal{K} \circ (\mathcal{H} \circ \mathcal{G})$ will be witnessed by the relation $\beta : \Sigma_{\mathcal{G}} \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{K}}) \twoheadrightarrow (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times \Sigma_{\mathcal{K}}$ generated by $(\sigma, (\tau, \mu)) \overset{\beta}{\sim} ((\sigma, \tau), \mu)$. Let $\sigma \in \Sigma_{\mathcal{G}}$, $\tau \in \Sigma_{\mathcal{H}}$, and $\mu \in \Sigma_{\mathcal{K}}$. Then $\mathrm{P}_{(\mathcal{K} \circ \mathcal{H}) \circ \mathcal{G})}(\sigma, (\tau, \mu)) = \mathrm{P}_{\mathcal{K} \circ (\mathcal{H} \circ \mathcal{G})}((\sigma, \tau), \mu)$ by associativity of composition in $\mathbf{Lens}_\mathcal{C}$. Let $[p, k] \in \mathbb{C}(\Phi, \Upsilon)$ be a context. Then

$$\begin{aligned}
&\mathrm{B}_{(\mathcal{K}\circ\mathcal{H})\circ\mathcal{G}}([p,k])(\sigma,(\tau,\mu)) \\
&= \mathrm{B}_\mathcal{G}\Big(\big[p, k \circ \mathrm{P}_\mathcal{K}(\mu) \circ \mathrm{P}_\mathcal{H}(\tau)\big]\Big)(\sigma) \\
&\quad \times \Big(\mathrm{B}_\mathcal{H}\Big(\big[\mathrm{P}_\mathcal{G}(\sigma) \circ p, k \circ \mathrm{P}_\mathcal{K}(\mu)\big]\Big)(\tau) \times \mathrm{B}_\mathcal{K}\Big(\big[\mathrm{P}_\mathcal{H}(\tau) \circ \mathrm{P}_\mathcal{G}(\sigma) \circ p, k\big]\Big)(\mu)\Big) \\
&\overset{\beta}{\sim} \Big(\mathrm{B}_\mathcal{G}\Big(\big[p, k \circ \mathrm{P}_\mathcal{K}(\mu) \circ \mathrm{P}_\mathcal{H}(\tau)\big]\Big)(\sigma) \times \mathrm{B}_\mathcal{H}\Big(\big[\mathrm{P}_\mathcal{G}(\sigma) \circ p, k \circ \mathrm{P}_\mathcal{K}(\mu)\big]\Big)(\tau)\Big) \\
&\quad \times \mathrm{B}_\mathcal{K}\Big(\big[\mathrm{P}_\mathcal{H}(\tau) \circ \mathrm{P}_\mathcal{G}(\sigma) \circ p, k\big]\Big)(\mu) \\
&= \mathrm{B}_{\mathcal{K}\circ(\mathcal{H}\circ\mathcal{G})}([p,k])((\sigma,\tau),\mu)
\end{aligned}$$

□

Theorem 6.9.0.2. *If* $\mathbf{Lens}_\mathcal{C}$ *exists, there exists a category* $\mathbf{Game}_\mathcal{C}$ *with object class* $\mathcal{C} \times \mathcal{C}$ *and equivalence classes of open games as morphisms.*

Proof. All that remains to be checked is that the identity computation defined in 6.7.0.3 is an identity morphism, and this follows from easy checks. □

6.10 The symmetric monoidal structure of open games

We now prove that $\otimes$ is functorial. The proof is a good demonstration of the utility of coend diagrams. In the commutative squares in the following lemma, the top path describes how local contexts are formed in, say, $(\mathcal{H} \otimes \mathcal{H}') \circ (\mathcal{G} \otimes \mathcal{G}')$ and the bottom path describes how local contexts are formed in $(\mathcal{H} \circ \mathcal{G}) \otimes (\mathcal{H}' \circ \mathcal{G}')$. That the squares commute follows by inspection of the appropriate coend diagrams.

Lemma 6.10.0.1. *Suppose we have coend lenses*

$$\Phi \xrightarrow{\quad l \quad} \Psi \xrightarrow{\quad m \quad} \Xi$$

$$\Phi' \xrightarrow{\quad l' \quad} \Psi' \xrightarrow{\quad m' \quad} \Xi'$$

The following diagrams commute:

1.

$$
\begin{array}{ccc}
\mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{L}(-, l' \circ m')} & \mathbb{C}(\Phi, \Xi) \\
{\scriptstyle \mathbb{C}(\Phi \otimes \Phi', m \otimes m')} \downarrow & & \downarrow {\scriptstyle \mathbb{C}(\Phi, m)} \\
\mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') & \xrightarrow[\mathcal{L}(-, l')]{} & \mathbb{C}(\Phi, \Psi)
\end{array}
$$

2.

$$
\begin{array}{ccc}
\mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{L}(-, l' \circ m')} & \mathbb{C}(\Phi, \Xi) \\
{\scriptstyle \mathbb{C}(l \otimes l', m \otimes m')} \downarrow & & \downarrow {\scriptstyle \mathbb{C}(l, \Xi)} \\
\mathbb{C}(\Psi \otimes \Psi', \Xi \otimes \Xi') & \xrightarrow[\mathcal{L}(-, m')]{} & \mathbb{C}(\Psi, \Xi)
\end{array}
$$

3.

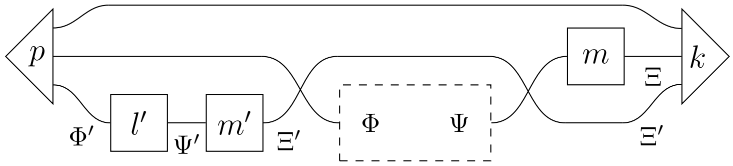
$$
\begin{array}{ccc}
\mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{R}(-, l \circ m)} & \mathbb{C}(\Phi', \Xi') \\
{\scriptstyle \mathbb{C}(\Phi \otimes \Phi', m \otimes m')} \downarrow & & \downarrow {\scriptstyle \mathbb{C}(\Phi, m)} \\
\mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') & \xrightarrow[\mathcal{R}(-, l)]{} & \mathbb{C}(\Phi', \Psi')
\end{array}
$$

4.

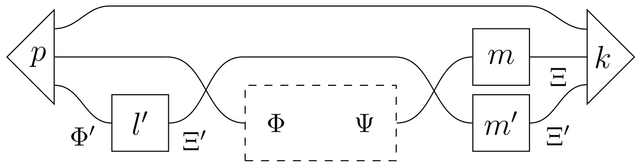
$$
\begin{array}{ccc}
\mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{R}(-, l \circ m)} & \mathbb{C}(\Phi', \Xi') \\
{\scriptstyle \mathbb{C}(l \otimes l', \Xi \otimes \Xi')} \downarrow & & \downarrow {\scriptstyle \mathbb{C}(l', \Xi')} \\
\mathbb{C}(\Psi \otimes \Psi', \Xi \otimes \Xi') & \xrightarrow[\mathcal{R}(-, m)]{} & \mathbb{C}(\Psi', \Xi')
\end{array}
$$

Proof. The four squares are given respectively by the following equalities of coend diagrams:

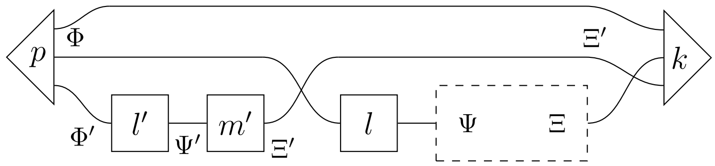
1.


p
m
k
Ξ
Φ′
l′
Ψ′
m′
Ξ′
Φ
Ψ
Ξ′

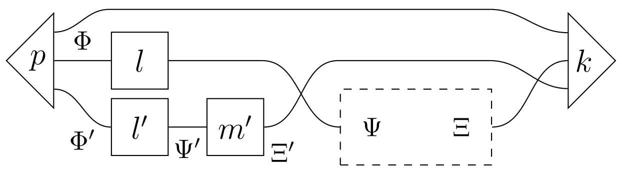
=


p
m
k
Ξ
Φ′
l′
Ξ′
Φ
Ψ
m′
Ξ′

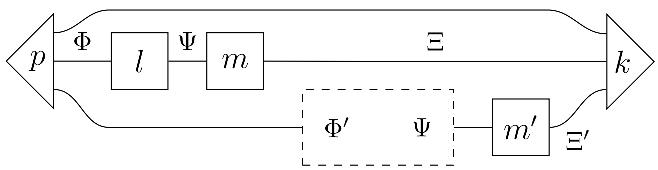
2.


p
Φ
Ξ′
k
Φ′
l′
Ψ′
m′
Ξ′
l
Ψ
Ξ

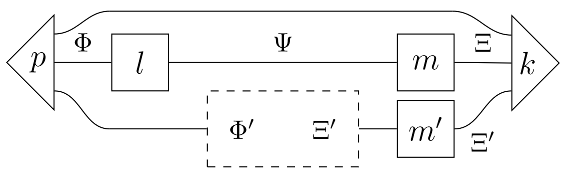
=


p
Φ
l
k
Φ′
l′
Ψ′
m′
Ξ′
Ψ
Ξ

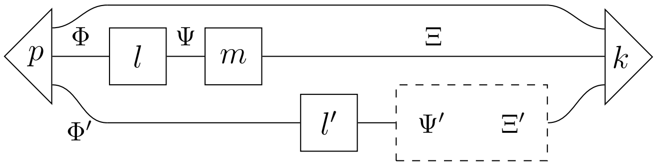
3.


p
Φ
l
Ψ
m
Ξ
k
Φ′
Ψ
m′
Ξ′

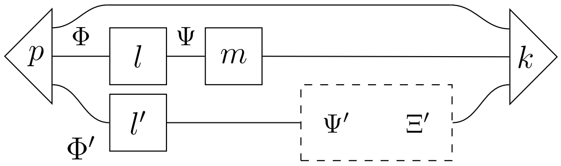
=


p
Φ
l
Ψ
m
Ξ
k
Φ′
Ξ′
m′
Ξ′

4.



$$=$$



□

Functoriality of the tensor in $\mathbf{Game}_{\mathcal{C}}$ then follows easily.

Corollary 6.10.0.2. *$\otimes : \mathbf{Game}_{\mathcal{C}} \times \mathbf{Game}_{\mathcal{C}} \to \mathbf{Game}_{\mathcal{C}}$ is a functor.*

Proof. Suppose we have open games

$$\Phi \xrightarrow{\mathcal{G}} \Psi \xrightarrow{\mathcal{H}} \Xi$$

$$\Phi' \xrightarrow{\mathcal{G}'} \Psi' \xrightarrow{\mathcal{H}'} \Xi'$$

Note that $\Sigma_{(\mathcal{H}\circ\mathcal{G})\otimes(\mathcal{H}'\circ\mathcal{G}')} = (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times (\Sigma_{\mathcal{G}'} \times \Sigma_{\mathcal{H}'})$ and $\Sigma_{(\mathcal{H}\otimes\mathcal{H}')\circ(\mathcal{G}\otimes\mathcal{G}')} = (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{G}'}) \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{H}'})$. The relation $\beta : (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times (\Sigma_{\mathcal{G}'} \times \Sigma_{\mathcal{H}'}) \twoheadrightarrow (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{G}'}) \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{H}'})$ witnessing the equivalence between $(\mathcal{H} \circ \mathcal{G}) \otimes (\mathcal{H}' \circ \mathcal{G}')$ and $(\mathcal{H} \otimes \mathcal{H}') \circ (\mathcal{G} \otimes \mathcal{G}')$ is generated by $((\sigma,\tau),(\sigma',\tau') \overset{\beta}{\sim} ((\sigma,\sigma'),(\tau,\tau'))$. $\mathbf{Lens}_{\mathcal{C}}$ is symmetric monoidal and, hence,

$$\mathrm{P}_{(\mathcal{H}\circ\mathcal{G})\otimes(\mathcal{H}'\otimes\mathcal{G}')}((\sigma,\tau),(\sigma',\tau')) = \mathrm{P}_{(\mathcal{H}\otimes\mathcal{H}')\circ(\mathcal{G}\otimes\mathcal{G})}((\sigma,\sigma'),(\tau,\tau')).$$

Using 6.10.0.1,

$$
\begin{aligned}
& \mathrm{B}_{(\mathcal{H}\circ\mathcal{G})\otimes(\mathcal{H}'\circ\mathcal{G}')}(c)\big((\sigma,\tau),(\sigma',\tau')\big) \\
= \; & \Big(\mathrm{B}_{\mathcal{G}}\Big(\mathbb{C}(\Phi,\mathrm{P}_{\mathcal{H}}(\tau))\circ\mathcal{L}(-,\mathrm{P}_{\mathcal{H}'}(\tau')\circ\mathrm{P}_{\mathcal{G}'}(\sigma'))(c)\Big)(\sigma) \\
& \times \mathrm{B}_{\mathcal{H}}\Big(\mathbb{C}(\mathrm{P}_{\mathcal{G}}(\sigma),\Xi)\circ\mathcal{L}(-,\mathrm{P}_{\mathcal{H}'}(\tau')\circ\mathrm{P}_{\mathcal{G}'}(\sigma'))(c)\Big)(\tau)\Big) \\
& \times \Big(\mathrm{B}_{\mathcal{G}'}\Big(\mathbb{C}(\Phi',\mathrm{P}_{\mathcal{H}'}(\tau'))\circ\mathcal{R}(-,\mathrm{P}_{\mathcal{H}}(\tau)\circ\mathrm{P}_{\mathcal{H}}(\sigma))(c)\Big)(\sigma') \\
& \times \mathrm{B}_{\mathcal{H}'}\Big(\mathbb{C}(\mathrm{P}_{\mathcal{G}'}(\sigma'),\Xi')\circ\mathcal{R}(i,\mathrm{P}_{\mathcal{H}}(\tau)\circ\mathrm{P}_{\mathcal{H}}(\sigma))(c)\Big)(\tau')\Big) \\
\overset{\beta}{\sim} \; & \Big(\mathrm{B}_{\mathcal{G}}\Big(\mathcal{L}(-,\mathrm{P}_{\mathcal{G}'}(\sigma'))\circ\mathbb{C}(\Phi\otimes\Phi',\mathrm{P}_{\mathcal{H}}(\tau)\otimes\mathrm{P}_{\mathcal{H}'}(\tau'))(c)\Big)(\sigma) \\
& \times \mathrm{B}_{\mathcal{G}'}\Big(\mathcal{R}(-,\mathrm{P}_{\mathcal{G}}(\sigma))\circ\mathbb{C}(\Phi\otimes\Phi',\mathrm{P}_{\mathcal{H}}(\tau)\otimes\mathrm{P}_{\mathcal{H}'}(\tau'))(c)\Big)(\sigma')\Big) \\
& \times \Big(\mathrm{B}_{\mathcal{H}}\Big(\mathcal{L}(-,\mathrm{P}_{\mathcal{H}'}(\tau'))\circ\mathbb{C}(\mathrm{P}_{\mathcal{G}}(\sigma)\otimes\mathrm{P}_{\mathcal{G}'}(\sigma'),\Xi\otimes\Xi')(c)\Big)(\tau) \\
& \times \mathrm{B}_{\mathcal{H}'}\Big(\mathcal{R}(-,\mathrm{P}_{\mathcal{H}}(\tau))\circ\mathbb{C}(\mathrm{P}_{\mathcal{G}}(\sigma)\otimes\mathrm{P}_{\mathcal{G}'}(\sigma'),\Xi\otimes\Xi')(c)\Big)(\tau')\Big) \\
= \; & \mathrm{B}_{(\mathcal{H}\otimes\mathcal{H}')\circ(\mathcal{G}\otimes\mathcal{G}')}(c)\big((\sigma,\sigma'),(\tau,\tau')\big)
\end{aligned}
$$

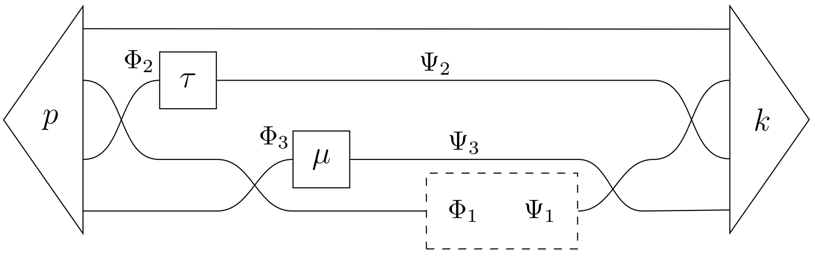
□

Definition 6.10.0.3. The structural isomorphisms in $\mathbf{Game}_{\mathcal{C}}$ are given by

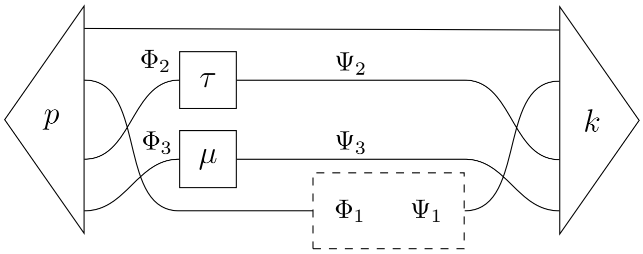
$$
\begin{aligned}
\alpha_{(X,A),(Y,B),(Z,C)} &= \langle \alpha_{X,Y,Z}, \alpha^{-1}_{A,B,C} \rangle \\
\rho_{(X,A)} &= \langle \rho_X, \rho^{-1}_A \rangle \\
\lambda_{(X,A)} &= \langle \lambda_X, \lambda^{-1}_A \rangle \\
s_{(X,A),(Y,B)} &= \langle s_{X,Y}, s_{B,Y} \rangle
\end{aligned}
$$

Lemma 6.10.0.4. *The structural isomorphisms are natural in* $\mathbf{Game}_{\mathcal{C}}$*.*

Proof. We show that the associator is natural. Naturality of the other structural maps follow by similar arguments. Let $\mathcal{G}_i : \Phi_i \to \Psi_i$ for $i \in \{1,2,3\}$. Note that $\Sigma_{\alpha\circ(\mathcal{G}_1\otimes(\mathcal{G}_2\otimes\mathcal{G}_3))} = \big(\Sigma_{\mathcal{G}_1} \times (\Sigma_{\mathcal{G}_2} \times \Sigma_{\mathcal{G}_3})\big) \times \{\alpha\}$ and $\Sigma_{((\mathcal{G}_1\otimes\mathcal{G}_2)\otimes\mathcal{G}_3)\circ\alpha} = \{\alpha\} \times \big((\Sigma_{\mathcal{G}_1} \times \Sigma_{\mathcal{G}_2}) \times \Sigma_{\mathcal{G}_3}\big)$. The equivalence between $\alpha \circ \big(\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3)\big)$ and $\big((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3\big) \circ \alpha$ will be witness by the relation generated by $\big((\sigma,(\tau,\mu)),\alpha\big) \overset{\beta}{\sim} \big(\alpha,((\sigma,\tau),\mu)\big)$. Let $\sigma \in \Sigma_{\mathcal{G}_1}, \tau \in \Sigma_{\mathcal{G}_2}, \mu \in \Sigma_{\mathcal{G}_3}$, and $[p,k] \in \mathbb{C}\big((\Phi_1 \otimes (\Phi_2 \otimes \Phi_3)),((\Psi_1 \otimes \Psi_2) \otimes \Psi_3)\big)$. We note that the local context for $\mathcal{G}_1$ given this data is the same for both games. The local context of $\mathcal{G}_1$ is given by



in $\alpha \circ \big(\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3)\big)$ and by



in $\big((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3\big) \circ \alpha$. This two morphisms are evidently equal. Similar diagrams demonstrate that the local contexts for $\mathcal{G}_2$ and $\mathcal{G}_3$ are the same in both games also. □

Theorem 6.10.0.5. $\mathbf{Game}_{\mathcal{C}}$ *is symmetric monoidal.*

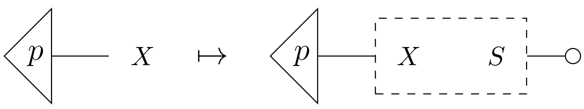
Proof. All that remains to be shown is that the MacLane pentagon and triangle axioms are satisfied, but this follows easily as the underlying category $\mathcal{C}$ is symmetric monoidal. □

6.11 Nice categories of open games

In this section we show how the notion of 'cohistory' collapses when the monoidal unit I of the underlying monoidal category $\mathcal{C}$ is terminal. With cohistories gone, we will see that $\mathbf{Game}_{\mathcal{C}}$ has a very natural game theoretic interpretation.

Lemma 6.11.0.1 ([Ril18])**.** *If the monoidal unit of* $\mathcal{C}$ *is terminal, then* $\mathbf{Lens}_{\mathcal{C}}(I,(X,S)) \cong \mathcal{C}(I,X)$. □

The isomorphism $i : \mathcal{C}(I,X) \to \mathbf{Lens}_{\mathcal{C}}(I,(X,S))$ is explicitly given by $p \mapsto [p, !_s]$. In a diagram,



Corollary 6.11.0.2. *If the monoidal unit of $\mathcal{C}$ is terminal, then*

$$\mathbb{C}((X,S),(Y,R)) \cong \mathbf{Lens}_{\mathcal{C}}((I,R),(X,Y))$$

Proof. Using 6.11.0.1 and the fact that $\mathcal{C}(Y,R) \cong \mathbf{Lens}_{\mathcal{C}}\big(I,(Y,R)\big)$ (6.6.0.1),

$$\begin{aligned}
\mathbb{C}((X,S),(Y,R)) &= \int^{(A,B)\in\mathbf{Lens}_{\mathcal{C}}} \mathbf{Lens}_{\mathcal{C}}(I,(A\otimes X, B\otimes S)) \times \mathbf{Lens}_{\mathcal{C}}((A\otimes Y, B\otimes R), I) \\
&\cong \int^{A:\mathcal{C}} \mathcal{C}(I, A\otimes X) \times \mathcal{C}(A\otimes Y, R) \\
&= \mathbf{Lens}_{\mathcal{C}}((I,R),(X,Y))
\end{aligned}$$

□

Unpacking definitions, the isomorphism $i : \mathbf{Lens}_{\mathcal{C}}((I,R),(X,Y)) \to \mathbb{C}((X,S),(Y,R))$ is explicitly given by

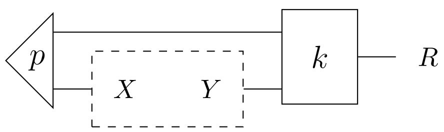
$$[p : I \to A\otimes X, k : A\otimes Y \to R] \mapsto \Big[[p, !_S], [k, \mathrm{id}_R]\Big].$$

In the case where the monoidal unit of $\mathcal{C}$ is terminal, the type of best response for an open game $\mathcal{G} : (X,S) \to (Y,R)$ is equivalently

$$\mathrm{B}_{\mathcal{G}} : \mathbf{Lens}_{\mathcal{C}}((I,R),(X,Y)) \to \mathrm{Rel}(\Sigma_{\mathcal{G}}).$$

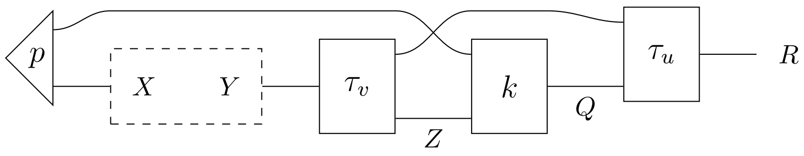
We have seen that expressing contexts as states in the double lens category is a good level of abstraction for categories of open games, allowing for elegant proofs with pretty diagrams. From a game theoretic perspective, however, it will make more sense to express contexts as equivalence classes $[p,k,\Theta] : \mathbf{Lens}_{\mathcal{C}}((I,R),(X,Y))$. This is because a state $p : \mathcal{C}(I, \Theta\otimes X)$ is easily seen to correspond to a *history* for an open game and the function $k : \Theta\otimes Y \to R$ acts like an *outcome function*. In this way, we can specify a context for an open game in much the same way as we did for concrete open games in chapter 5.

The coend diagram

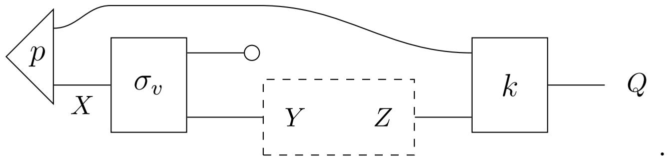


of a context $[p,k] \in \mathbf{Lens}_{\mathcal{C}}((I,R),(X,Y))$ neatly illustrates that a context is a game state with a ‘hole’ in it. If we think of a game $\mathcal{G} : (X,S) \to (Y,R)$ as a player in a larger game, then p corresponds to the things that have happened in the game before $\mathcal{G}$ gets to act; k corresponds to what will happen in the game after $\mathcal{G}$ acts; and the gap in the diagram corresponds to the part of the game where $\mathcal{G}$ gets to influence the outcome. Alternatively, a context is that which becomes a game once $\mathcal{G}$ has decided which strategy to play, whereby playing that strategy will fill in the gap in the context.

Given open games $\mathcal{G} : (X, S) \to (Y, R)$, $\mathcal{H} : (Y, R) \to (Z, Q)$, a context $[p, k] \in \mathbf{Lens}_{\mathcal{C}}((I, R), (X, Z))$, and strategies $\sigma \in \Sigma_{\mathcal{G}}, \tau \in \Sigma_{\mathcal{H}}$, the local context for $\mathcal{G}$ in $\mathcal{H} \circ \mathcal{G}$ is given by

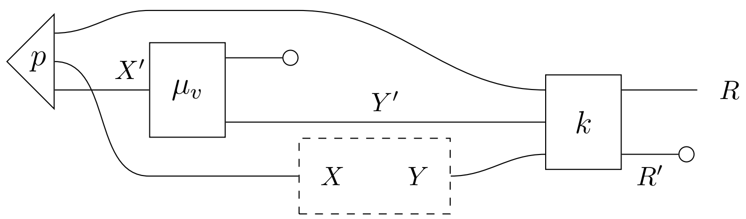


and the local context for $\mathcal{H}$ is given by

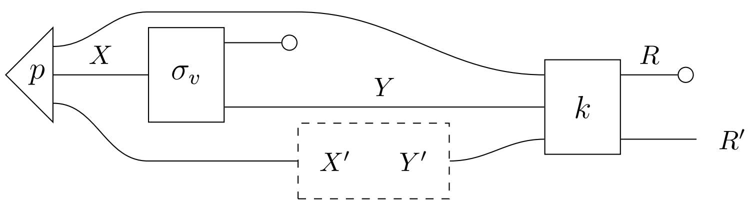


.

Given another open game $\mathcal{K} : (X', S') \to (Y', R')$, a context $[p, k] \in \mathbf{Lens}_{\mathcal{C}}((I, R \otimes R'), (X \otimes X', Y \otimes Y'))$, and a strategy $\mu \in \Sigma_{\mathcal{H}}$, the local contexts for $\mathcal{G}$ and $\mathcal{H}$ in $\mathcal{G} \otimes \mathcal{H}$ are given by



and



respectively.

Chapter 7

Bayesian open games

In this chapter we descend from the abstraction of the previous chapter to analyse open games over a particular category, namely the Kleisli category of the distribution monad $\mathbf{Kl}(D)$. We will see that $\mathbf{Kl}(D)$ allows us to model various types of *Bayesian games* as open games, vastly expanding the expressive capabilities of the open game formalism.

7.1 Chapter Overview

In 7.2 we introduce some more well-known results about monoidal categories; in 7.3 we define the category of *sets and random functions*, the underlying category of the category of Bayesian open games; 7.4 presents a brief introduction to classical Bayesian games; 7.5 formally introduces *Bayesian open games*; 7.6 defines *Bayesian agents*, atomic Bayesian open games with a best response function that takes Bayesian updating into account; 7.7 shows how the Bayesian games of classical game theory can be modelled using Bayesian open games.

7.2 Commutative monads

Recall that a monad T over a monoidal category $\mathcal{C}$ is *strong* if it comes with a *strength* natural transformation $t_{A,B} : A \otimes TB \to T(A \otimes B)$ satisfying various coherence conditions.

We have the following result guaranteeing the existence of a large class of coend lens categories. We refer the reader to [Ril18] for a much more in-depth discussion of the following result, and many more examples of when lens categories exist.

Theorem 7.2.0.1 ([Ril18]). *If T is a strong monad over a category $\mathcal{C}$, then* $\mathbf{Lens}_{\mathbf{Kl}(T)}\big(A, B\big)$ *is a set for all $A, B \in \mathcal{C}$.*[1]. □

Definition 7.2.0.2 (Commutative monad). Let T be a strong monad with strength t over a monoidal category $\mathcal{C}$. Define the *costrength* natural transformation $t'_{A,B}$: $TA \otimes B \to T(A \otimes B)$ to be the composite

$$TA \otimes B \xrightarrow{s_{TA,B}} B \otimes TA \xrightarrow{t_{B,A}} T(B \otimes A) \xrightarrow{T(s_{B,A})} T(A \otimes B).$$

T is *commutative* if the diagram

$$\begin{array}{ccccc} TA \otimes TB & \xrightarrow{t_{TA,B}} & T(TA \otimes B) & \xrightarrow{T(t'_{A,B})} & T^2(A \otimes B) \\ \downarrow{\scriptstyle t'_{A,TB}} & & & & \downarrow{\scriptstyle \mu} \\ T(A \otimes TB) & \xrightarrow[T(t_{A,B})]{} & T^2(A \otimes B) & \xrightarrow[\mu]{} & T(A \otimes B) \end{array}$$

commutes for all objects A and B in $\mathcal{C}$.

If a monad is commutative then we get that its Kleisli category is symmetric monoidal with the monoidal tensor $\otimes$ (on objects) and unit being the same as in the underlying category $\mathcal{C}$.

Lemma 7.2.0.3 ([PR97]). *If T is a commutative monad over a symmetric monoidal category $\mathcal{C}$, then* $\mathbf{Kl}(T)$ *is symmetric monoidal.* □

Commutative monads over **Set** also come with canonical copy/delete comonoid structures for every object. Copying $c_X : X \to T(X \times X)$ is given by

$$X \xrightarrow{\Delta} X \times X \xrightarrow{\eta} T(X \times X)$$

and deleting $d_X : X \to I$ is given by

$$X \xrightarrow{!} \{\star\} \xrightarrow{\eta} T(\{\star\}) \, .$$

From this comonoid structure we obtain canonical projections

$$X \otimes Y \xrightarrow{\mathrm{id} \otimes d} X \otimes I \xrightarrow{\rho} X$$

[1] In [Ril18], lenses over a Kleisli category are called *effectful optics*.

and

$$X \otimes Y \xrightarrow{d \otimes \mathrm{id}} I \otimes Y \xrightarrow{\lambda} Y \ .$$

Crucially, it is *not* guaranteed that the monoidal tensor of $\mathbf{Kl}(T)$ is cartesian. In particular, the Kleisli category of the distribution monad defined in the next section is not cartesian.

7.3 The category of sets and random functions

We now turn to the category of interest for this chapter.

The *finitary distribution monad* $D : \mathbf{Set} \to \mathbf{Set}$ maps a set X to the set of finitely supported probability distributions on X. We make use of *finitary* distributions to keep things simpler, and because all the games we are interested in modelling have finite sets of actions in any case.

Definition 7.3.0.1 (Finitary distribution monad)**.** Define $D : \mathbf{Set} \to \mathbf{Set}$ by

$$D(X) = \left\{ \alpha : X \to [0,1] \;\middle|\; \mathrm{supp}(X) < \aleph_0, \sum_{x \in \mathrm{supp}(\alpha)} \alpha(x) = 1 \right\}$$

where $\mathrm{supp}(\alpha)$ is $\left\{ x \in X \;\middle|\; \alpha(x) \neq 0 \right\}$, the *support* of α. D acts on morphisms by

$$D(f : X \to Y)(\alpha : D(X))(y) = \sum_{f(x)=y} \alpha(x).$$

The monad structure of D is given as follows. The unit is given by

$$\eta_X(x) = \delta_x$$

where

$$\delta_x(x') = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}$$

and the extension $f^\dagger : D(X) \to D(Y)$ of $f : X \to D(Y)$ is

$$f^\dagger(\alpha) = \sum_{x \in \mathrm{supp}(\alpha)} f(x)(y)$$

Lemma 7.3.0.2. *D is a commutative monad.* □

Corollary 7.3.0.3. $\mathbf{Kl}(D)$ *is symmetric monoidal with canonical copy/delete comonoids and projection maps.*

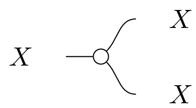
The monoidal tensor in $\mathbf{Kl}(D)$ is *not* cartesian.

Definition 7.3.0.4 (Copy/delete comonoid for $\mathbf{Kl}(D)$)**.** The copying map in $\mathbf{Kl}(D)$ is given explicitly by $c_X : X \to D(X \times X)$ where

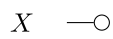
$$c(x)(x_1, x_2) = \begin{cases} 1 & \text{if } x = x_1 = x_2 \\ 0 & \text{otherwise.} \end{cases}$$

The monoidal unit of $\mathbf{Kl}(D)$ is terminal, and hence the deleting map $d_X : X \to D(\{\star\})$ must be given by $d(x)(\star) = 1$. As this map is unique, we refer to it as !.

Whenever we are working in $\mathbf{Kl}(D)$ we denote c_X and $!_X$ by



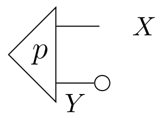
and



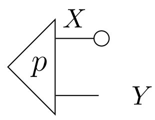
respectively.

The canonical projections in $\mathbf{Kl}(D)$ correspond to taking marginals.

Definition 7.3.0.5 (Marginals)**.** Let $p : D(X \times Y)$. The *left marginal* $p_X : D(X)$ is given by $p_X(x) = \sum_{y \in \mathrm{supp}(p(x,-))} p(x, y)$. The *right marginal* $p_Y : D(Y)$ is given similarly by $p_Y(y) = \sum_{x \in \mathrm{supp}(p(-,y))} p(x, y)$. As diagrams, these are given by



and



respectively.

An important operation on probability distributions is *Bayesian updating* where an agent has some prior distribution (initial belief), makes an observation, and then updates their prior to a new posterior distribution.

Definition 7.3.0.6 (Update operator)**.** Let X and A be sets. We think of A as a type of values which an agent has a probabilistic belief about, and X as a type of values that will be observed by an agent. Define the *update operator* $\mathcal{U}_A : D(A \times X) \to (X \to D(A))$ by

$$\mathcal{U}_A(p)(x)(\vartheta) = \frac{p(\vartheta, x)}{\sum_{(\vartheta', x) \in \mathrm{supp}(p)} p(\vartheta', x)}.$$

We also write $p(\vartheta | x)$ for $\mathcal{U}_A(p)(x)(\vartheta)$.

Lemma 7.3.0.7. *The update operator is natural in A. That is, the following diagram commutes for any $f : A_1 \to A_2$:*

$$\begin{array}{ccc}
D(A_1 \times X) & \xrightarrow{\mathcal{U}_{A_1}} & X \to D(A_1) \\
{\scriptstyle D(f \times X)} \downarrow & & \downarrow {\scriptstyle \circ D(f)} \\
D(A_2 \times X) & \xrightarrow[\mathcal{U}_{A_2}]{} & X \to D(A_2)
\end{array}$$

Proof. Let $p \in D(A_1 \times X), x \in X, \vartheta_1 \in A_1$, and $\vartheta_2 \in A_2$. The top of the square is given by

$$\begin{aligned}
\Big(D(f) \circ \mathcal{U}_{A_1}\Big)(p)(x)(\vartheta_2) &= \sum_{f(\vartheta_1)=\vartheta_2} \mathcal{U}_{A_1}(p)(x)(\vartheta_1) \\
&= \frac{\sum_{f(\vartheta_1)=\vartheta_2} p(\vartheta_1, x)}{\sum_{\vartheta'_1 \in \mathrm{supp}(p(-,x))} p(\vartheta'_1, x)}. \qquad (\star)
\end{aligned}$$

The bottom of the square is given by

$$\begin{aligned}
\mathcal{U}_{A_2}\Big(D(f \times X)(p)\Big)(x)(\vartheta_2) &= \frac{D(f \times X)(p)(\vartheta_2, x)}{\sum_{\vartheta'_2} D(f \times X)(p)(\vartheta'_2, x)} \\
&= \frac{\sum_{f(\vartheta_1)=\vartheta_2} p(\vartheta_1, x)}{\sum_{\vartheta'_2} \sum_{f(\vartheta'_1)=\vartheta'_2} p(\vartheta'_1, x)}. \qquad (\star\star)
\end{aligned}$$

The result follows, noting that the denominators of $(\star)$ and $(\star\star)$ are equal.

□

7.4 Bayesian games

In classical game theory, *Bayesian games* are games in which players have probabilistic beliefs about the other players in the game. We first state the standard definition, then discuss what it means.

Definition 7.4.0.1 (Bayesian game)**.** A *Bayesian game* is a tuple (N, A, Θ, p, u) where

1. $N = \{1, \cdots, n\}$ is a finite set of *players*,
2. $A = A_1 \times \cdots \times A_n$ where A_i is a finite set of *actions* available to player i,

3. $\Theta = \Theta_1 \times \cdots \times \Theta_n$ where Θ_i is the *flavour space* of player i,

4. $p : \Theta \to [0, 1]$ is a *common prior* over flavours;

5. $u = (u_1, \cdots, u_n)$ where $u_i : A \times \Theta \to \mathbb{R}$ is the *outcome function* for player i.

A *pure strategy* for player i is a function $\sigma_i : \Theta_i \to A_i$. A *mixed strategy* for player i is a function $s_i : \Theta_i \to D(A_i)$.

The sets N and A are self-explanatory. N specifies the number of players in the game and A_i specifies the set of actions that player i can choose from. Each player i is associated with a set Θ_i of *flavours* [2]. The flavour space of a player might be something like {good, evil}, {smart, dumb}, or {conservative, risk-taker, chaotic}. The common prior $p : \Theta \to [0, 1]$ describes the players' probabilistic beliefs about which flavours other players have been assigned. This information is strategically relevant as players' outcomes are allowed to depend on their flavour and, consequently, which actions are utility-maximising also depends on players' flavours. In a playthrough of a Bayesian game, we imagine that each player is assigned a flavour sampled from the common prior; each player observes their own flavour and, using this information, updates their belief about the distribution of the other players' flavours; then each player chooses an action in an attempt to maximise their outcome function.

Example 7.4.0.2 (Education game)**.** The *education game* is a game of two players: an *employer* e and an *applicant* a. The flavour space of the employer is the one-element set $\{\star\}$. We suppose that the applicant is talented with probability $\frac{1}{10}$ and untalented with probability $\frac{9}{10}$ and that their flavour space is $\{t, \sim t\}$ where t and $\sim t$ correspond to being talented and untalented respectively. The applicant makes the choice either to attend university or to not attend university, represented by the action set $\{u, \sim u\}$. We suppose that attending university incurs a greater cost if the applicant is untalented. This cost is described by the function cost : $\{t, \sim t\} \times \{u, \sim u\}$ given by

$$\text{cost}(t, u) = 1$$
$$\text{cost}(\sim t, u) = 3$$
$$\text{cost}(t, \sim u) = \text{cost}(\sim t, \sim u) = 0$$

The employer decides whether to the pay the applicant a high wage or a low wage represented by the action set $\{2, 1\}$. The objective of the employer is to offer a high

[2] In the game theory literature, the flavour space is known as the *type space*, but we have chosen the different term to prevent confusion with the types of category theory

wage only if the applicant is talented. We can represent this game as a Bayesian game where

- $N = \{e, a\}$;
- $A_e = \{2, l\}$, $A_a = \{u, \sim u\}$;
- $\Theta_e = \{\star\}$, $\Theta_a = \{t, \sim t\}$;
- $p : \{\star\} \times \{t, \sim t\} \to [0, 1]$ is given by

$$p(\star, x) = \begin{cases} \frac{1}{10} & \text{if } x = t \\ \frac{9}{10} & \text{otherwise.} \end{cases}$$

- The outcome function $u_a : \{t, \sim t\} \times \{u, \sim u\} \times \{2, 1\} \to \mathbb{R}$ for a is given by $u_a(x, y, z) = z - \text{cost}(x, y)$ and the outcome function $u_e : \{t, \sim t\} \times \{u, \sim u\} \times \{2, 1\} \to \mathbb{R}$ is given by, for any $y \in \{u, \sim u\}$,

$$u_e(t, y, 2) = u_e(\sim t, y, 1) = 1$$
$$u_e(t, y, 1) = u_e(\sim t, y, 2) = 0.$$

7.4.1 Epistemics in Bayesian games

In order to define a sensible solution concept for Bayesian games, we must first consider that there exist multiple epistemic states of players in a Bayesian game. The expected outcome of a choice can change as players learn more about the flavours in a game. We are particularly interested in the following epistemic states.

1. *ex post* - the flavours of all the players are common knowledge;
2. *ex interim* - players know only their own flavour;
3. *ex ante* - no flavours are known whatsoever.

Each of these epistemic scenarios has a distinct associated expected outcome function.

Definition 7.4.1.1 (Ex post utility)**.** The *ex post* expected utility of player i in a Bayesian game (N, A, Θ, p, u) given a mixed strategy profile $s = (s_1, \cdots, s_n)$ and a specification of flavours $\vartheta = (\vartheta_1, \cdots, \vartheta_n)$ is given by

$$\mathbf{ep}_i(s, \vartheta) = \sum_{a \in A} \left(\prod_{j \in N} s_j(\vartheta_j)(a_j) \right) \big(u_i(a, \vartheta) \big).$$

Ex interim utility can be defined as a weighted sum of ex post utility functions.

Definition 7.4.1.2 (Ex interim utility)**.** The *ex interim* expected utility of player i with flavour $\vartheta_i \in \Theta_i$ given a mixed strategy profile $s = (s_1, \cdots, s_n)$ is given by

$$\mathbf{ei}_i(s, \vartheta_i) = \sum_{\vartheta_{-i} \in \Theta_{-i}} p(\vartheta_{-i} | \vartheta_i) \mathbf{ep}_i(s, \vartheta_i, \vartheta_{-i})$$

Similarly, ex ante utility can be defined as a weighted sum of ex interim utility or, consequently, of ex post utility

Definition 7.4.1.3 (Ex ante utility)**.** The *ex ante* expected utility of player i given a mixed strategy profile $(s_1, \cdots, s_n)$ is given by

$$\begin{aligned} \mathbf{ea}_i(s) &= \sum_{\vartheta \in \Theta} p(\vartheta) \mathbf{ep}_i(s, \vartheta) \\ &= \sum_{\vartheta_i \in \Theta_i} p(\vartheta_i) \mathbf{ei}_i(s, \vartheta_i) \end{aligned}$$

We can then define *Bayesian best response functions* that pick out the most profitable unilateral deviations from a mixed strategy profile s. We define this relative to the ex ante utility function.

Definition 7.4.1.4 (Bayesian best response)**.** Player i's best responses to the mixed strategy profile s_{-i} are given by

$$\mathrm{B}_i(s_{-i}) = \arg\max_{s'_i \in S_i} \mathbf{ea}_i(s'_i, s_{-i})$$

Bayesian best response then yields a Bayesian solution concept by considering the strategy profiles which are fixed points.

Definition 7.4.1.5 (Bayesian Nash equilibrium)**.** A strategy profile s is a *Bayesian Nash equilibrium* if $\forall i \in N$,

$$s_i \in \mathrm{B}_i(s_{-i}).$$

7.5 Bayesian open games

Definition 7.5.0.1 (Bayesian open game)**.** A *Bayesian open game* is a morphism in $\mathbf{Game}_{\mathbf{Kl}(D)}$. Explicitly, a Bayesian open game $\mathcal{G} : (X, S) \to (Y, R)$ is given by

1. A set of strategies Σ,

2. A play function

$$\mathrm{P} : \Sigma \to \int^{A:\mathcal{C}} \big(X \to D(A \times Y)\big) \times \big((A \times R) \to D(S)\big),$$

3. A best response function

$$\mathrm{B} : \mathbf{Lens}_{\mathbf{Kl}(D)}((I, R), (X, Y)) \to \mathrm{Rel}(\Sigma)$$

We refer to atoms in the category of Bayesian open games as *Bayesian atoms*.

We unpack the definition of the play function to emphasize that, when we wish to actually specify a Bayesian open game, it is usually easier to specify $\mathrm{P}(\sigma)$ as the equivalence class of a pair of morphisms.

7.6 Bayesian agents

We will now define *Bayesian agents* which, as with concrete open games, have constant best response functions. Bayesian agents capture the notion of rational agents that

1. Have a correct prior about the various types in a game;

2. Update this prior based on an observation;

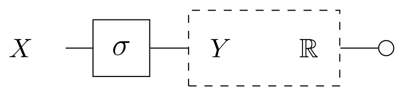
3. Attempt to maximise their expected utility given their updated prior.

Definition 7.6.0.1 (Bayesian agent)**.** Let X, Y be sets. The *Bayesian agent* $\mathcal{A}_{(X,Y)}$: $(X, I) \to (Y, \mathbb{R})$ is the Bayesian atom given by

1. $\Sigma_{\mathcal{A}} = X \to D(Y)$,

2.

$$\mathrm{P}\sigma = [\sigma, !_{\mathbb{R}}] =$$



3. The preference function $\varepsilon : \mathbf{Lens}_{\mathbf{Kl}(D)}((I, \mathbb{R}), (X, Y)) \to \mathcal{P}(X \to D(Y))$ is given by

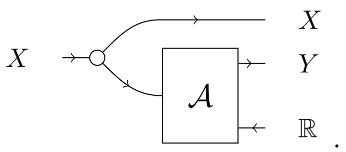
$$\varepsilon_{\mathcal{A}}([p, k]) = \Big\{ \sigma : X \to D(Y) \;\Big|\; \forall x \in \mathrm{supp}(p), \\ \sigma(x) \in \arg\max_{\alpha \in D(Y)} \Big(\mathbb{E}\Big[k(\mathcal{U}_{\Theta}(p)(x), \alpha) \Big] \Big) \Big\}$$

Lemma 7.6.0.2. *The preference function of a Bayesian agent is well-defined. That is, it is independent of the choice of representative of the coend equivalence relation.*

Proof. This result follows from the fact that Bayesian updating is natural in the bound type of a coend lens (7.3.0.7). □

In the next definition we formalise the idea that a player in a game might be assigned a (game theoretic) type on which their utility function depends. We can do this simply using a Bayesian agent and a copying computation.

Definition 7.6.0.3. Let $\mathcal{A}_{(X,Y)} : (X, I) \to (Y, \mathbb{R})$ be a Bayesian agent. Define $\mathcal{A}^{\Delta}_{(X,Y)} : (X, I) \to (X \times Y, \mathbb{R})$ to be the Bayesian open game

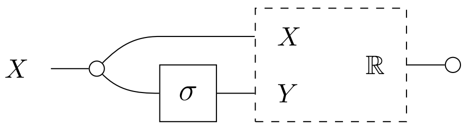


.

Lemma 7.6.0.4. $\mathcal{A}^{\Delta}_{(X,Y)}$ *is explicitly given, up to isomorphism, by*

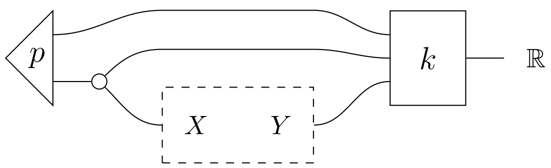
1. $\Sigma_{\mathcal{A}_{(X,Y)^\Delta}} = X \to D(Y)$;
2. $\mathrm{P}_{\mathcal{A}^{\Delta}_{(X,Y)}} : \Sigma_{\mathcal{A}_{(X,Y)^\Delta}} \to \mathbf{Lens}_{\mathbf{Kl}(D)}\big((X, I), (X \times Y, \mathbb{R})\big)$ *is given by*

$$\mathrm{P}_{\mathcal{A}^{\Delta}_{(X,Y)}}(\sigma) \quad = $$



;

3. *Let* $[p, k] \in \mathbf{Lens}_{\mathbf{Kl}(D)}\big((I, \mathbb{R}), (X, X \times Y)\big)$. *Best response is given, up to isomorphism, by* $\mathrm{B}_{\mathcal{A}_{(X,Y)^\Delta}}([p, k])(\sigma) = \mathrm{B}_{\mathcal{A}_{(X,Y)}}([p, k]')$ *where* $[p, k]'$ *is the context given by*



.

Proof. This result follows from definition chasing, noting that $\mathcal{A}_{(X,Y)}$ is the only component with non-trivial strategy profile set. □

Lemma 7.6.0.5. *Let* $\mathcal{A}_{(X_i,Y_i)}$ *be Bayesian agents for* $i \in \{1, \cdots, n\}$. *Then* $\mathcal{A}^{\Delta^n} = \bigotimes_{i=1}^{n} \mathcal{A}^{\Delta}_{(X_i,Y_i)}$ *is explicitly given as follows.*

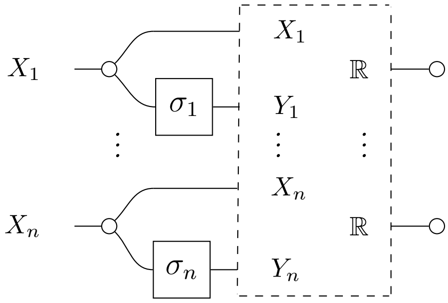
1. $\Sigma_{\mathcal{A}^{\Delta^n}} = \prod_{i=1}^n (X_i \to D(Y_i))$;

2. *The play function*

$$\mathrm{P}_{\mathcal{A}^{\Delta^n}} : \Sigma_{\mathcal{A}^{\Delta^n}} \to \mathbf{Lens}_{\mathbf{Kl}(D)}\Big(\big(\prod_{i=1}^n X_i, I\big), \big(\prod_{i=1}^n (X_i \times Y_i), \mathbb{R}^n\big)\Big)$$

is given by

$$\mathrm{P}_{\mathcal{A}^n_\circ}(\sigma_1, \cdots, \sigma_n) \quad =$$



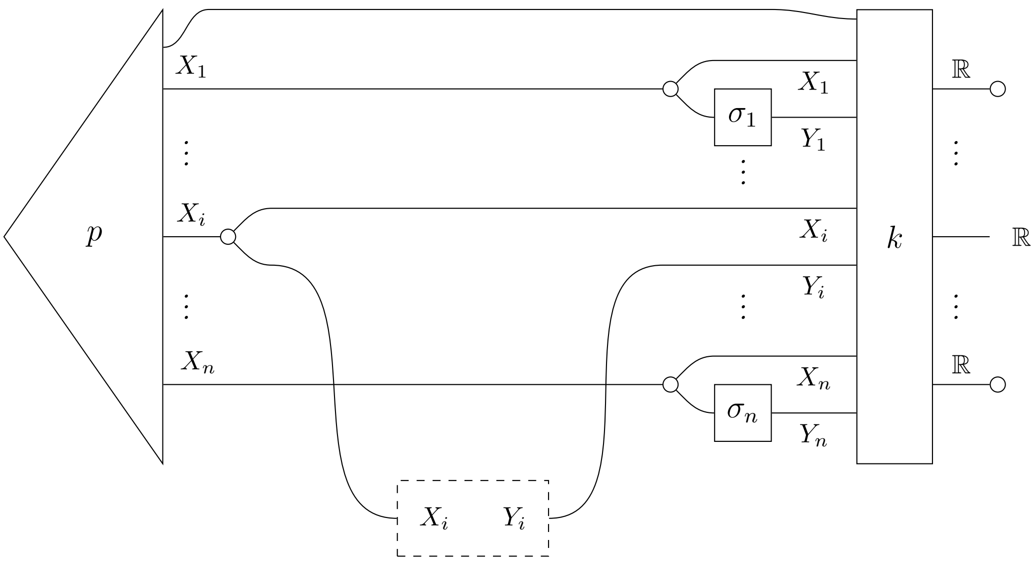
3. *Let* $[p, k] \in \mathbf{Lens}_{\mathbf{Kl}(D)}\Big((I, \mathbb{R}^n), (\prod_i X_i, \prod_i X_i \times Y_i)\Big)$. *Best response*

$$\mathrm{B}_{\mathcal{A}^{\Delta^n}}([p, k])(\sigma_1, \cdots, \sigma_n)$$

is given, up to isomorphism, by

$$\prod_{i=1}^n \Bigg\{ \sigma_i : X_i \to D(Y_i) \;\Bigg|\; \forall x_i \in \operatorname{supp}(p^{\sigma_{-i}}),$$
$$\sigma_i(x_i) \in \arg\max_{\alpha_i \in D(Y_i)} \Big(\mathbb{E}\big[k^{\sigma_{-i}}\big(\mathcal{U}_{A_{-i}}(p^{\sigma_{-i}})(x_i), \alpha_i\big]\Big) \Bigg\}$$

where $[p^{\sigma_{-i}}, k^{\sigma_{-i}}]$ *is the context given by*

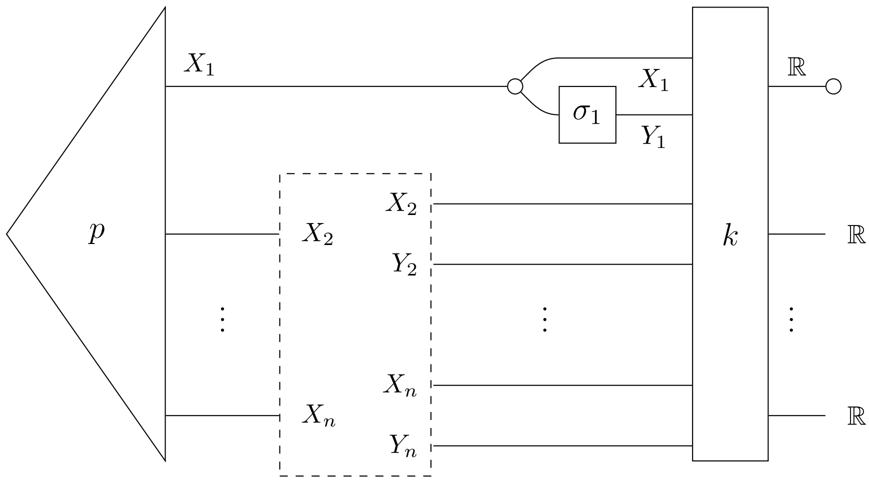


.

Proof. 1 and 2 follow easily from definitions. As for 3, we need to prove that the local context for each $\mathcal{A}_{(X_i,Y_i)}$ is $[p^{\sigma_{-i}}, k^{\sigma_{-i}}]$. Note that the previous lemma 7.6.0.4 serves as the base case ($n = 1$) for an induction argument. The result then follows easily by considering that

$$\mathrm{B}_{\mathcal{A}^{\Delta n}}([p,k])(\sigma_1, \cdots, \sigma_n) = \mathrm{B}_{\mathcal{A}^{\Delta}_{(X_1,Y_1)}}([p,k]_1)(\sigma_1) \times \mathrm{B}_{\bigotimes_{i=2}^n \mathcal{A}^{\Delta}_{(X_i,Y_i)}}([p,k]_{-1})(\sigma_{-1})$$

and applying the inductive hypothesis, where $[p,k]_{-1}$ is the context



□

7.7 Bayesian games as Bayesian open games

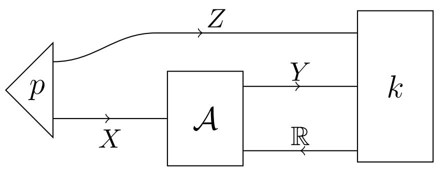
Given a Bayesian game $\mathcal{G} = (N, A, \Theta, p, u)$, we can then model $\mathcal{G}$ with the Bayesian open game $\bigotimes_{i=1}^n \mathcal{A}^{\Delta}_{(\Theta_i,A_i)}$ and applying the best response function to the context $[p,k]$ where k is the outcome function given by

$$k\big((\vartheta_1, a_1), \cdots, (\vartheta_n, a_n)\big) = \delta_{(u_1(\vartheta_1,a_1),\cdots,u_n(\vartheta_n,a_n))}.$$

7.7.1 Decisions under risk

In this section we introduce another type of situation involving a Bayesian agent that can be modeled using Bayesian open games.

A *decision problem under risk* is a decision problem for which one can sensibly assign probabilities to possible outcomes. A good example is roulette. When making a bet in roulette, you can calculate the likelihood of success and also your expected return on any bet. Decision problems under risk are generally represented by Bayesian open games constructed from computations and precisely one Bayesian agent. A simple subclass of decision problems under risk is represented by Bayesian open games of form

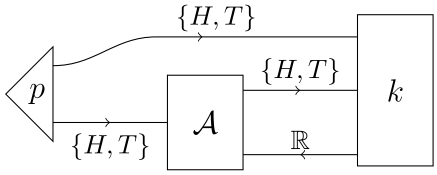


in which an agent $\mathcal{A}$ attempts to maximise their outcome which is, in part, dependent on the type Z which $\mathcal{A}$ does not observe.

We now give a fully worked out example of a Bayesian open game in which an agent has a prior, makes an observation, updates their prior as a consequence of that observation, and then makes a prediction based on their updated prior.

Example 7.7.1.1 (Biased coins)**.** Suppose we give an agent $\mathcal{A}$ a biased coin which lands on one side 75% of the time and the other side 25% of the time. It is not known which side the coin is biased towards, but it is known that it is equally likely to be biased towards heads as towards tails. $\mathcal{A}$ flips the coin whilst another identical coin (i.e. another coin biased the same way) is flipped in secret. $\mathcal{A}$ observes her coin flip and is then asked to predict which side up the secret coin landed. If she is correct she receives an outcome of 1 with probability 1. If she is wrong she receives an outcome of 0 with probability 1. The optimal strategy for $\mathcal{A}$ is to guess that the coin flipped in secret will land the same way up as the coin she flipped. If, for instance, $\mathcal{A}$'s coin comes up heads, then there is a 75% chance that both coins are biased towards heads. Consequently the coin flipped in secret is more likely to show heads. A symmetric argument applied if $\mathcal{A}$'s coin shows tails.

We can represent this game using the open game



where

$$\begin{aligned}
p : D(\{H,T\}^2) &= \frac{1}{2}\left(\frac{9}{16}(T,T) + \frac{1}{16}(H,H) + \frac{3}{16}(T,H) + \frac{3}{16}(H,T)\right) \\
&\quad + \frac{1}{2}\left(\frac{9}{16}(H,H) + \frac{1}{16}(T,T) + \frac{3}{16}(H,T) + \frac{3}{16}(T,H)\right) \\
&= \frac{5}{16}(H,H) + \frac{5}{16}(T,T) + \frac{3}{16}(T,H) + \frac{3}{16}(H,T)
\end{aligned}$$

and

$$
\begin{aligned}
k : \{H,T\}^2 &\to D\mathbb{R} \\
(x,y) &\mapsto \begin{cases} \delta_1 & \text{if } x = y \\ \delta_0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

Explicitly, the game is given by $\mathcal{G} := u \circ (\mathcal{A} \otimes \mathrm{id}_{\{H,T\}}) \circ p$. Note that $\Sigma_{\mathcal{G}} \cong \Sigma_{\mathcal{A}} = \{H,T\} \to D(\{H,T\})$. Also note that there is precisely one context for $\mathcal{G}$ and, moreover, as the best response functions for $k, id_{\{H,T\}}$, and p are total, the best response function for $\mathcal{G}$ is isomorphic to the constant relation

$$
\mathrm{B}_{\mathcal{A}}[p,k] = \Big\{ \sigma : \{H,T\} \to D(\{H,T\}) \;\Big|\; \forall x \in \{H,T\}, \sigma(x) \in \arg\max_{\alpha \in D(\{H,T\})} \mathbb{E}[k(\mathcal{U}_{\{H,T\}}(p)(x), \alpha)] \Big\}.
$$

The updated prior $\mathcal{U}_{\{H,T\}}(p)(H)(H)$ is given by

$$
\begin{aligned}
\mathcal{U}_{\{H,T\}}(p)(H)(H) &= \frac{p(H,H)}{\sum_{(\vartheta,H)\in \mathrm{supp}(p)} p(\vartheta,H)} \\
&= \frac{5}{8}
\end{aligned}
$$

and hence $\mathcal{U}_{\{H,T\}}(p)(H)(T) = \frac{3}{8}$. Similarly, $\mathcal{U}_{\{H,T\}}(p)(T) = \frac{5}{8}T + \frac{3}{8}H$. It follows that

$$
\begin{aligned}
\arg\max_{\alpha}\Big(\mathbb{E}[k^{\dagger}(\mathcal{U}_{\{H,T\}}(p)(H),\alpha)]\Big) &= \{\delta_H\} \\
\arg\max_{\alpha}\Big(\mathbb{E}[k^{\dagger}(\mathcal{U}_{\{H,T\}}(p)(T),\alpha)]\Big) &= \{\delta_T\}.
\end{aligned}
$$

Hence $\mathrm{B}_{\mathcal{A}}[p,k]$ is a singleton set containing the strategy σ where $\sigma(H) = \delta_H$ and $\sigma(T) = \delta_T$, as expected.

We now sketch some examples of games that can be represented as Bayesian open games.

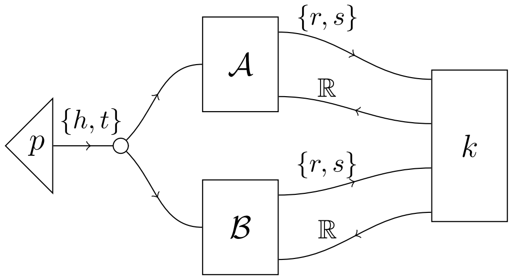
Example 7.7.1.2 (Correlated equilibria)**.** *The battle of the sexes* is a two-player normal form game in which two players $\mathcal{A}$ and $\mathcal{B}$ attempt to coordinate with each other. $\mathcal{A}$ and $\mathcal{B}$ agreed to meet in the evening, but neither can recall whether they agreed to go to restaurant r or to restaurant s. $\mathcal{A}$ has a slight preference for r and $\mathcal{B}$ has a slight preference for s, but failing to coordinate leads to the worst outcome for both $\mathcal{A}$ and $\mathcal{B}$. We model this with the outcome function $u : \{r,s\}^2 \to \mathbb{R}^2$ given by

$$
\begin{aligned}
u(r,r) &= (3,2) \\
u(r,s) = u(s,r) &= (0,0) \\
u(s,s) &= (2,3)
\end{aligned}
$$

where the left and right projections are the outcomes for $\mathcal{A}$ and $\mathcal{B}$ respectively. This game has two deterministic Nash equilibria where both players choose the same restaurant, but these are both 'unfair' in the sense that one player receives a higher payoff than the other. There is also a 'fair' Nash equilibria in mixed strategies, where $\mathcal{A}$ and $\mathcal{B}$ choose their preferred restaurant with probability $\frac{3}{5}$, but in this equilibrium the players miscoordinate a large proportion of the time.

Suppose now that there is a flip of a fair coin that both $\mathcal{A}$ and $\mathcal{B}$ observe before choosing which restaurant to go to. Consider a kind of strategy profile where both players choose r if the coin lands on heads, and both choose s if the coin lands on tails. In this situation neither player has incentive to deviate from the option suggested by the coin flip, both players receive the same expected outcome, and there is no possibility of miscoordination. This strategy profile is said to be a *correlated equilibrium.* Correlated equilibria were introduced as a generalisation of Nash equilibria by Aumann in the paper [Aum87].

We can represent this game with the following Bayesian open game



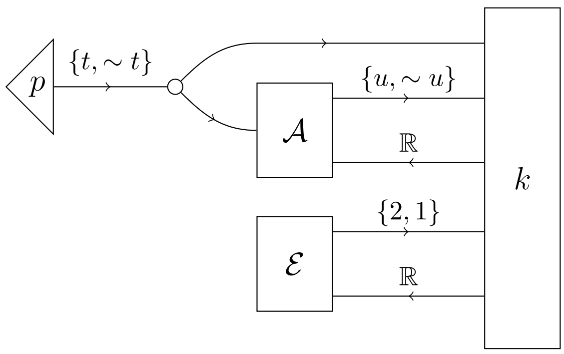
where $p : D(\{h,t\})$ is the fair coin flip; $\mathcal{A}$ and $\mathcal{B}$ are Bayesian agents (although there is no non-trivial Bayesian updating in this game); and $k : \{r,s\}^2 \to \mathbb{R}^2$ is given by

$$k(r,r) = \delta_{(3,2)}$$
$$k(r,s) = k(s,r) = \delta_{(0,0)}$$
$$k(s,s) = \delta_{(2,3)}$$

The fixed points of the best response function for this Bayesian game are then the correlated equilibria.

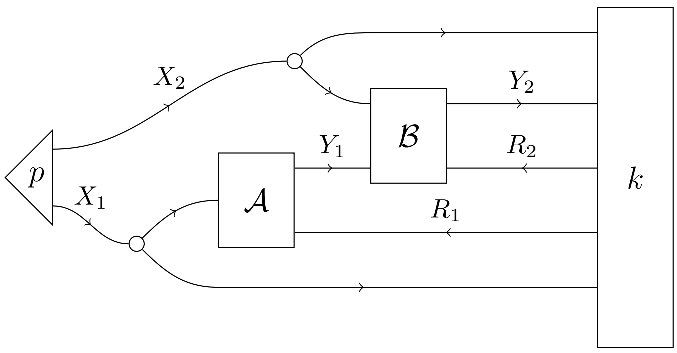
Example 7.7.1.3 (Signalling games)**.** We now return to the education game introduced in 7.4.0.2 and show how it can be described as a Bayesian open game. Recall that the education game is between two players, an *employer* $\mathcal{E}$ and an *applicant* $\mathcal{A}$. $\mathcal{E}$ has to decide whether to offer $\mathcal{A}$ a high wage or a low wage (say $\mathcal{E}$ chooses wages from the set of choices $\{2,1\}$); and $\mathcal{A}$ has to decide whether to attend university or not (represented by $\{u, \sim u\}$). $\mathcal{A}$ is talented with probably $\frac{1}{10}$, and attending university incurs a greater cost if they are not talented. $\mathcal{E}$ wishes to pay $\mathcal{A}$ a high wage if and only if $\mathcal{A}$ is talented, but cannot directly observe whether $\mathcal{A}$ is talented or not. They can only infer $\mathcal{A}$'s talent from their (correct) prior about the likelihood of $\mathcal{A}$'s being talented and from $\mathcal{A}$'s decision where to attend university.

We can represent the education game as the following Bayesian open game.



where p describes the probably that $\mathcal{A}$ is talented; $\mathcal{A}$ and $\mathcal{E}$ are Bayesian agents; and $k : \{t, \sim t\} \times \{u, \sim u\} \times \{2, 1\} \to D(\mathbb{R}^2)$ is given by $u \circ \eta_{\mathbb{R}^2}$ where $u = (u_a, u_e)$ as defined in 7.4.0.2.

Example 7.7.1.4 (Bayesian game with sequential play)**.** We can also describe situations in which there is both sequential play and Bayesian updating. Consider the following Bayesian open game where $\mathcal{A}$ and $\mathcal{B}$ are both Bayesian agents.



This describes a situation similar to a Bayesian game. Both players are assigned a flavour with some probability (the types X_1 and X_2 are the flavour classes for $\mathcal{A}$ and $\mathcal{B}$ respectively). The players observe their flavour and update their prior (that is, p) before making a decision, but in this sequential case player $\mathcal{B}$ also gets to observe $\mathcal{A}$'s choice of move.

Chapter 8

Conclusion and further work

There is much work still to be done in the study of open games, and here we give a list of some of the important work that is still outstanding.

8.1 Incomplete information

Whilst Bayesian open games greatly increase the expressive power of the open games formalism, there are still classes of games to elude it.

1. *Instances where there is an epistemic hierarchy.* Bayesian open games accommodate instances where an agent has a correct prior, makes an observation, and updates that prior. They do *not* account for more complex situations in which, for instance, there are two agents $\mathcal{A}$ and $\mathcal{B}$ and $\mathcal{A}$ has beliefs about $\mathcal{B}$'s beliefs about $\mathcal{A}$'s beliefs about ... and so on. In classical game theory, such epistemic hierarchies are modelled using *Harsanyi type spaces* (introduced in the paper [Har67]). As far as the author is aware, no attempt has yet been made to incorporate Harsanyi type spaces into open games.

2. *Games with no common prior.* Due to the way sequential and tensor composition of open games are defined, it is not currently known how to model games where different players have different priors at the start of the game.

3. *Games with false priors.* Similar to the previous point. Open games currently do not capture games where players are mistaken in their beliefs.

8.2 Subgame perfection

The fixed points of the best response function of an open game correspond to Nash equilibria. Subgame perfect Nash equilibria are, arguably, more relevant for games involving sequential play. For a comprehensive compositional account of game theory, there should exist a category of open games where the fixed points of best response functions correspond to subgame perfect equilibria. This problem is hard because open games have both sequential and simultaneous play. Intuitively it is not clear what a subgame perfect Nash equilibrium would look like for games with simultaneous play. On a technical level, many of the best candidates for such a category turn out to only be premonoidal categories for which the tensor is not a true bifunctor. In particular, this means that the order in which a game is composed from its subcomponents matters. Moreover, without a symmetric monoidal category, the diagrammatic calculi of monoidal categories cannot be leveraged.

Some progress has been made towards a category of open games which computes subgame perfect equilibria in the paper [GKLF18], but this progress is currently restricted to a small subclass of open games.

8.3 Compact closure

The category of open games has *counit* games, but no corresponding *unit* games. A question remains as to whether there exists a meaningful quotient of the category of open games which is compact closed.

8.4 Higher categorical structure

That one has to take a quotient at all to form the category of open games suggests a higher categorical structure for open games. That is, instead of taking a quotient, we should consider morphisms between open games. The higher categorical structure of open games has been studied in the papers [Hed18b] and [GKLF18], but there is much work left to be done. In particular, the author believes that the appropriate morphisms between open games are the *simulations* used in this thesis, but this idea remains to be studied in any detail. An account of the higher categorical structure of open games with simulations as morphisms between games would involve investigating the allegorical structure of open games (the most influential introduction to allegories is the book [FS90]).

8.5 Extensive form

Notable by its absence in Part II of this thesis is any mention of extensive form games. A comprehensive translation of classical game theory to open game theory must include a translation of extensive form games, yet open games are not well-suited to describing games in extensive form. It is certainly possible, but an elegant solution is yet to be found.

8.6 Concluding thoughts

Open games are an exciting area of active research, and there is clearly much interesting and difficult work left to be done. In the author's opinion, the greatest difficulty that lies ahead is convincing the current practitioners of game theory (mainly economists) of the benefits of adopting the open games formalism. This is certainly some way off, but the readily-comprehensible string diagrams of monoidal category theory offer some hope that open games may become more widely adopted, allowing the finer details of the underlying category theory to be hidden from view.

The work in this thesis on Bayesian open games, together with the work on iterated games in [GKLF18], contribute greatly to the comprehensiveness of open games. Games of incomplete information and extensive form games are perhaps the remaining important classes of games left to be modelled using open games.

The achievements of the second part of this thesis can be summarized as follows.

1. Various strategic situations involving Bayesian agents can now be represented as open games. These include Bayesian games, decisions under risk, signalling games, and Bayesian games with sequential play.

2. The above kinds of game are represented as distinct mathematical objects in the classical game theory literature. Bayesian open games provide a unifying framework, showing that various types of game can be seen as instances of open games.

3. A compositional account has now been provided for games with Bayesian agents.

The future for open games looks promising, and the author is excited to see how the field will develop in the future.

Bibliography

[Aum87] Robert J Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica: Journal of the Econometric Society*, pages 1–18, 1987.

[Awo10] Steve Awodey. *Category theory*. Oxford University Press, 2010.

[Bae06] John Baez. Quantum quandaries: a category-theoretic perspective. *The structural foundations of quantum gravity*, pages 240–265, 2006.

[BBC+93] Charles H Bennett, Gilles Brassard, Claude Crépeau, Richard Jozsa, Asher Peres, and William K Wootters. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. *Physical review letters*, 70(13):1895, 1993.

[BCG+17] Joe Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden, and Robin Piedeleu. Interacting conceptual spaces i: Grammatical composition of concepts. *arXiv preprint arXiv:1703.08314*, 2017.

[BCS07] Alexandru Baltag, Bob Coecke, and Mehrnoosh Sadrzadeh. Epistemic actions as resources. *Journal of Logic and Computation*, 17(3):555–585, 2007.

[BP17] John C Baez and Blake S Pollard. A compositional framework for reaction networks. *Reviews in Mathematical Physics*, 29(09):1750028, 2017.

[BS81] François Bancilhon and Nicolas Spyratos. Update semantics of relational views. *ACM Transactions on Database Systems (TODS)*, 6(4):557–575, 1981.

[BW85] M. Barr and C. Wells. *Toposes, Triples and Theories*. Springer, 1985.

[CK17] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes*. Cambridge University Press, 2017.

[CS12] Bob Coecke and Robert W Spekkens. Picturing classical and quantum Bayesian inference. *Synthese*, 186(3):651–696, 2012.

[EO10a] Martin Escardó and Paulo Oliva. Selection functions, bar recursion and backward induction. *Mathematical structures in computer science*, 20(2):127–168, 2010.

[EO10b] Martin Escardó and Paulo Oliva. Sequential games and optimal strategies. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20100471. The Royal Society, 2010.

[EO14] Martín Hötzel Escardó and Paulo Oliva. The Herbrand functional interpretation of the double negation shift. *CoRR*, abs/1410.4353, 2014.

[EO15] Martín Escardó and Paulo Oliva. Bar recursion and products of selection functions. *The Journal of Symbolic Logic*, 80(1):1–28, 2015.

[Esc04] Martín Escardó. Synthetic topology: of data types and classical spaces. *Electronic Notes in Theoretical Computer Science*, 87:21–156, 2004.

[Esc07] Martin Escardo. Infinite sets that admit fast exhaustive search. In *Logic in Computer Science, 2007. LICS 2007. 22nd Annual IEEE Symposium on*, pages 443–452. IEEE, 2007.

[Esc08] Martin Escardo. Exhaustible sets in higher-type computation. *arXiv preprint arXiv:0808.0441*, 2008.

[FS90] P.J. Freyd and A. Scedrov. *Categories, Allegories.* ISSN. Elsevier Science, 1990.

[GKLF18] Neil Ghani, Clemens Kupke, Alasdair Lambert, and Fredrik Nordvall Forsberg. A compositional treatment of iterated open games. *Theoretical Computer Science*, 741:48 – 57, 2018. An Observant Mind : Essays Dedicated to Don Sannella on the Occasion of his 60th Birthday.

[Har67] John C Harsanyi. Games with incomplete information played by "Bayesian" players, i–iii part i. the basic model. *Management science*, 14(3):159–182, 1967.

[Hed14] Jules Hedges. Monad transformers for backtracking search. *arXiv preprint arXiv:1406.2058*, 2014.

[Hed15] Jules Hedges. The selection monad as a CPS transformation. *arXiv preprint arXiv:1503.06061*, 2015.

[Hed16] Jules Hedges. *Towards compositional game theory*. PhD thesis, Queen Mary University of London, 2016.

[Hed17] Jules Hedges. Coherence for lenses and open games. *arXiv preprint arXiv:1704.02230*, 2017.

[Hed18a] Jules Hedges. Morphisms of open games. *Electronic Notes in Theoretical Computer Science*, 341:151 – 177, 2018. Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).

[Hed18b] Jules Hedges. Morphisms of open games. *Electronic Notes in Theoretical Computer Science*, 341:151 – 177, 2018. Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).

[HOS+17a] Jules Hedges, Paulo Oliva, Evguenia Shprits, Viktor Winschel, and Philipp Zahn. Higher-order decision theory. In *International Conference on Algorithmic DecisionTheory*, pages 241–254. Springer, 2017.

[HOS+17b] Jules Hedges, Paulo Oliva, Evguenia Shprits, Viktor Winschel, and Philipp Zahn. Selection equilibria of higher-order games. In *International Symposium on Practical Aspects of Declarative Languages*, pages 136–151. Springer, 2017.

[Koc70] Anders Kock. Monads on symmetric monoidal closed categories. *Archiv der Mathematik*, 21(1):1–10, 1970.

[KSPC13] Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, Stephen Pulman, and Bob Coecke. Reasoning about meaning in natural language with compact closed categories and frobenius algebras. *Logic and Algebraic Structures in Quantum Computing*, page 199, 2013.

[LB08] Kevin Leyton-Brown. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. Morgan and Claypool, 2008.

[Lei16] Tom Leinster. Basic category theory. *arXiv preprint arXiv:1612.09375*, 2016.

[Lor15] Fosco Loregian. This is the (co) end, my only (co) friend. *arXiv preprint arXiv:1501.02503*, 2015.

[ML71] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag New York, 1971.

[Mog91] Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93(1):55–92, 1991.

[Mos79] Andrzej Mostowski. On a generalization of quantifiers. *Studies in Logic and the Foundations of Mathematics*, 93:311–335, 1979.

[MSZ13] Michael Maschler, Eilon Solan, and Shmuel Zamir. *Game Theory.* Cambridge University Press, 2013.

[OP15] Paulo Oliva and Thomas Powell. A constructive interpretation of ramsey's theorem via the product of selection functions. *Mathematical structures in computer science*, 25(8):1755–1778, 2015.

[PR97] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Mathematical Structures in Computer Science*, 7(5):453–468, 1997.

[Ril18] Mitchell Riley. Categories of optics. *arXiv:1809.00738v2*, 2018.

[S$^+$67] Norman E Steenrod et al. A convenient category of topological spaces. *Michigan Math. J*, 14(2):133–152, 1967.

[Sel10] Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010.

[Sel12] Peter Selinger. Finite dimensional hilbert spaces are complete for dagger compact closed categories. *arXiv preprint arXiv:1207.6972*, 2012.

[VNM44] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior.* Princeton University Press, 1944.