

# Bayesian open games

Joe Bolt, Jules Hedges<sup>1,2,3</sup>, and Philipp Zahn<sup>2,3</sup>

<sup>1</sup>Department of Computer and Information Sciences, University of Strathclyde, 26 Richmond Street, Glasgow, G11XH, U.K.

<sup>2</sup>20squares, <https://20squares.xyz>

<sup>3</sup>CyberCat Institute, <https://cybercat.institute>

This paper generalises the treatment of compositional game theory as introduced by Ghani et al. in 2018, where games are modelled as morphisms of a symmetric monoidal category. From an economic modelling perspective, the notion of a game in the work by Ghani et al. is not expressive enough for many applications. This includes stochastic environments, stochastic choices by players, as well as incomplete information regarding the game being played. The current paper addresses these three issues all at once.

## 1 Introduction

In [13] the first compositional treatment of economic game theory was introduced. Following the literature on categorical open systems [7], *open games* are modelled as morphisms of a symmetric monoidal category.

A distinctive and non-obvious feature of this approach is that the Nash equilibrium condition [30], one of the central concepts in classical game theory to analyse rational behaviour of agents (cf. [12, Chapter 1.2] and [33, Chapter 2]), is itself compositional.

While an important first step, the treatment in [13] has two severe limitations:

1. Games are deterministic and as a consequence, there are no chance elements in the games and players have to choose deterministically.
2. Players have complete information about all relevant data of the game such as payoffs, number of players etc.

Many interesting strategic situations feature chance elements. Poker is one example – already discussed in the ground-breaking work of von Neumann and Morgenstern [31]. In an economic context, the environment also often is non-deterministic. Two competing companies face uncertain demand, exchange rates, lawsuits etc.

More subtle but also important is that players may need and may want to randomise their actions. There are well known situations like Matching Pennies (see, for instance, [12, p. 16]) where playing deterministically means being ‘beaten’ all the time. Conceptually, from a game theory perspective, this means that there are games where equilibria do not exist when players are limited to deterministic strategies (known as ‘pure strategies’) whereas they do exist when players can choose stochastically (known as ‘mixed strategies’).

Lastly, it is a crude approximation to assume that players have complete information. Examples abound. A used-car dealer knows how good the car is that he is trying to sell to you. You may not know. Banks sitting on toxic assets know how little value they actually have. The government trying to buy these assets in order to save the financial system from collapse may not know. An agent bidding in an auction may not know how many other bidders he competes with. In most situations incomplete information is the norm and not the exception.

---

Jules Hedges: [jules.hedges@strath.ac.uk](mailto:jules.hedges@strath.ac.uk)

Philipp Zahn: [philipp@20squares.xyz](mailto:philipp@20squares.xyz)

The above limitations restrict the applicability of compositional game theory to economic phenomena. And it restricts its usefulness for economists. After all, classical game theory already deals with these complications.

In this paper we provide a generalisation of open games which solves the three problems above *in one go*. We adapt the core definition of compositional game theory such that the environment can be stochastic and players can also choose in a non-deterministic fashion. Doing so, we also introduce a way to deal with incomplete information. Essentially we are lifting the same ‘trick’, which has been introduced in classical game theory to deal with games of incomplete information by John Harsanyi [17–19], to compositional game theory.

Harsanyi argued that instead of dealing with games of incomplete information directly, which poses formidable conceptual problems, we can transform such games into games of imperfect information by introducing the notion of (game) types. Players have access to probability distributions characterising these games as well as partial access to this information. For instance, in an auction a player may know how much he values the good to be auctioned. However, he may not know how other agents value the good. Assuming that players update information according to Bayes’ rule and adapting the equilibrium notion of Nash, to what is called *Bayesian Nash equilibrium*, game theorists can work with such interactions no differently to how they deal with chance elements in Poker. Thus by transforming the problem, Harsanyi essentially opened the path to using tools that were more or less already introduced by von Neumann and Morgenstern [31].

We are applying the same strategy. By introducing stochastic environments and adapting the equilibrium notion from Nash to Bayesian Nash we show that our compositional framework captures exactly Bayesian Games and thus allows to deal with stochastic environments as well as with situations of incomplete information. This distinguishes our work also from [15] which addresses the issue of deterministic players in isolation.

## 1.1 Technical introduction

Contrary at least to our own initial beliefs, addressing these issues requires some significant adaptations of open games as defined in [13].

The recent understanding of open games has been based on *lenses*, which consist of a pair of functions  $X \rightarrow Y$  and  $X \times R \rightarrow S$  packaged into a single morphism  $(X, S) \rightarrow (Y, R)$  of a category. Here, the function  $X \rightarrow Y$  is the *play function*, which plays out a given strategy by taking an initial state to a final state of the open game. The function  $X \times R \rightarrow S$ , known as the *coplay function* or *coutility function*, is more subtle: It ‘backpropagates’ payoffs into the past, given an initial state. This operation is ‘counterfactual’, and the composition of lenses (which is not trivial to define, nor is obvious to see is associative) intertwines ordinary forward and counterfactual (or ‘teleological’) information flow.

An open game can then be viewed as a family of lenses indexed by a set of strategy profiles, together with another component describing which strategy profiles are Nash equilibria in a given *context*. A context for an open game consists of an initial state ( $X$ ) and a function from final states to payoffs ( $Y \rightarrow R$ ). Contexts turn out also to be intimately connected to lenses, and indeed this was the initial hint that viewing open games in terms of lenses is a deep idea rather than a coincidence.

To someone trained in thinking about processes with side effects, it is entirely natural to begin by inserting a (finite support) probability monad  $D$ , and take the components of the lenses to be Kleisli morphisms  $X \rightarrow D(Y)$  and  $X \times R \rightarrow D(S)$ , or equivalently to use lenses over the category of sets and (finite support) stochastic functions. This allows the strategies of an open game to describe probabilistic behaviours, which are known as *behavioural strategies* in game theory. Unfortunately this doesn’t work: In order to prove that lenses form a category (i.e. are associative and unital) it is necessary that the forwards maps  $X \rightarrow Y$  are homomorphisms of copying, and in the category of stochastic processes this characterises those processes that are actually deterministic.

Fortunately this problem has already been solved in the theory of lenses, although the solution is far from obvious. We use the *existential lenses* or *coend lenses* as developed by Riley [36]. This means we replace the pair of functions  $X \rightarrow D(Y)$  and  $X \times R \rightarrow D(S)$  with three things: A choice of set  $A$ , a function  $X \rightarrow D(A \times Y)$  and a function  $A \times R \rightarrow D(S)$ . Moreover a certain equivalence relation needs to be imposed, and this is precisely given by the following *coend* [27] (one of the

universal constructions of category theory):

$$\int^A (X \rightarrow D(A \times Y)) \times (A \times R \rightarrow D(S)).$$

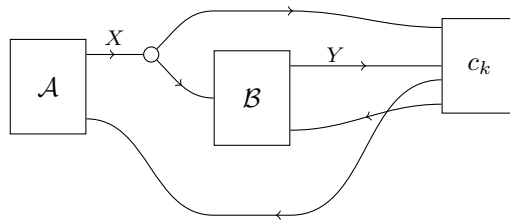
The proofs in this paper make heavy use of a diagrammatic language for existential lenses developed in [36].

The second question is what should be considered a context of a Bayesian open game, i.e. a replacement for the pair  $X \times (Y \rightarrow R)$ . There is an existing characterisation of these contexts in terms of deterministic lenses, namely as a ‘state’ lens  $(1, 1) \rightarrow (X, S)$  and a ‘costate’ lens  $(Y, R) \rightarrow (1, 1)$ . However this turns out to be a red herring: in Section 3.4 we show that generalising from this causes the category of open games to fail to be monoidal in an unexpected way.

It turns out that the appropriate notion of context consists of three things: a set  $\Theta$  of *unobservable* states, a joint distribution on  $\Theta \times X$  (i.e. an element of  $D(\Theta \times X)$ ) and a function  $\Theta \times Y \rightarrow D(R)$ . Again we need to impose a certain equivalence relation, which again turns out to be precisely a coend. Remarkably this is equivalent to a state in the category of *double lenses*, i.e. lenses over the category of lenses. This brings an unexpected theoretical unity to Bayesian open games, and means that the graphical language of [36] can be used throughout.

## 2 Concrete open games

We begin with a self-contained introduction to deterministic open games. In essence, we will introduce the necessary machinery so that we can represent simple classical games with diagrams as depicted below.



This diagram displays an interaction between two agents, A and B. Player A moves first; player B observes the choice by A and then moves afterwards. Both moves are consumed by an environment  $c_k$  which provides the payoff for both players.

To get to a full understanding of this diagram (and deterministic open games), in this section we introduce several building blocks. Roughly, they can be classified in two kinds.

First, we need a way to architecture the information flow. As we will see in Sections 2.1 to 2.3, *lenses* play a crucial role by providing us with a categorical structure on which to build open games.

Second, we need to flesh out the internals of the boxes in the diagram. Specifically, how does strategic reasoning actually take place? Central here is the notion of an *agent* who makes observations and chooses a course of action. As we will see in Section 2.5 and Section 2.6, the key insight is to model an agent as choosing against a *context* which comprises how the environment reacts to an agent’s choices. The context is also the glue that keeps the outside information flow and the internal reasoning together.

Once we have introduced all the relevant parts, we will come back to the example above.

Note: The exposition in this section slightly deviates from previous work. We believe this eases the way for the generalisations to come in Section 3 and thereafter.

### 2.1 Lenses

The history of mathematical lenses is complicated, involving many independent discoveries and fresh starts across numerous areas of mathematics and computer science [2, 5, 9, 26, 32, 34]. An in-depth description of this history can be found at [25]. We use lenses to describe the flow of

information through a game. 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.

The lenses used in this paper are direct descendants of the lenses of database theory. Given some data  $x$  of type  $X$  we may want to view some part of it  $y$  of type  $Y$ . This is encapsulated by a *view function*  $v : X \rightarrow Y$ . 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 data  $x$ . That is, given initial data  $x$  and an updated view  $y' : Y$ , we should specify some updated  $x' : X$  given by some *update function*  $u : X \times Y \rightarrow X$ . The pair  $(v, u)$  is a *lens* with type  $X \rightarrow Y$ . The connection to our previous abstract definition of lenses is as follows:

- The covariant computation associated with the lens is the view function  $v : X \rightarrow Y$ ,
- 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  $u : X \times Y \rightarrow X$ .

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  $(X, S) \rightarrow (Y, R)$  where the covariant component of the lens is of type  $X \rightarrow Y$  and the contravariant component is of type  $X \times R \rightarrow S$ .

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 parallel composition as well as sequential composition, we obtain a nuanced notion of information flow in a game.

In the next subsections 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**. We then come to the core of this section, the definition of a *concrete open game*.

## 2.2 The category of concrete lenses

**Definition 2.2.1** (Concrete lens). Let  $X, S, Y$  and  $R$  be sets. A *concrete lens*  $l : (X, S) \rightarrow (Y, R)$  is a pair of functions  $(l_v : X \rightarrow Y, l_u : X \times R \rightarrow S)$ .

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

**Example 2.2.2.** Let  $f : X \rightarrow Y$  and  $g : R \rightarrow S$ . Define a concrete lens  $\langle f, g \rangle : (X, S) \rightarrow (Y, R)$  by

$$\begin{aligned} \langle f, g \rangle_v &= f \\ \langle f, g \rangle_u(x, r) &= g(r). \end{aligned}$$

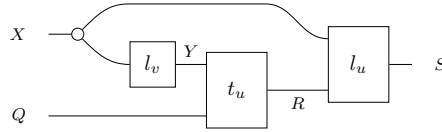
**Definition 2.2.3** (Sequential composition of concrete lenses). Let  $l : (X, S) \rightarrow (Y, R)$  and  $t : (Y, R) \rightarrow (Z, Q)$  be concrete lenses. The *sequential composite*  $t \circ l : (X, S) \rightarrow (Z, Q)$  is given by  $((t \circ l)_v : X \rightarrow Z, (t \circ l)_u : X \times Q \rightarrow S)$  where

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

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

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

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



**Lemma 2.2.4** (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  $n \circ (m \circ l) = (n \circ m) \circ l$ .

**Theorem 2.2.5** (Concrete lenses form a category). *There is a category  $\mathbf{CL}$  with pairs of sets as objects and concrete lenses as morphisms.*

### 2.3 The monoidal structure of concrete lenses

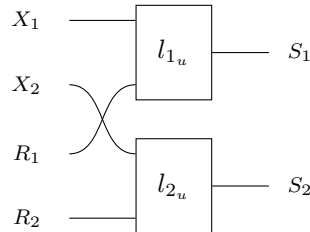
**Definition 2.3.1** (Tensor composition of concrete lenses). Let  $l_1 : (X_1, S_1) \rightarrow (Y_1, R_1)$  and  $l_2 : (X_2, S_2) \rightarrow (Y_2, R_2)$  be concrete lenses. The *tensor composition*  $l_1 \otimes l_2 : (X_1 \times X_2, S_1 \times S_2) \rightarrow (Y_1 \times Y_2, R_1 \times R_2)$  is given by  $((l_1 \otimes l_2)_v, (l_1 \otimes l_2)_u)$  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 2.3.2.**  $\otimes$  is a functor.

**Theorem 2.3.3.** *There is a symmetric monoidal category  $\mathbf{CL}$  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. The monoidal unit is  $I = (\{*\}, \{*\})$ .*

The following observations about states and effects in  $\mathbf{CL}$  will be useful in the remainder of this section.

**Lemma 2.3.4.**  $\mathbf{CL}(I, (X, S)) \cong X$ .

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

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

□

**Lemma 2.3.5.**  $\mathbf{CL}((Y, R), I) \cong (Y \rightarrow R)$

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

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

□

## 2.4 Concrete open games

Now we have the necessary prerequisites in place to introduce the notion of a *concrete open game*. A concrete open game consists of a set of strategy profiles; a family of concrete lenses indexed by the set of strategy profiles; and a best-response function.

**Definition 2.4.1** (Concrete open game). Let  $X, S, Y$ , and  $R$  be sets. A *concrete open game*  $\mathcal{G} : (X, S) \rightarrow (Y, R)$  is given by

1. A set of *strategy profiles*  $\Sigma$ ;
2. A *play function*  $P : \Sigma \rightarrow \mathbf{CL}((X, S), (Y, R))$ ; and
3. A *best-response function*  $B : X \times (Y \rightarrow R) \rightarrow \text{Rel}(\Sigma)$ .

Here  $\text{Rel}(\Sigma) = \mathcal{P}(\Sigma \times \Sigma)$  is the set of binary relations on  $\Sigma$ . In this paper we will usually specify a relation in terms of its forward images  $\Sigma \rightarrow \mathcal{P}(\Sigma)$ .

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 of  $\mathcal{G}$ . 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 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  $\sigma$  and  $\tau$  are related if  $\tau$  is the result of players unilaterally deviating from  $\sigma$  to their most profitable unilateral deviation. Nash equilibria are then the fixed points of this relation. For convenience, in the definition of a concrete open game we work directly with a best-response relation rather than preference relations.

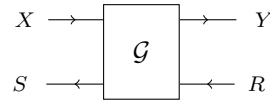
The play function takes a strategy as argument and returns a concrete lens that describes an *open play* of the game  $\mathcal{G}$  ('open' here means 'lacking a particular observation and outcome function' and is explained in the next paragraph). To justify this interpretation, recall that a concrete lens  $l : (X, S) \rightarrow (Y, R)$  consists of  $v : X \rightarrow Y$  and  $u : X \times R \rightarrow S$ . 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 2.4.2.** Let  $\mathcal{G} : (X, S) \rightarrow (Y, R)$ . A *history* for  $\mathcal{G}$  is an element  $x$  of  $X$ , an *outcome function* for  $\mathcal{G}$  is a function  $k : Y \rightarrow R$ , and a *context* for  $\mathcal{G}$  is a pair  $(x, k) : X \times (Y \rightarrow R)$ .

We are now in a position to justify the type of the best-response function. The best response functions 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  $\sigma$ .

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



This diagrammatic notation emphasises 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, as detailed in [21].

**Notation 2.4.3.** 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 2.7 and 2.8), 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 that are sensitive only to a given context.

**Definition 2.4.4** (Atomic concrete open game). A concrete open game  $a : (X, S) \rightarrow (Y, R)$  is *atomic* if

1.  $\Sigma_a \subseteq \mathbf{CL}((X, S), (Y, R))$ ;
2. For all  $l \in \Sigma_a$ ,  $P_{\mathcal{G}}(l) = l$ ; and
3. For all contexts  $c : X \times (Y \rightarrow R)$ ,  $B_a(c) : \Sigma_a \rightarrow \mathcal{P}(\Sigma_a)$  is constant.

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

Note that an atom  $a : (X, S) \rightarrow (Y, R)$  is fully determined by a subset  $\Sigma_a \subseteq \mathbf{CL}((X, S), (Y, R))$  and a *selection function*<sup>1</sup>  $\varepsilon : X \times (Y \rightarrow R) \rightarrow \mathcal{P}(\Sigma_a)$ .

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

**Example 2.4.5** (Computation). Let  $f : X \rightarrow Y$  and  $g : R \rightarrow S$ . The atom  $\langle f, g \rangle : (X, S) \rightarrow (Y, R)$  is given by

1.  $\Sigma = \{\langle f, g \rangle\}$ ; and
2. For all  $c : X \times (Y \rightarrow R)$ ,  $\varepsilon(c) = \{\langle f, g \rangle\}$ .

Similar to  $\mathbf{CL}$ , the following computations will turn out to be the underlying structural maps for the symmetric monoidal category of concrete open games.

<sup>1</sup>Selection functions have been studied in a game-theoretic context in other form. See, e.g. [6] or [24].

**Definition 2.4.6** (Structural computations). Define identity, associator, swaps, and left/right unitor computations to be the atomic concrete open games given by

$$\begin{aligned} \text{id}_{(X,S)} &= \langle \text{id}_X, \text{id}_S \rangle \\ \alpha_{X \otimes (Y \otimes Z), A \otimes (B \otimes C)} &= \langle \alpha_{X,Y,Z}, \alpha_{A,B,C}^{-1} \rangle \\ s_{(X,A),(Y,B)} &= \langle s_{(X,Y)}, s_{(A,B)}^{-1} \rangle \\ \rho_{(X,Y)} &= \langle \rho_X, \rho_Y^{-1} \rangle \\ \lambda_{(X,Y)} &= \langle \lambda_X, \lambda_Y^{-1} \rangle \end{aligned}$$

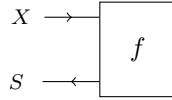
where the **Set** functions on the right-hand side of the equalities are the obvious **Set** isomorphisms.

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

**Definition 2.4.7** (Counit). Let  $f : X \rightarrow S$ . Define an atomic concrete open game  $c_f : (X, S) \rightarrow (\{\star\}, \{\star\})$  by

1.  $\Sigma_{c_f} = \{\langle !, f \rangle\}$ ; and
2. For all  $c : X \times (\{\star\} \rightarrow \{\star\})$ ,  $\varepsilon(c) = \Sigma_{c_f}$ .

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

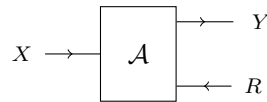


## 2.5 Agents

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 2.5.1** (Agent). An *agent*  $\mathcal{A} : (X, \{\star\}) \rightarrow (Y, R)$  is an atom whose set of strategies is  $\Sigma = \mathbf{CL}((X, \{\star\}), (Y, R))$ .

Recall that a concrete lens  $l : \mathbf{CL}((X, \{\star\}), (Y, R))$  is a pair  $(v : X \rightarrow Y, u : X \times R \rightarrow \{\star\})$  and, hence, is uniquely determined by a function of type  $X \rightarrow Y$ . 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  $c : X \times (Y \rightarrow R)$ ,  $B_{\mathcal{A}}(c)$  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 2.5.2** (Utility maximising agent). The *utility maximising agent*  $\mathcal{A} : (X, \{\star\}) \rightarrow (Y, \mathbb{R})$  is given by

$$\varepsilon(x, k) = \{\sigma : X \rightarrow Y \mid \sigma(x) \in \arg \max(k)\}.$$

There are other decision criteria one could use. For instance, MinMax and regret minimisation would be candidates. We could also consider models from behavioural game theory such as prospect theory. The only (very weak) requirement is that the decision criterion can be described by a selection function [23, 24]. In this paper we focus on utility-maximising agents as a simplification and in order to match traditional game theory.



## 2.6 Best-response with concrete lenses

Recall from 2.3.4 and 2.3.5 that  $\mathbf{CL}(I, (X, S)) \cong X$  and that  $\mathbf{CL}(I, (Y, R)) \cong Y \rightarrow R$ . Using these facts we can rephrase the type of best response for a concrete open game  $\mathcal{G} : (X, S) \rightarrow (Y, R)$  as

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

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  $x^*$  when talking about concrete lenses with type  $\mathbf{CL}(I, (X, S))$ . Similarly, we write  $k : Y \rightarrow R$  when talking about functions in  $\mathbf{Set}$  and we write  $k_*$  when talking about effects in  $\mathbf{CL}((Y, R), I)$ .

## 2.7 Sequential composition of concrete open games

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

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  $\Sigma_{\mathcal{G} \circ \mathcal{H}}$  to be the set of strategy profiles of  $\mathcal{H} \circ \mathcal{G}$ .

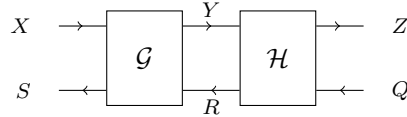
The play function of the sequential composite is defined straightforwardly using the sequential composition of concrete lenses defined in 2.2.3.

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  $c = (x : X, k : Z \rightarrow Q)$  and a strategy  $(\sigma, \tau)$  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  $\tau$  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  $\sigma$ . 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 2.7.1** (Sequential composition for concrete open games). Let  $\mathcal{G} = (\Sigma_{\mathcal{G}}, \mathbf{P}_{\mathcal{G}}, \mathbf{B}_{\mathcal{G}}) : (X, S) \rightarrow (Y, R)$  and  $\mathcal{H} = (\Sigma_{\mathcal{H}}, \mathbf{P}_{\mathcal{H}}, \mathbf{B}_{\mathcal{H}}) : (Y, R) \rightarrow (Z, Q)$  be concrete open games. Define

1.  $\Sigma_{\mathcal{H} \circ \mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$ ,
2.  $\mathbf{P}_{\mathcal{H} \circ \mathcal{G}}(\sigma, \tau) = \mathbf{P}_{\mathcal{H}}(\tau) \circ \mathbf{P}_{\mathcal{G}}(\sigma)$  (where  $\circ$  composition is in  $\mathbf{CL}$ ), and
3.  $\mathbf{B}_{\mathcal{H} \circ \mathcal{G}}(x^*, k_*)(\sigma, \tau) = \mathbf{B}_{\mathcal{G}}(x^*, k_* \circ \mathbf{P}_{\mathcal{H}}(\tau))\sigma \times \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}(\sigma) \circ x^*, k_*)(\tau)$ .

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



## 2.8 Tensor composition for concrete open games

The tensor composition of open games represents simultaneous play. Given concrete open games  $\mathcal{G} : (X_1, S_1) \rightarrow (Y_1, R_1)$  and  $\mathcal{H} : (X_2, S_2) \rightarrow (Y_2, R_2)$ , 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  $\mathbf{CL}$  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  $\mathcal{H}$ .

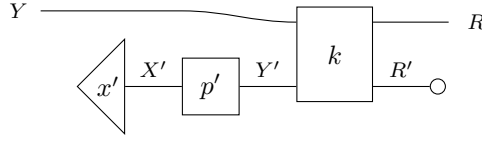
**Definition 2.8.1** (Local contexts for tensor composition). Define the *left local tensor context operator*

$$\mathcal{L} : \left( X' \times (X' \rightarrow Y') \times (Y \times Y' \rightarrow R \times R') \right) \rightarrow (Y \rightarrow R)$$

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



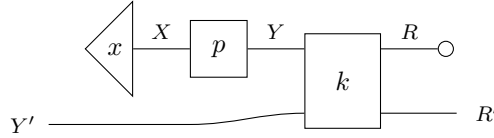
Similarly, define the *right local tensor context operator*

$$\mathcal{R} : \left( X \times (X \rightarrow Y) \times (Y \times Y' \rightarrow R \times R') \right) \rightarrow (Y' \rightarrow R')$$

by

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

As a diagram,



Suppose we have concrete open games  $\mathcal{G} : (X, S) \rightarrow (Y, R)$  and  $\mathcal{H} : (X', S') \rightarrow (Y', R')$  and we wish to combine them to create some game  $\mathcal{G} \otimes \mathcal{H} : (X \times X', S \times S') \rightarrow (Y \times Y', R \times R')$ . 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 2.8.2** (Tensor composition of concrete open games). Let  $\mathcal{G} : (X, S) \rightarrow (Y, R)$  and  $\mathcal{H} : (X', S') \rightarrow (Y', R')$  be concrete open games. Define

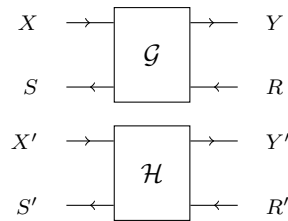
$$\mathcal{G} \otimes \mathcal{H} : (X \times X', S \times S') \rightarrow (Y \times Y', R \times R')$$

by

1.  $\Sigma_{\mathcal{G} \otimes \mathcal{H}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$ ,
2.  $P_{\mathcal{G} \otimes \mathcal{H}}(\sigma, \tau) = P_{\mathcal{G}}(\sigma) \otimes P_{\mathcal{H}}(\tau)$  (in **CL**),
3.  $B_{\mathcal{G} \otimes \mathcal{H}} : \left( (X \times X') \times (Y \times Y' \rightarrow R \times R') \right) \rightarrow \text{Rel}(\Sigma_{\mathcal{G} \otimes \mathcal{H}})$  is given by

$$B_{\mathcal{G} \otimes \mathcal{H}}((x, x')^*, k_*)(\sigma, \tau) = B_{\mathcal{G}}(x^*, \mathcal{L}(x', (P_{\mathcal{H}}(\tau))_v, k)_*)(\sigma) \\ \times B_{\mathcal{H}}(x'^*, \mathcal{R}(x, (P_{\mathcal{G}}(\sigma))_v, k)_*)(\tau)$$

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



## 2.9 Equivalence of open games

One subtlety remains before we can define the category of concrete open games. We aim to define a category with pairs of sets as objects 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  $\mathcal{K} \circ (\mathcal{H} \circ \mathcal{G})$  is  $(\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times \Sigma_{\mathcal{K}}$  whilst the strategy set of  $(\mathcal{K} \circ \mathcal{H}) \circ \mathcal{G}$  is  $\Sigma_{\mathcal{G}} \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{K}})$ . In order for concrete open games to form a category, we must first take an appropriate quotient.

There are several different reasonable choices of quotient. Since this is an orthogonal consideration to this paper's topic, we choose the most straightforward, which is to identify open games that have a compatible isomorphism between their sets of strategies. Other choices that can be made are bisimulations [3] and surjections [8]. Alternatively, instead of taking a quotient, open games can be considered as the 1-cells of a bicategory [22].

**Definition 2.9.1.** Let  $\mathcal{G}, \mathcal{H} : (X, S) \rightarrow (Y, R)$  be concrete open games. An *isomorphism*  $\alpha : \mathcal{G} \rightarrow \mathcal{H}$  is given by a bijection  $\alpha : \Sigma_{\mathcal{G}} \rightarrow \Sigma_{\mathcal{H}}$  such that

1.  $P_{\mathcal{G}}(\sigma) = P_{\mathcal{H}}(\alpha(\sigma))$  for all  $\sigma \in \Sigma_{\mathcal{G}}$
2. For all  $\sigma, \sigma' \in \Sigma_{\mathcal{G}}$  and  $c \in X \times (Y \rightarrow R)$ ,  $(\sigma, \sigma') \in B_{\mathcal{G}}(c)$  iff  $(\alpha(\sigma), \alpha(\sigma')) \in B_{\mathcal{H}}(c)$

**Definition 2.9.2.** Let  $\mathcal{G}, \mathcal{H} : (X, S) \rightarrow (Y, R)$  be concrete open games.  $\mathcal{G}$  and  $\mathcal{H}$  are *equivalent*, written  $\mathcal{G} \sim \mathcal{H}$ , if there exists an isomorphism  $\alpha : \mathcal{G} \rightarrow \mathcal{H}$ . We write  $[\mathcal{G}]$  for the equivalence class of  $\mathcal{G}$  under this relation. We also say that the isomorphism  $\alpha$  *witnesses* the equivalence between  $\mathcal{G}$  and  $\mathcal{H}$  and write  $\mathcal{G} \stackrel{\alpha}{\sim} \mathcal{H}$ .

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

**Lemma 2.9.3.** Let  $\mathcal{G}, \mathcal{G}' : (X, S) \rightarrow (Y, R)$  and  $\mathcal{H}, \mathcal{H}' : (Y, R) \rightarrow (Z, Q)$  be concrete open games. If  $\mathcal{G} \sim \mathcal{G}'$  and  $\mathcal{H} \sim \mathcal{H}'$ , then  $\mathcal{H} \circ \mathcal{G} \sim \mathcal{H}' \circ \mathcal{G}'$ .

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

$$(\alpha \times \beta)(\sigma, \tau) = (\alpha(\sigma), \beta(\tau))$$

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

**Lemma 2.9.4.** Let  $\mathcal{G}, \mathcal{H} : (X, S) \rightarrow (Y, R)$  and  $\mathcal{G}', \mathcal{H}' : (X', S') \rightarrow (Y', R')$  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  $\mathcal{G} \stackrel{\alpha}{\sim} \mathcal{H}$  and  $\mathcal{G}' \stackrel{\beta}{\sim} \mathcal{H}'$ , then  $\mathcal{G} \otimes \mathcal{G}' \stackrel{\alpha \times \beta}{\sim} \mathcal{H} \otimes \mathcal{H}'$  as in the previous lemma. □

## 2.10 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 2.10.1.** In string diagrams we refer to a play function applied to a strategy simply by the strategy. For example,  $\sigma$  may refer to  $P_{\mathcal{G}}(\sigma)$ . 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 2.10.2.** *Sequential composition of concrete open games is associative up to equivalence.*

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

**Lemma 2.10.3.** Let  $\mathcal{G} : (X, S) \rightarrow (Y, R)$ . Then  $[\mathcal{G}] = [\mathcal{G} \circ \langle \text{id}_X, \text{id}_S \rangle] = [\langle \text{id}_Y, \text{id}_R \rangle \circ \mathcal{G}]$ .

**Corollary 2.10.4.** *There is a category **ConGame** with pairs of sets as object and equivalence classes of concrete open games as morphisms.* □

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

**Lemma 2.10.5.**  $\otimes : \mathbf{ConGame} \times \mathbf{ConGame} \rightarrow \mathbf{ConGame}$  is a functor.

**Lemma 2.10.6.** The associator in **ConGame** is natural.

We include the proof of this lemma specifically because it will be important later.

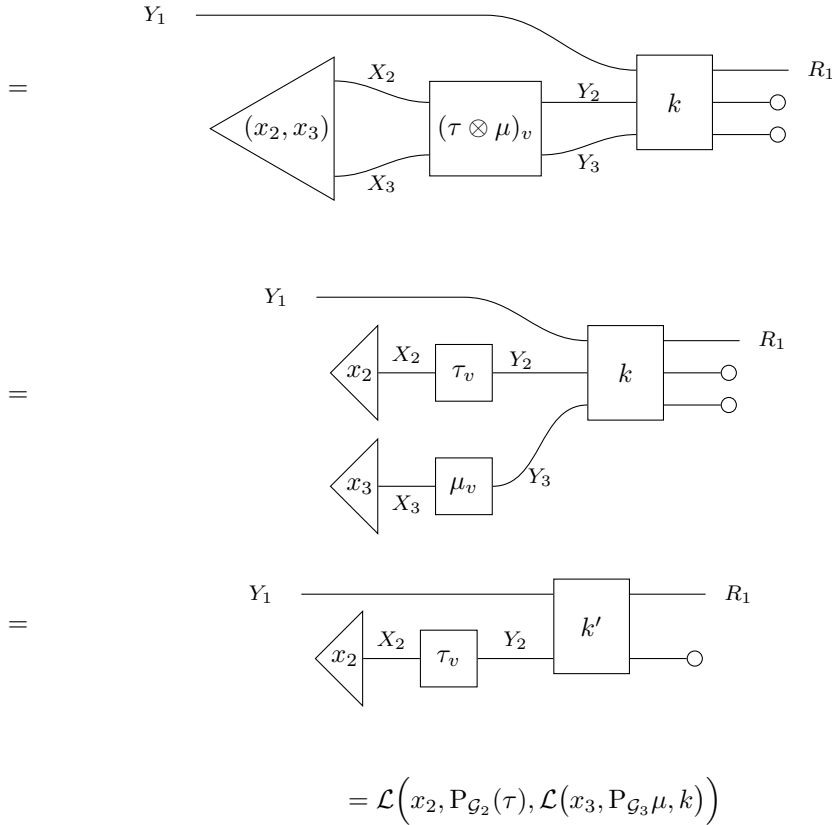
*Proof.* Let  $\mathcal{G}_i : (X_i, S_i) \rightarrow (Y_i, R_i)$  be open games where  $i \in \{1, 2, 3\}$ . We need to show that  $\alpha \circ (\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3)) \sim ((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3) \circ \alpha$ . Define  $\beta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

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

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

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

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



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

$$k_{\mathcal{G}_2} := \mathcal{R}(x_1, P_{\mathcal{G}_1}(\sigma), \mathcal{L}(x_3, P_{\mathcal{G}_3}(\mu), k)) = \mathcal{L}(x_3, P_{\mathcal{G}_3}(\mu), \mathcal{R}(x_1, P_{\mathcal{G}_1}(\sigma), k))$$

and

$$k_{\mathcal{G}_3} := \mathcal{R}((x_1, x_2), P_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma, \tau), k) = \mathcal{R}(x_2, P_{\mathcal{G}_2}(\tau), \mathcal{R}(x_1, P_{\mathcal{G}_1}(\sigma), k)).$$

Then

$$\begin{aligned} & ((\star, (\sigma, (\tau, \mu))), (\star, (\sigma', (\tau', \mu')))) \in B_{\alpha \circ (\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3))}((x_1, (x_2, x_3)), k) \\ \iff & (\sigma, \sigma') \in B_{\mathcal{G}_1}(x_1, k_{\mathcal{G}_1}) \text{ and } (\tau, \tau') \in B_{\mathcal{G}_2}(x_2, k_{\mathcal{G}_2}) \text{ and } (\mu, \mu') \in B_{\mathcal{G}_3}(x_3, k_{\mathcal{G}_3}) \\ \iff & (((\sigma, \tau), \mu), \star), (((\sigma', \tau'), \mu'), \star)) \in B_{((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3) \circ \alpha}(((x_1, x_2), x_3), k) \end{aligned}$$

□

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

$$\begin{array}{c} \triangleleft \\ \hline s \\ \hline \triangleleft \\ \hline S_1 \\ \hline S_2 \end{array} = \begin{array}{c} \triangleleft \\ \hline s_1 \\ \hline S_1 \\ \hline \triangleleft \\ \hline s_2 \\ \hline S_2 \end{array}$$

This poses a significant barrier to generalising concrete open games to monoidal categories where the monoidal tensor is not cartesian, and Section 3 addresses this problem.

**Lemma 2.10.7.** *The structural computations  $\lambda, \rho$ , and  $s$  are natural in **ConGame**.*

**Theorem 2.10.8.** ***ConGame** is symmetric monoidal.*

### 2.11 Encoding functions as games

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

**Lemma 2.11.1** ([20]). *Define  $F : \mathbf{Set} \times \mathbf{Set}^{\text{op}} \rightarrow \mathbf{ConGame}$  by  $F(X, S) = (X, S)$  and  $F(f : X \rightarrow Y, g : R \rightarrow S) = \langle f, g \rangle$ . Then  $F$  is a faithful monoidal functor.*  $\square$

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

$$\begin{array}{c} X \longrightarrow \boxed{f} \longrightarrow Y \\ S \longleftarrow \boxed{g} \longleftarrow R \end{array}$$

Two particularly useful examples of this notation are the covariant and contravariant copying computations  $\langle \Delta_X, \text{id}_1 \rangle : (X, 1) \rightarrow (X \times X, 1)$  and  $\langle \text{id}_1, \Delta_R \rangle : (1, R \times R) \rightarrow (1, R)$  which are represented by

$$\begin{array}{c} X \longrightarrow \circ \begin{array}{l} \nearrow X \\ \searrow X \end{array} \end{array} \quad \text{and} \quad \begin{array}{c} \begin{array}{l} R \nearrow \\ R \searrow \end{array} \circ \longleftarrow R \end{array}$$

respectively.

### 2.12 Game theory with concrete open games

In this section we give some examples of games modelled 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 [20] for more details.

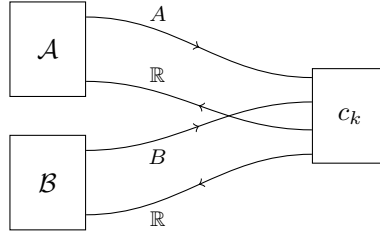
#### 2.12.1 Bimatrix games

Bimatrix games are simply two-player normal-form games, the most well-known example of which is likely the prisoner’s dilemma. We assume the set of actions available to each player is finite for simplicity.

**Definition 2.12.2.** *A bimatrix game consists of*

1. Finite set of actions  $A$  and  $B$ ; and
2. An outcome function  $k : A \times B \rightarrow \mathbb{R}^2$ .

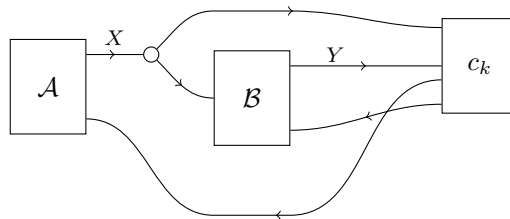
A bimatrix game  $\mathcal{G} = (A, B, k)$  is represented by the concrete open game



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  $B : A \times B \rightarrow \mathcal{P}(A \times B)$  of the above concrete open game, i.e. strategy profiles  $(a, b)$  satisfying  $(a, b) \in B(a, b)$ .

### 2.12.3 Two-player sequential game

A two-player sequential game is defined by the same data as a bimatrix game (sets  $A$  and  $B$  and a function  $k : A \times B \rightarrow \mathbb{R}^2$ ), but we allow the second player to observe the first player's move before making a choice, so strategies for the second player are functions  $A \rightarrow B$ . This 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  $k$ .

Crucially, the fixed points of the concrete open game are *not* subgame perfect Nash equilibria, but rather plain old Nash equilibria. It is also possible to define a concrete open game that captures subgame perfect equilibria, but this requires an additional operator defined in [14].

### 2.12.4 Normal-form games

Let  $\Gamma = (N, (S_i)_{i=1}^N, (u_i)_{i=1}^N)$  be a normal-form game for  $N$  players.  $S_i$  denotes the set of strategies available to player  $i$ . The function  $u_i$  maps a strategy tuple for all players,  $\prod_{i=1}^N S_i$ , to player  $i$ 's payoff, in  $\mathbb{R}$ . Define  $k : \prod_{i=1}^N S_i \rightarrow \mathbb{R}^N$  by  $s = (s_1, \dots, s_N) \mapsto (u_1(s), \dots, u_N(s))$ . We can model this normal-form game using the concrete open game

$$c_k \circ \left( \bigotimes_{i=1}^N \mathcal{A}_i \right)$$

where  $\mathcal{A}_i : I \rightarrow (S_i, \mathbb{R})$  is the utility maximising agent. The fixed points of this game's best-response relation are then the pure-strategy Nash equilibria of the normal-form game.

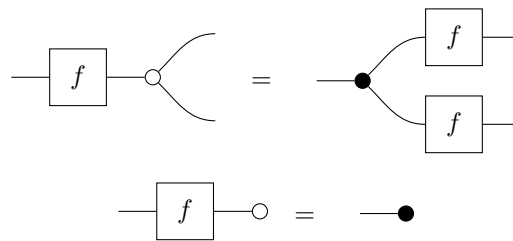
### 3 General open games

The notion of open game we introduced in the section before can emulate some standard games such as the prisoner’s dilemma. On the other hand, classical game theory has a much wider reach. It can model situations with which a concrete open game cannot deal. This involves stochastic environments, probabilistic choices by players, and incomplete information.

In this section, we will significantly generalise the notion of an open game, to make room for these three extensions (and beyond). The first order of business, to make progress in this direction, is to generalise concrete lenses.

#### 3.1 Generalising concrete lenses

In the proof of Lemma 2.2.4 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 Definition 2.2.3, 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 Section 4).  $\mathbf{Kl}(D)$  inherits a copy/delete comonoid from **Set**, but its comonoid homomorphisms are the deterministic maps (i.e. precisely the non-probabilistic maps).

In the next section we introduce *coends*, a piece of categorical machinery that allows for an elegant generalisation of concrete lenses to arbitrary symmetric monoidal categories. We call these generalised lenses *coend lenses* or, simply, *lenses*. We will first introduce the technical notion before magicking it away with a diagrammatic calculus that represents what is ‘really’ going on.

#### 3.2 Co-wedges and Coends

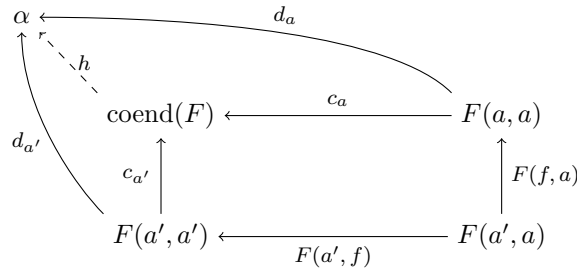
Co-wedges are a variant of co-cones of natural transformations applying to functors that act both covariantly and contravariantly on an argument. In Section 2.1 we noted that lenses have both covariant and contravariant components. We will see that this behaviour can be described by coends, which are initial co-wedges. For extra motivation, discussion, and examples, we refer the reader to [27].

**Definition 3.2.1** (Co-wedge). Let  $F : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{D}$  be a functor. A *co-wedge*  $c : F \rightarrow \alpha$  is an object  $\alpha : \mathcal{D}$  together with maps  $\{c_a : F(a, a) \rightarrow \alpha \mid a : \mathcal{C}\}$  such that, for any morphism  $f : a' \rightarrow a$ , the diagram

$$\begin{array}{ccc}
 \alpha & \xleftarrow{c_a} & F(a, a) \\
 c_{a'} \uparrow & & \uparrow F(f, a) \\
 F(a', a') & \xleftarrow{F(a', f)} & F(a', a)
 \end{array}$$

commutes.

**Definition 3.2.2** (Coend). A *coend* is a couniversal co-wedge. Diagrammatically, the coend of a functor  $F : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{D}$  is a co-wedge  $\{c_a : F(a, a) \rightarrow \text{coend}(F) \mid a : \mathcal{C}\}$  such that for any other co-wedge  $\{d_a : F(a, a) \rightarrow \alpha \mid a : \mathcal{C}\}$  and morphism  $f : a' \rightarrow a$  the diagram



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

We adopt the integral notation for coends, writing

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

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

**Lemma 3.2.3.** *Let  $F : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{D}$ . 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 pair of arrows*

$$\coprod_{\substack{a, a' : \mathcal{C} \\ f : a' \rightarrow a}} F(a, a') \begin{array}{c} \xrightarrow{F_1} \\ \xrightarrow{F_2} \end{array} \coprod_{a:\mathcal{C}} F(a, a),$$

where the  $f : a' \rightarrow a$  components of  $F_1$  and  $F_2$  are  $F(f, a')$  and  $F(a, f)$  respectively. □

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

### 3.3 Coend lenses

Much of the material in this section is worked out in much greater detail in [36], which serves as a good standard reference for coend lenses. We first give an abstract definition of coend lenses, then provide some justification.

**Definition 3.3.1** (Coend lens). Let  $X, S, Y$ , and  $R$  be objects in a symmetric monoidal category  $\mathcal{C}$ . A *coend lens*  $l : (X, S) \rightarrow (Y, R)$  is an element of the set

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

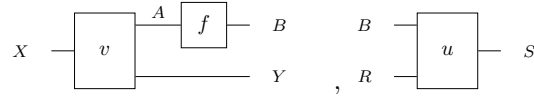
We think of the coend in the above definition as acting as a kind of existential quantifier over the type variable  $A$ , followed by a quotient (to be described) over the resulting structure. That is, a coend lens  $l : (X, S) \rightarrow (Y, R)$  consists of an equivalence relation over triples comprised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

By Lemma 3.2.3 we can characterise coend lenses  $(X, S) \rightarrow (Y, R)$  as the elements of a particular coequaliser. Moreover, coequalisers in **Set** are given by quotients. Unpacking the coequaliser explicitly, coend lenses  $(X, S) \rightarrow (Y, R)$  are given by the set of triples of the form described above, quotiented by the equivalence relation generated by (i.e. the smallest equivalence relation containing)

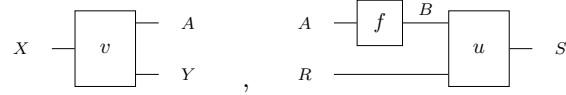
$$((f \otimes \text{id}_Y) \circ v, u) \sim (v, u \circ (f \otimes \text{id}_R))$$

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



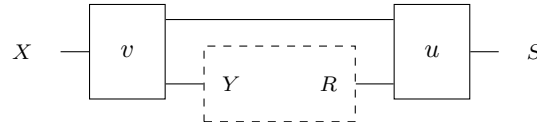


is related to the pair

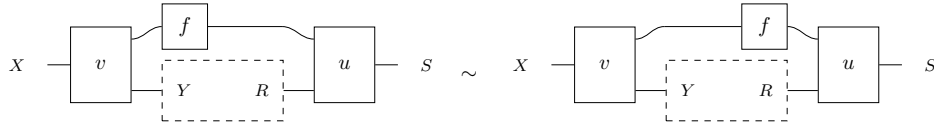


We refer to the types  $A$  and  $B$  as *bound types* ( $B$  is bound in the first diagram,  $A$  in the second). In Section 4, we will see that this bound type keeps track of *correlations* between random variables in the Kleisli category of the distribution monad.

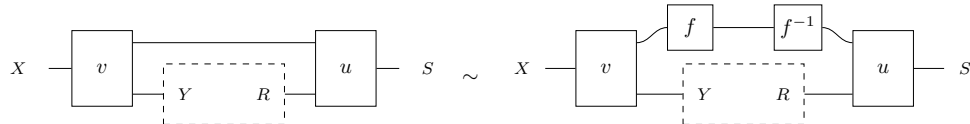
In vague terms, two pairs of morphisms are related if one can get from one to the other by ‘sliding’ a morphism off the bound type of one morphism on to the bound type of the other. Given a pair of morphisms  $(v : X \rightarrow A \otimes Y, u : A \otimes R \rightarrow S)$ , we write  $[v, u]$  for their 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 [36], taking the hint from the diagrammatic representation of the equivalence relation by representing a coend lens  $[v, u] : (X, S) \rightarrow (Y, R)$  as



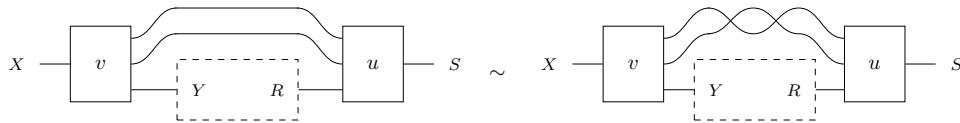
We usually omit the bound type in diagrams for clarity. The equivalence relation is then simply



The equivalence relation permits the cancelling of isomorphisms:

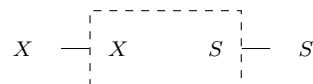


Many proofs in this section proceed by allowing symmetric monoidal structure to interact with coend structure as, for example, in the following diagram.

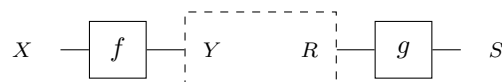


The formal foundations of this class of diagrams are investigated in [37].

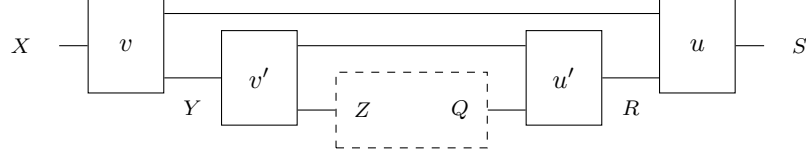
**Example 3.3.2** (Identity lens). The *identity lens*  $\text{id}_{(X,S)} : (X, S) \rightarrow (X, S)$  is given by  $[I, \text{id}_X : X \rightarrow X, \text{id}_S : S \rightarrow S]$ . Diagrammatically,



**Example 3.3.3.** A pair of morphisms  $(f : X \rightarrow Y, g : R \rightarrow S)$  is encoded by the coend lens  $[I, f, g] : (X, S) \rightarrow (Y, R)$ :



**Definition 3.3.4** (Sequential composition of coend lenses). Let  $[v, u] : (X, S) \rightarrow (Y, R)$  and  $[v', u'] : (Y, R) \rightarrow (Z, Q)$  be coend lenses. Define  $[v', u'] \circ [v, u] : (X, S) \rightarrow (Z, Q)$  to be



Explicitly,

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

**Theorem 3.3.5** (Coend lenses form a category). Suppose  $\mathcal{C}$  is a monoidal category such that, for all objects  $X, S, Y, R \in \mathcal{C}$ ,

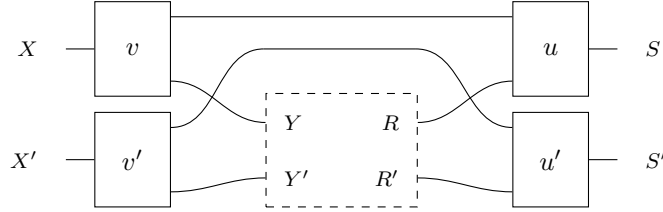
$$\int^{A:\mathcal{C}} \mathcal{C}(X, A \otimes Y) \times \mathcal{C}(A \otimes R, S)$$

exists. Then there is a category  $\mathbf{Lens}_{\mathcal{C}}$  whose objects are pairs of objects in  $\mathcal{C}$  and where

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

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

**Definition 3.3.6** (Tensor composition of coend lenses). Let  $[v, u] : (X, S) \rightarrow (Y, R)$  and  $[v', u'] : (X', S') \rightarrow (Y', R')$  be coend lenses. Define  $[v, u] \otimes [v', u'] : (X \otimes X', S \otimes S') \rightarrow (Y \otimes Y', R \otimes R')$  to be



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

$$\left[ (A \otimes A', \text{id}_A \otimes s_{Y, A'} \otimes \text{id}_{Y'}) \circ (v \otimes v'), (u \otimes u') \circ (\text{id}_A \otimes s_{A', R} \otimes \text{id}_{R'}) \right].$$

**Theorem 3.3.7** ( $\mathbf{Lens}_{\mathcal{C}}$  is symmetric monoidal). The category  $\mathbf{Lens}_{\mathcal{C}}$  is symmetric monoidal with the tensor given in Definition 3.3.6, monoidal unit  $I = (I_{\mathcal{C}}, I_{\mathcal{C}})$ , and with structural morphisms inherited from  $\mathcal{C}$  given by

$$\begin{aligned} \alpha_{(X,A),(Y,B),(Z,C)} &= [\alpha_{X,Y,Z}, \alpha_{A,B,C}^{-1}] \\ \lambda_{(X,A)} &= [\lambda_X, \lambda_A^{-1}] \\ \rho_{(X,A)} &= [\rho_X, \rho_A^{-1}] \\ s_{(X,A),(Y,B)} &= [s_{X,Y}, s_{B,A}]. \end{aligned}$$

**Lemma 3.3.8.**  $\mathbf{Lens}_{\mathbf{Set}}$  is isomorphic to  $\mathbf{CL}$ . (More generally, when  $\otimes$  is cartesian,  $\mathbf{Lens}_{\mathcal{C}}$  is isomorphic to an appropriately generalised definition of  $\mathbf{CL}_{\mathcal{C}}$ .)  $\square$

### 3.4 Towards generalising open games

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

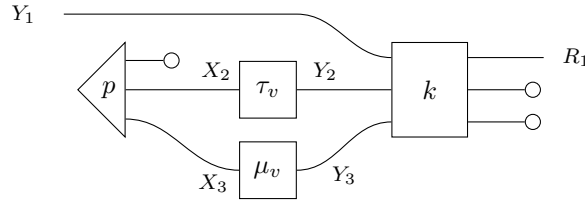
1. A set  $\Sigma$  of strategies;
2. A play function  $P : \Sigma \rightarrow \mathbf{Lens}_{\mathcal{C}}((X, S), (Y, R))$ ; and
3. A best-response function  $B : \mathcal{C}(I, X) \times \mathcal{C}(Y, R) \rightarrow \mathbf{Rel}(\Sigma)$ .

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.

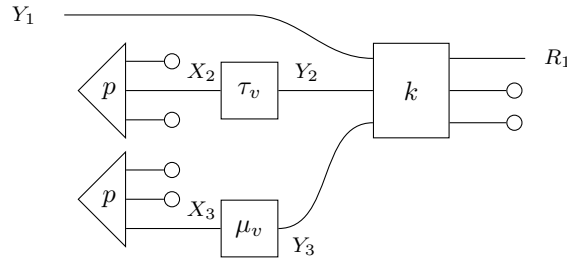
In proving that the associator was natural in **CL**, we used the fact that the monoidal tensor in **Set** is cartesian. If the tensor of  $\mathcal{C}$  is *not* cartesian, the local context of  $\mathcal{G}$  in  $\mathcal{G} \otimes (\mathcal{H} \otimes \mathcal{K})$  is different to the local context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

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

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

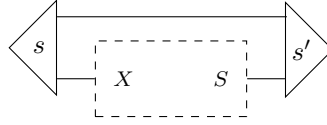


In general, these morphisms are *not* the same. In the case where  $\mathcal{C}$  is the Kleisli category of the distribution monad, the first morphism contains information about correlations between the types  $X_2$  and  $X_3$  whilst the second morphism does not. Consequently, the distinction between 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 that this abstractness allows for more elegant proofs and, in any case, disappears when dealing with the categories we are actually interested in.

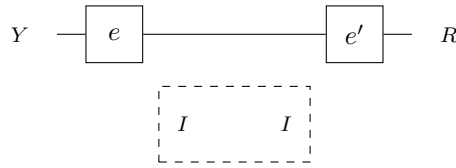
### 3.5 States, continuations, and contexts

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  $\mathcal{C}$ . A useful interpretation of states in  $\mathbf{Lens}_{\mathcal{C}}$  is as a *history/cohistory* pair. (Cohistories are not yet well understood. They make proofs easier, but vanish in categories which make game-theoretic sense.)

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



Concerning effects, we have the following result.

**Lemma 3.5.1.**  $\mathcal{C}(Y, R) \cong \mathbf{Lens}_{\mathcal{C}}((Y, R), I)$

*Proof.* The isomorphism  $i : \mathcal{C}(Y, R) \rightarrow \mathbf{Lens}_{\mathcal{C}}((Y, R), I)$  is given by

$$i(f : Y \rightarrow R) = [R, f, \text{id}_R] = [Y, \text{id}_Y, f].$$

□

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 of 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 *double coend*. This double coend turns out to be a state in the double lens category  $\mathbf{Lens}_{\mathbf{Lens}_{\mathcal{C}}}$ . From a purely technical standpoint, using double lenses allows for elegant proofs. From a heuristic perspective, we will see that the extra bound variable the double 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.

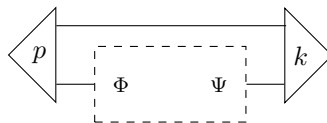
**Definition 3.5.2** (Context functor). The *context functor*  $\mathbb{C} : \mathbf{Lens}_{\mathcal{C}} \times \mathbf{Lens}_{\mathcal{C}}^{\text{op}} \rightarrow \mathbf{Set}$  is given by

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

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

(We use the letters  $\Phi, \Psi$  to refer to objects of  $\mathbf{Lens}_{\mathcal{C}}$ , which are pairs of objects of  $\mathcal{C}$ .)

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



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

### 3.6 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 of 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 3.6.1** (Open game). Let  $\Phi, \Psi \in \mathbf{Lens}_{\mathcal{C}}$ . An *open game*  $\mathcal{G} : \Phi \rightarrow \Psi$  consists of

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

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 3.6.2.** An *atomic open game*  $a : \Phi \rightarrow \Psi$  is an open game such that

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

Atomic open games are uniquely specified by a subset  $\Sigma \subseteq \mathbf{Lens}_{\mathcal{C}}(\Phi, \Psi)$  and a selection function  $B : \mathbb{C}(\Phi, \Psi) \rightarrow \mathcal{P}(\Sigma)$ , and we will sometimes specify atomic open games via this data. We refer to atomic open games simply as *atoms*.

**Example 3.6.3.** The *identity atom*  $\text{id}_{\Phi} : \Phi \rightarrow \Phi$  is given by  $\Sigma = \{\text{id}_{\Phi}\}$ ,  $B(c) = \{\text{id}_{\Phi}\}$  for all  $c \in \mathbb{C}(\Phi, \Phi)$ .

**Example 3.6.4** (Computation). Let  $f : \mathcal{C}(X, Y)$  and  $g : \mathcal{C}(R, S)$  be morphisms in  $\mathcal{C}$ . Define the atom  $\langle f, g \rangle : (X, S) \rightarrow (Y, R)$  by

1.  $\Sigma_{\langle f, g \rangle} = \{[f, g]\}$ ; and
2.  $B_{\langle f, g \rangle}(c) = \{[f, g]\}$  for all  $c \in \mathbb{C}((X, S), (Y, R))$ .

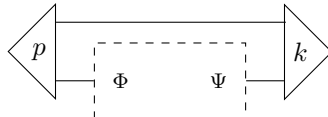
### 3.7 Composing open games

The heuristic for sequential composition of general open games is much the same as for concrete open games in Subsection 2.7. 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  $\mathcal{H}$ .

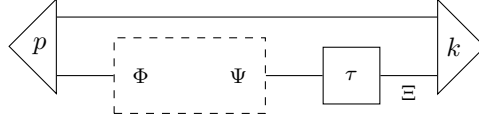
**Definition 3.7.1** (Sequential composition of open games). Let  $\mathcal{G} : \Phi \rightarrow \Psi$  and  $\mathcal{H} : \Psi \rightarrow \Xi$  be open games. Define  $\mathcal{H} \circ \mathcal{G} : \Phi \rightarrow \Xi$  by

1.  $\Sigma_{\mathcal{H} \circ \mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$ ,
2.  $P_{\mathcal{H} \circ \mathcal{G}}(\sigma, \tau) = P_{\mathcal{H}}(\tau) \circ P_{\mathcal{G}}(\sigma)$ ,
3.  $B_{\mathcal{H} \circ \mathcal{G}}([p, k])(\sigma, \tau) = B_{\mathcal{G}}([p, k \circ P_{\mathcal{H}}(\tau)])(\sigma) \times B_{\mathcal{H}}([P_{\mathcal{G}}(\sigma) \circ p, k])(\tau)$ .

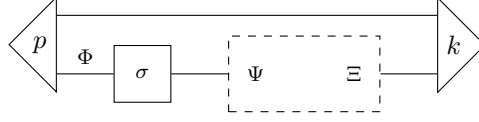
Given a context  $[p, k] \in \mathbb{C}(\Phi, \Xi)$  represented by the diagram



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.

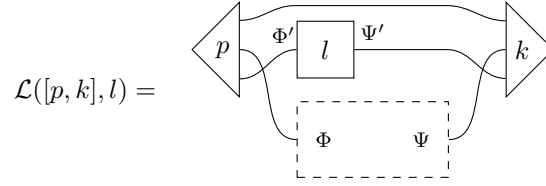
### 3.8 The tensor of open games

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 3.8.1** (Local contexts for tensor composition). Define the *left local context function*

$$\mathcal{L}_{\Phi, \Phi', \Psi, \Psi'} : \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') \times \mathbf{Lens}_{\mathcal{C}}(\Phi', \Psi') \rightarrow \mathbb{C}(\Phi, \Psi)$$

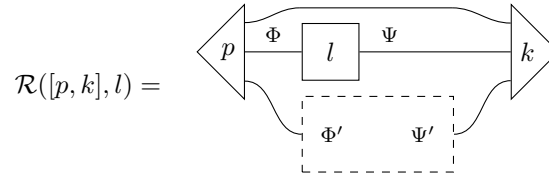
by



Define the *right local context function*

$$\mathcal{R}_{\Phi, \Phi', \Psi, \Psi'} : \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') \times \mathbf{Lens}_{\mathcal{C}}(\Phi, \Psi) \rightarrow \mathbb{C}(\Phi', \Psi')$$

by



We will usually suppress the subscripts of  $\mathcal{L}$  and  $\mathcal{R}$  as the types can be inferred from context.

**Definition 3.8.2** (Tensor composition of open games). Let  $\mathcal{G} : \Phi \rightarrow \Psi$  and  $\mathcal{H} : \Phi' \rightarrow \Psi'$  be open games. Define  $\mathcal{G} \otimes \mathcal{H} : \Phi \otimes \Phi' \rightarrow \Psi \otimes \Psi'$  by

- $\Sigma_{\mathcal{G} \otimes \mathcal{H}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$ ;
- $P_{\mathcal{G} \otimes \mathcal{H}}(\sigma, \tau) = P_{\mathcal{G}}(\sigma) \otimes P_{\mathcal{H}}(\tau)$  (in  $\mathbf{Lens}_{\mathcal{C}}$ );
- Define  $B_{\mathcal{G} \otimes \mathcal{H}} : \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') \rightarrow \mathbf{Rel}(\Sigma_{\mathcal{G} \otimes \mathcal{H}})$  by

$$B_{\mathcal{G} \otimes \mathcal{H}}(c)(\sigma, \tau) = B_{\mathcal{G}}(\mathcal{L}(c, P_{\mathcal{H}}(\tau)))(\sigma) \times B_{\mathcal{H}}(\mathcal{R}(c, P_{\mathcal{G}}(\sigma)))(\tau)$$

### 3.9 Equivalence of open games

As in Section 2.9, we need to quotient open games in order to obtain a category.

**Definition 3.9.1** (Isomorphism of open games). Let  $\mathcal{G}, \mathcal{H} : \Phi \rightarrow \Psi$  be open games. An *isomorphism* of open games  $\alpha : \mathcal{G} \rightarrow \mathcal{H}$  is a bijection  $\alpha : \Sigma_{\mathcal{G}} \rightarrow \Sigma_{\mathcal{H}}$  such that

1.  $P_{\mathcal{G}}(\sigma) = P_{\mathcal{H}}(\alpha(\sigma))$  for all  $\sigma \in \Sigma_{\mathcal{G}}$ ; and
2. For all  $\sigma, \sigma' \in \Sigma_{\mathcal{G}}$  and  $c \in \mathbb{C}(\Phi, \Psi)$ ,  $(\sigma, \sigma') \in B_{\mathcal{G}}(c)$  iff  $(\alpha(\sigma), \alpha(\sigma')) \in B_{\mathcal{H}}(c)$ .

**Definition 3.9.2** (Equivalence of open games). Let  $\mathcal{G}, \mathcal{H} : \Phi \rightarrow \Psi$  be open games.  $\mathcal{G}$  and  $\mathcal{H}$  are *equivalent*, written  $\mathcal{G} \sim \mathcal{H}$ , if there exists an isomorphism  $\alpha : \mathcal{G} \rightarrow \mathcal{H}$ . We write  $[\mathcal{G}]$  for the equivalence class of  $\mathcal{G}$  under this relation.

**Lemma 3.9.3.** Let  $\mathcal{G}, \mathcal{G}' : \Phi \rightarrow \Psi$ ,  $\mathcal{H}, \mathcal{H}' : \Psi \rightarrow \Xi$ , and  $\mathcal{K}, \mathcal{K}' : \Phi' \rightarrow \Psi'$  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 bijection between strategy sets.

### 3.10 The category of open games

That equivalence classes of open games form a category follows easily from the fact that coend lenses form a category.

**Lemma 3.10.1.** *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 isomorphism  $\beta : \Sigma_{\mathcal{G}} \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{K}}) \rightarrow (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times \Sigma_{\mathcal{K}}$ ,  $(\sigma, (\tau, \mu)) \mapsto ((\sigma, \tau), \mu)$ . Let  $\sigma \in \Sigma_{\mathcal{G}}$ ,  $\tau \in \Sigma_{\mathcal{H}}$ , and  $\mu \in \Sigma_{\mathcal{K}}$ . Then  $P_{(\mathcal{K} \circ \mathcal{H}) \circ \mathcal{G}}(\sigma, (\tau, \mu)) = P_{\mathcal{K} \circ (\mathcal{H} \circ \mathcal{G})}((\sigma, \tau), \mu)$  by associativity of composition in  $\mathbf{Lens}_{\mathcal{C}}$ . Let  $[p, k] \in \mathbb{C}(\Phi, \Upsilon)$  be a context. Then

$$\begin{aligned} & ((\sigma, (\tau, \mu)), (\sigma', (\tau', \mu'))) \in B_{(\mathcal{K} \circ \mathcal{H}) \circ \mathcal{G}}([p, k]) \\ \iff & (\sigma, \sigma') \in B_{\mathcal{G}}\left([p, k \circ P_{\mathcal{K}}(\mu) \circ P_{\mathcal{H}}(\tau)]\right) \text{ and } (\tau, \tau') \in B_{\mathcal{H}}\left([P_{\mathcal{G}}(\sigma) \circ p, k \circ P_{\mathcal{K}}(\mu)]\right) \\ & \text{and } (\mu, \mu') \in B_{\mathcal{K}}\left([P_{\mathcal{H}}(\tau) \circ P_{\mathcal{G}}(\sigma) \circ p, k]\right) \\ \iff & (((\sigma, \tau), \mu), ((\sigma', \tau'), \mu')) \in B_{\mathcal{K} \circ (\mathcal{H} \circ \mathcal{G})}([p, k]) \end{aligned}$$

□

**Theorem 3.10.2.** *If  $\mathbf{Lens}_{\mathcal{C}}$  exists, there exists a category  $\mathbf{Game}_{\mathcal{C}}$  with pairs of objects in  $\mathcal{C}$  as objects and equivalence classes of open games as morphisms.*

*Proof.* All that remains to be checked is that the identity computation defined in Example 3.6.3 is an identity morphism, and this follows from easy checks. □

### 3.11 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 3.11.1.** *Suppose we have coend lenses*

$$\begin{array}{ccccc} \Phi & \xrightarrow{l} & \Psi & \xrightarrow{m} & \Xi \\ \Phi' & \xrightarrow{l'} & \Psi' & \xrightarrow{m'} & \Xi' \end{array}$$

The following diagrams commute:

1.

$$\begin{array}{ccc} \mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{L}(-, l' \circ m')} & \mathbb{C}(\Phi, \Xi) \\ \mathbb{C}(\Phi \otimes \Phi', m \otimes m') \downarrow & & \downarrow \mathbb{C}(\Phi, m) \\ \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') & \xrightarrow{\mathcal{L}(-, l')} & \mathbb{C}(\Phi, \Psi) \end{array}$$

2.

$$\begin{array}{ccc} \mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{L}(-, l' \circ m')} & \mathbb{C}(\Phi, \Xi) \\ \mathbb{C}(l \otimes l', m \otimes m') \downarrow & & \downarrow \mathbb{C}(l, \Xi) \\ \mathbb{C}(\Psi \otimes \Psi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{L}(-, m')} & \mathbb{C}(\Psi, \Xi) \end{array}$$

3.

$$\begin{array}{ccc} \mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{R}(-, l \circ m)} & \mathbb{C}(\Phi', \Xi') \\ \mathbb{C}(\Phi \otimes \Phi', m \otimes m') \downarrow & & \downarrow \mathbb{C}(\Phi, m) \\ \mathbb{C}(\Phi \otimes \Phi', \Psi \otimes \Psi') & \xrightarrow{\mathcal{R}(-, l)} & \mathbb{C}(\Phi', \Psi') \end{array}$$

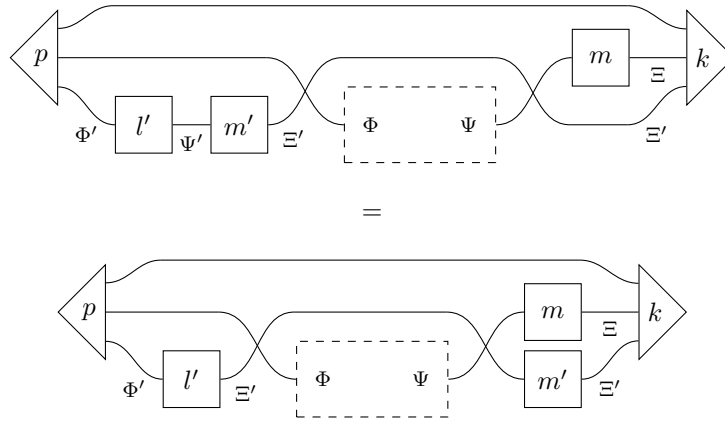
4.

$$\begin{array}{ccc} \mathbb{C}(\Phi \otimes \Phi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{R}(-, l \circ m)} & \mathbb{C}(\Phi', \Xi') \\ \mathbb{C}(l \otimes l', \Xi \otimes \Xi') \downarrow & & \downarrow \mathbb{C}(l', \Xi') \\ \mathbb{C}(\Psi \otimes \Psi', \Xi \otimes \Xi') & \xrightarrow{\mathcal{R}(-, m)} & \mathbb{C}(\Psi', \Xi') \end{array}$$

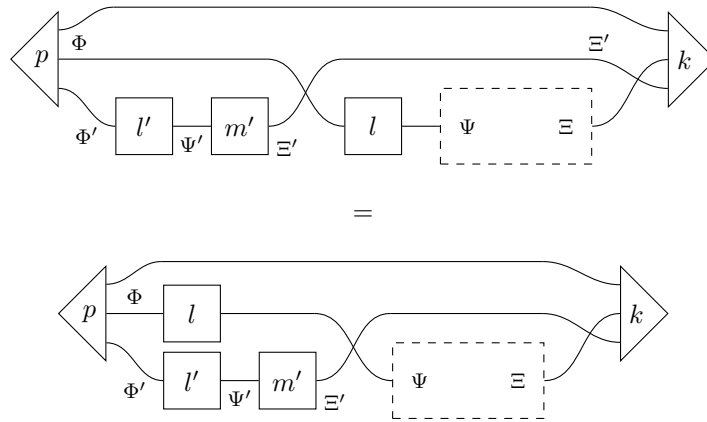
*Proof.* The four squares are given respectively by the following equalities of coend diagrams:

1.

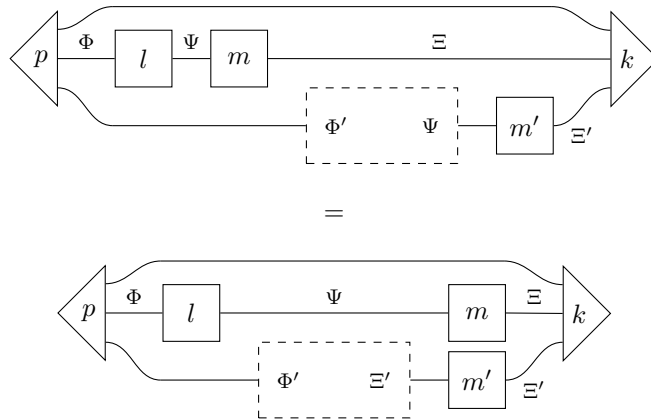




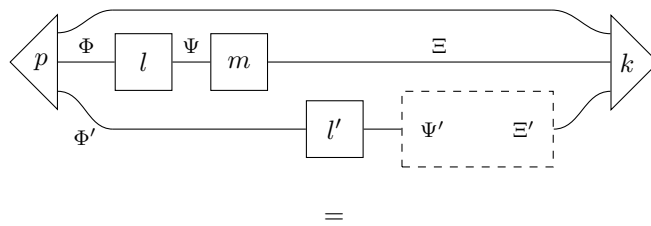
2.

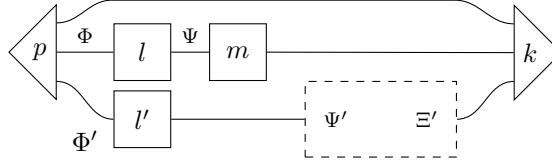


3.



4.





□

Functoriality of the tensor in  $\mathbf{Game}_C$  then follows easily.

**Corollary 3.11.2.**  $\otimes : \mathbf{Game}_C \times \mathbf{Game}_C \rightarrow \mathbf{Game}_C$  is a functor.

*Proof.* Suppose we have open games

$$\begin{array}{ccccc} \Phi & \xrightarrow{\mathcal{G}} & \Psi & \xrightarrow{\mathcal{H}} & \Xi \\ \Phi' & \xrightarrow{\mathcal{G}'} & \Psi' & \xrightarrow{\mathcal{H}'} & \Xi' \end{array}$$

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 isomorphism  $\beta : (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times (\Sigma_{\mathcal{G}'} \times \Sigma_{\mathcal{H}'}) \rightarrow (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{G}'}) \times (\Sigma_{\mathcal{H}} \times \Sigma_{\mathcal{H}'})$  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 given by  $((\sigma, \tau), (\sigma', \tau')) \mapsto ((\sigma, \sigma'), (\tau, \tau'))$ .  $\mathbf{Lens}_C$  is symmetric monoidal and, hence,

$$P_{(\mathcal{H} \circ \mathcal{G}) \otimes (\mathcal{H}' \circ \mathcal{G}')}((\sigma, \tau), (\sigma', \tau')) = P_{(\mathcal{H} \otimes \mathcal{H}') \circ (\mathcal{G} \otimes \mathcal{G}')}((\sigma, \sigma'), (\tau, \tau')).$$

Using Lemma 3.11.1,

$$\begin{aligned} & (((\sigma, \tau), (\sigma', \tau')), ((\sigma'', \tau''), (\sigma''', \tau'''))) \in B_{(\mathcal{H} \circ \mathcal{G}) \otimes (\mathcal{H}' \circ \mathcal{G}')} (c) \\ \iff & (\sigma, \sigma'') \in B_{\mathcal{G}} \left( \mathbb{C}(\Phi, P_{\mathcal{H}}(\tau)) \circ \mathcal{L}(-, P_{\mathcal{H}'}(\tau') \circ P_{\mathcal{G}'}(\sigma'))(c) \right) \\ & \text{and } (\tau, \tau'') \in B_{\mathcal{H}} \left( \mathbb{C}(P_{\mathcal{G}}(\sigma), \Xi) \circ \mathcal{L}(-, P_{\mathcal{H}'}(\tau') \circ P_{\mathcal{G}'}(\sigma'))(c) \right) \\ & \text{and } (\sigma', \sigma''') \in B_{\mathcal{G}'} \left( \mathbb{C}(\Phi', P_{\mathcal{H}'}(\tau')) \circ \mathcal{R}(-, P_{\mathcal{H}}(\tau) \circ P_{\mathcal{H}}(\sigma))(c) \right) \\ & \text{and } (\tau', \tau''') \in B_{\mathcal{H}'} \left( \mathbb{C}(P_{\mathcal{G}'}(\sigma'), \Xi') \circ \mathcal{R}(i, P_{\mathcal{H}}(\tau) \circ P_{\mathcal{H}}(\sigma))(c) \right) \\ \iff & (\sigma, \sigma'') \in B_{\mathcal{G}} \left( \mathcal{L}(-, P_{\mathcal{G}'}(\sigma')) \circ \mathbb{C}(\Phi \otimes \Phi', P_{\mathcal{H}}(\tau) \otimes P_{\mathcal{H}'}(\tau'))(c) \right) \\ & \text{and } (\sigma', \sigma''') \in B_{\mathcal{G}'} \left( \mathcal{R}(-, P_{\mathcal{G}}(\sigma)) \circ \mathbb{C}(\Phi \otimes \Phi', P_{\mathcal{H}}(\tau) \otimes P_{\mathcal{H}'}(\tau'))(c) \right) \\ & \text{and } (\tau, \tau'') \in B_{\mathcal{H}} \left( \mathcal{L}(-, P_{\mathcal{H}'}(\tau')) \circ \mathbb{C}(P_{\mathcal{G}}(\sigma) \otimes P_{\mathcal{G}'}(\sigma'), \Xi \otimes \Xi')(c) \right) \\ & \text{and } (\tau', \tau''') \in B_{\mathcal{H}'} \left( \mathcal{R}(-, P_{\mathcal{H}}(\tau)) \circ \mathbb{C}(P_{\mathcal{G}}(\sigma) \otimes P_{\mathcal{G}'}(\sigma'), \Xi \otimes \Xi')(c) \right) \\ \iff & (((\sigma, \sigma'), (\tau, \tau')), ((\sigma'', \sigma'''), (\tau'', \tau'''))) \in B_{(\mathcal{H} \otimes \mathcal{H}') \circ (\mathcal{G} \otimes \mathcal{G}')} (c). \end{aligned}$$

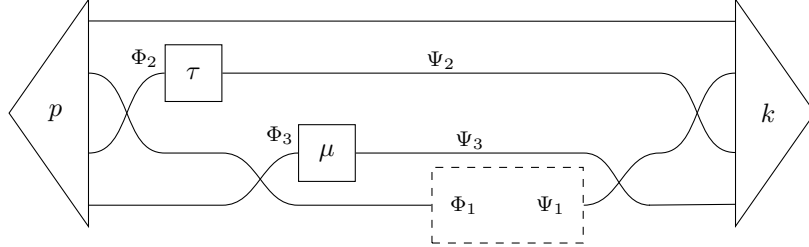
□

**Definition 3.11.3.** The structural isomorphisms in  $\mathbf{Game}_C$  are given by

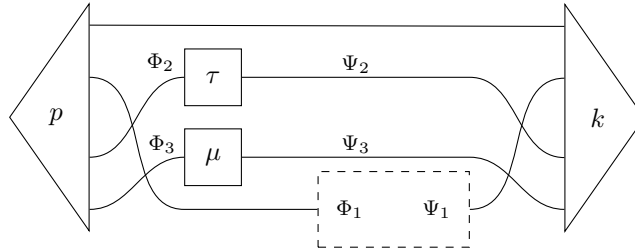
$$\begin{aligned} \alpha_{(X,A),(Y,B),(Z,C)} &= \langle \alpha_{X,Y,Z}, \alpha_{A,B,C}^{-1} \rangle \\ \rho_{(X,A)} &= \langle \rho_X, \rho_A^{-1} \rangle \\ \lambda_{(X,A)} &= \langle \lambda_X, \lambda_A^{-1} \rangle \\ s_{(X,A),(Y,B)} &= \langle s_{X,Y}, s_{B,Y} \rangle. \end{aligned}$$

**Lemma 3.11.4.** The structural isomorphisms are natural in  $\mathbf{Game}_C$ .

*Proof.* We show that the associator is natural. Naturality of the other structural maps follow by similar arguments. Let  $\mathcal{G}_i : \Phi_i \rightarrow \Psi_i$  for  $i \in \{1, 2, 3\}$ . Note that  $\Sigma_{\alpha \circ (\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3))} = (\Sigma_{\mathcal{G}_1} \times (\Sigma_{\mathcal{G}_2} \times \Sigma_{\mathcal{G}_3})) \times \{\alpha\}$  and  $\Sigma_{((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3) \circ \alpha} = \{\alpha\} \times ((\Sigma_{\mathcal{G}_1} \times \Sigma_{\mathcal{G}_2}) \times \Sigma_{\mathcal{G}_3})$ . The equivalence between  $\alpha \circ (\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3))$  and  $((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3) \circ \alpha$  will be witnessed by the isomorphism  $((\sigma, (\tau, \mu)), \alpha) \mapsto (\alpha, ((\sigma, \tau), \mu))$ . Let  $\sigma \in \Sigma_{\mathcal{G}_1}, \tau \in \Sigma_{\mathcal{G}_2}, \mu \in \Sigma_{\mathcal{G}_3}$ , and  $[p, k] \in \mathbb{C}((\Phi_1 \otimes (\Phi_2 \otimes \Phi_3)), ((\Psi_1 \otimes \Psi_2) \otimes \Psi_3))$ . 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  $\alpha \circ (\mathcal{G}_1 \otimes (\mathcal{G}_2 \otimes \mathcal{G}_3))$  and by



in  $((\mathcal{G}_1 \otimes \mathcal{G}_2) \otimes \mathcal{G}_3) \circ \alpha$ . 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.  $\square$

**Theorem 3.11.5.** *Game $_{\mathcal{C}}$  is symmetric monoidal.*

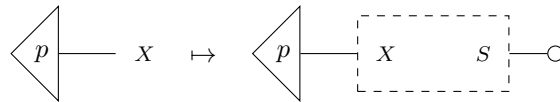
*Proof.* All that remains to be shown is that the Mac Lane pentagon and triangle axioms are satisfied, but this follows easily as the underlying category  $\mathcal{C}$  is symmetric monoidal.  $\square$

### 3.12 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 **Game $_{\mathcal{C}}$**  has a very natural game-theoretic interpretation.

**Lemma 3.12.1** ([36]). *If the monoidal unit of  $\mathcal{C}$  is terminal, then  $\mathbf{Lens}_{\mathcal{C}}(I, (X, S)) \cong \mathcal{C}(I, X)$ .*  $\square$

The isomorphism  $i : \mathcal{C}(I, X) \rightarrow \mathbf{Lens}_{\mathcal{C}}(I, (X, S))$  is explicitly given by  $p \mapsto [p, !_s]$ . In a diagram,



The following fact appears as [27, exercise 1.13]; thanks to Guillaume Boisseau and Amar Hadzihasanovic for the discussion at [39].

**Lemma 3.12.2.** *If  $F \dashv U : \mathcal{D} \rightarrow \mathcal{C}$  is any adjunction and  $G : \mathcal{D}^{\text{op}} \times \mathcal{C} \rightarrow \mathbf{Set}$  any functor, then*

$$\int^{C \in \mathcal{C}} G(F(C), C) \cong \int^{D \in \mathcal{D}} G(D, U(D)).$$

*Proof.*

$$\begin{aligned}
\int^{C \in \mathcal{C}} G(F(C), C) &\cong \iint^{C \in \mathcal{C}, D \in \mathcal{D}} G(D, C) \times \mathcal{D}(F(C), D) && \text{(ninja Yoneda lemma)} \\
&\cong \iint^{D \in \mathcal{D}, C \in \mathcal{C}} G(D, C) \times \mathcal{C}(C, U(D)) && \text{(adjunction, Fubini theorem)} \\
&\cong \int^{D \in \mathcal{D}} G(D, U(D)) && \text{(ninja Yoneda lemma)} \quad \square
\end{aligned}$$

**Lemma 3.12.3.** *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.* Let  $F : \mathcal{C} \rightarrow \mathbf{Lens}_{\mathcal{C}}$  be the embedding  $F(X) = (X, I)$ . When the monoidal unit of  $\mathcal{C}$  is terminal, this functor has a right adjoint  $U : \mathbf{Lens}_{\mathcal{C}} \rightarrow \mathcal{C}$  that is given on objects by  $U(X, S) = X$ . On morphisms  $U$  is defined by the universal maps

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

induced by the dinatural (in  $A$ ) maps  $\mathcal{C}(X, A \otimes Y) \times \mathcal{C}(A \otimes R, S) \rightarrow \mathcal{C}(X, Y)$ , taking  $(v, u)$  to  $X \xrightarrow{v} A \otimes Y \xrightarrow{!A \otimes Y} I \otimes Y \cong Y$ . The adjunction  $F \dashv U$  is given by the natural isomorphism

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

In the previous lemma we take  $G : \mathbf{Lens}_{\mathcal{C}}^{\text{op}} \times \mathcal{C} \rightarrow \mathbf{Set}$  to be

$$G((\Phi, \Phi'), \Theta) = \mathcal{C}(I, \Theta \otimes X) \times \mathbf{Lens}_{\mathcal{C}}((\Phi, \Phi') \otimes (Y, R), I).$$

Then

$$\mathbb{C}((X, S), (Y, R)) \cong \int^{(\Theta, \Theta') \in \mathbf{Lens}_{\mathcal{C}}} G((\Theta, \Theta'), \Theta) \cong \int^{\Theta \in \mathcal{C}} G((\Theta, I), \Theta) \cong \mathbf{Lens}_{\mathcal{C}}((I, R), (X, Y))$$

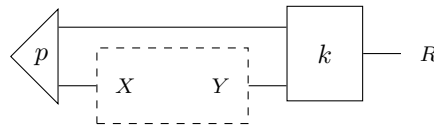
with the three isomorphisms respectively using Lemmas 3.12.1, 3.12.2 and 3.5.1.  $\square$

In the case where the monoidal unit of  $\mathcal{C}$  is terminal, the type of best response for an open game  $\mathcal{G} : (X, S) \rightarrow (Y, R)$  is equivalently

$$\mathbf{B}_{\mathcal{G}} : \mathbf{Lens}_{\mathcal{C}}((I, R), (X, Y)) \rightarrow \mathbf{Rel}(\Sigma_{\mathcal{G}}).$$

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 diagrammatic proofs. 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  $k : \Theta \otimes Y \rightarrow R$  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 section 2.

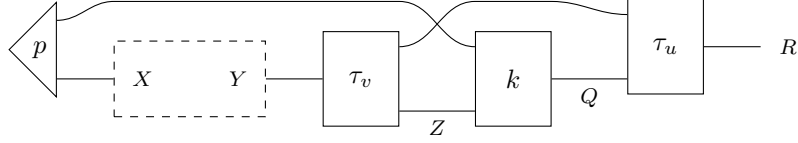
The coend diagram



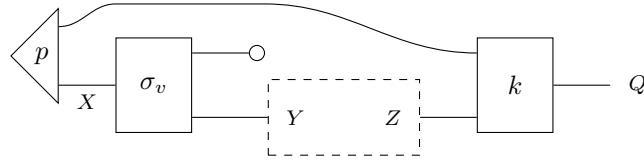
of a context  $[p, k] \in \mathbf{Lens}_{\mathcal{C}}((I, R), (X, Y))$  neatly illustrates that a context is a game state with a ‘hole’ in it. If we think of a game  $\mathcal{G} : (X, S) \rightarrow (Y, R)$  as a player in a larger game, then  $p$  corresponds to the things that have happened in the 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.

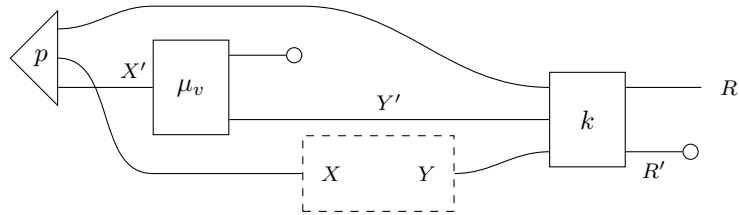
Given open games  $\mathcal{G} : (X, S) \rightarrow (Y, R)$ ,  $\mathcal{H} : (Y, R) \rightarrow (Z, Q)$ , strategies  $\sigma \in \Sigma_{\mathcal{G}}, \tau \in \Sigma_{\mathcal{H}}$ , and a context  $[p, k] \in \mathbf{Lens}_{\mathcal{C}}((I, R), (X, Z))$ , the local context for  $\mathcal{G}$  in  $\mathcal{H} \circ \mathcal{G}$  is given by



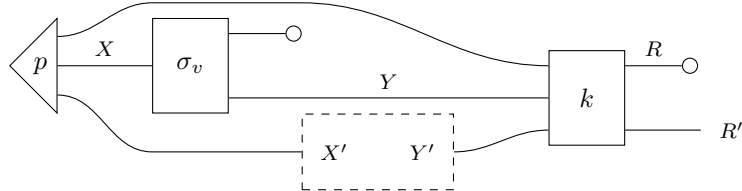
and the local context for  $\mathcal{H}$  is given by



Given another open game  $\mathcal{K} : (X', S') \rightarrow (Y', R')$ , 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.

## 4 Bayesian open games

In this section we will zero in on open games with a specific lens structure. As we will show this class of open games will address the shortcomings of concrete open games.

### 4.1 Commutative monads

Recall that a monad  $T$  over a monoidal category  $\mathcal{C}$  is *strong* if it comes with a *strength* natural transformation  $t_{A,B} : A \otimes TB \rightarrow T(A \otimes B)$  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 [36] for a much more in-depth discussion of the following result, and many more examples of when lens categories exist.

**Theorem 4.1.1** ([36]). *If  $T$  is a strong monad, then  $\mathbf{Lens}_{\mathbf{Kl}(T)}$  exists<sup>2</sup>.* □

<sup>2</sup>In [36], lenses over a Kleisli category are called *effectful optics*.

**Definition 4.1.2** (Commutative monad). Let  $T$  be a strong monad with strength  $t$  over a monoidal category  $\mathcal{C}$ . Define the *costrength* natural transformation  $t'_{A,B} : TA \otimes B \rightarrow T(A \otimes B)$  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

$$\begin{array}{ccccc} TA \otimes TB & \xrightarrow{t_{TA,B}} & T(TA \otimes B) & \xrightarrow{T(t'_{A,B})} & T^2(A \otimes B) \\ t'_{A,TB} \downarrow & & & & \downarrow \mu \\ T(A \otimes TB) & \xrightarrow{T(t_{A,B})} & T^2(A \otimes B) & \xrightarrow{\mu} & T(A \otimes B) \end{array}$$

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 for free with the monoidal tensor  $\otimes$  (on objects) and unit being the same as in the underlying category  $\mathcal{C}$ .

**Lemma 4.1.3** ([35]). *If  $T$  is a commutative monad over a symmetric monoidal category  $\mathcal{C}$ , then  $\mathbf{Kl}(T)$  is symmetric monoidal.*  $\square$

Commutative monads over **Set** also come with canonical copy/delete comonoid structures for every object. Copying  $c_X : X \rightarrow T(X \times X)$  is given by

$$X \xrightarrow{\Delta} X \times X \xrightarrow{\eta} T(X \times X)$$

and deleting  $d_X : X \rightarrow I$  is given by

$$X \xrightarrow{!} \{\star\} \xrightarrow{\eta} T(\{\star\}).$$

From this comonoid structure we obtain canonical projections

$$X \otimes Y \xrightarrow{\text{id} \otimes d} X \otimes I \xrightarrow{\rho} X$$

and

$$X \otimes Y \xrightarrow{d \otimes \text{id}} I \otimes Y \xrightarrow{\lambda} Y.$$

Crucially, it is *not* guaranteed that the monoidal tensor of  $\mathbf{Kl}(T)$  is cartesian.

## 4.2 The category of sets and random functions

We now turn to the category of interest for this section.

The *finitary distribution monad*  $D : \mathbf{Set} \rightarrow \mathbf{Set}$  maps a set  $X$  to the set of finitary probability distributions on  $X$  (finitary in the sense that only finitely many elements are assigned non-zero probability).

**Definition 4.2.1** (Finitary distribution monad). Define  $D : \mathbf{Set} \rightarrow \mathbf{Set}$  by

$$D(X) = \left\{ \alpha : X \rightarrow [0, 1] \mid \text{supp}(\alpha) < \aleph_0, \sum_{x \in \text{supp}(\alpha)} \alpha(x) = 1 \right\}$$

where  $\text{supp}(\alpha)$  is  $\{x \in X \mid \alpha(x) \neq 0\}$ , the *support* of  $\alpha$ .  $D$  acts on morphisms by

$$D(f : X \rightarrow Y)(\alpha : D(X))(y) = \sum_{f(x)=y} \alpha(x).$$

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  $f^\dagger : D(X) \rightarrow D(Y)$  of  $f : X \rightarrow D(Y)$  is

$$f^\dagger(\alpha) = \sum_{x \in \text{supp}(\alpha)} f(x)(y).$$

**Lemma 4.2.2.**  $D$  is a commutative monad. □

**Corollary 4.2.3.**  $\mathbf{Kl}(D)$  is symmetric monoidal with canonical copy/delete comonoids and projection maps.

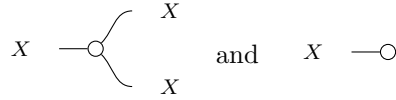
The monoidal tensor in  $\mathbf{Kl}(D)$ , which represents joint distributions, is *not* cartesian.

**Definition 4.2.4** (Copy/delete comonoid for  $\mathbf{Kl}(D)$ ). The copying map in  $\mathbf{Kl}(D)$  is given explicitly by  $c_X : X \rightarrow D(X \times X)$  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  $d_X : X \rightarrow D(\{\star\})$  must be given by  $d(x)(\star) = 1$ . As this map is unique, we refer to it as  $!$ .

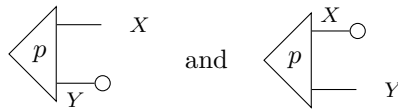
Whenever we are working in  $\mathbf{Kl}(D)$  we denote  $c_X$  and  $!_X$  by



respectively.

The canonical projections in  $\mathbf{Kl}(D)$  correspond to taking marginals.

**Definition 4.2.5** (Marginals). Let  $p : D(X \times Y)$  be a joint distribution. The *left marginal*  $p_X : D(X)$  is given by  $p_X(x) = \sum_{y \in \text{supp}(p(x, -))} p(x, y)$ . The *right marginal*  $p_Y : D(Y)$  is given similarly by  $p_Y(y) = \sum_{x \in \text{supp}(p(-, y))} p(x, y)$ . As diagrams, these are given by



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 4.2.6** (Update operator). Let  $X$  and  $\Theta$  be sets. We think of  $\Theta$  as a type which an agent has a probabilistic belief about, and  $X$  as a type that will be observed by an agent. Define the *update operator*  $\mathcal{U}_\Theta : D(\Theta \times X) \rightarrow (X \rightarrow D(\Theta))$  by

$$\mathcal{U}_\Theta(p)(x)(\vartheta) = \frac{p(\vartheta, x)}{\sum_{(\vartheta', x) \in \text{supp}(p)} p(\vartheta', x)}.$$

We also write  $p(\vartheta|x)$  for  $\mathcal{U}_\Theta(p)(x)(\vartheta)$ .

**Lemma 4.2.7.** *The update operator is natural in  $\Theta$ . That is, the following diagram commutes for any  $f : \Theta_1 \rightarrow \Theta_2$ :*

$$\begin{array}{ccccc}
 D(\Theta_1 \times X) & \xrightarrow{\mathcal{U}_{\Theta_1}} & X & \rightarrow & D(\Theta_1) \\
 \downarrow D(f \times X) & & & & \downarrow D(f) \circ - \\
 D(\Theta_2 \times X) & \xrightarrow{\mathcal{U}_{\Theta_2}} & X & \rightarrow & D(\Theta_2).
 \end{array}$$

*Proof.* Let  $p \in D(\Theta_1 \times X)$ ,  $x \in X$ ,  $\vartheta_1 \in \Theta_1$ , and  $\vartheta_2 \in \Theta_2$ . The top of the square is given by

$$\begin{aligned}
 (D(f) \circ \mathcal{U}_{\Theta_1})(p)(x)(\vartheta_2) &= \sum_{f(\vartheta_1)=\vartheta_2} \mathcal{U}_{\Theta_1}(p)(x)(\vartheta_1) \\
 &= \frac{\sum_{f(\vartheta_1)=\vartheta_2} p(\vartheta_1, x)}{\sum_{\vartheta'_1 \in \text{supp}(p(-, x))} p(\vartheta'_1, x)}. \tag{*}
 \end{aligned}$$

The bottom of the square is given by

$$\begin{aligned}
 \mathcal{U}_{\Theta_2}(D(f \times X)(p))(x)(\vartheta_2) &= \frac{D(f \times X)(p)(\vartheta_2, x)}{\sum_{\vartheta'_2} D(f \times X)(p)(\vartheta'_2, x)} \\
 &= \frac{\sum_{f(\vartheta_1)=\vartheta_2} p(\vartheta_1, x)}{\sum_{\vartheta'_2} \sum_{f(\vartheta'_1)=\vartheta'_2} p(\vartheta'_1, x)}. \tag{**}
 \end{aligned}$$

The result follows, noting that the denominators of (\*) and (\*\*) are equal.  $\square$

### 4.3 Bayesian open games

**Definition 4.3.1** (Bayesian open game). A *Bayesian open game* is a morphism in  $\mathbf{Game}_{\mathbf{Kl}(D)}$ . Explicitly, a Bayesian open game  $\mathcal{G} : (X, S) \rightarrow (Y, R)$  is given by

1. A set of strategies  $\Sigma$ ,
2. A play function

$$P : \Sigma \rightarrow \int^{\Theta:\mathbf{Set}} (X \rightarrow D(\Theta \times Y)) \times ((\Theta \times R) \rightarrow D(S)),$$

3. A best-response function

$$B : \mathbf{Lens}_{\mathbf{Kl}(D)}((I, R), (X, Y)) \rightarrow \text{Rel}(\Sigma)$$

We refer to atoms in the category of Bayesian open games as *Bayesian atoms*.

We unpack the definition of the play function to emphasise that, when we wish to actually specify a Bayesian open game, it is usually easier to specify  $P(\sigma)$  as the equivalence class of a pair of morphisms.



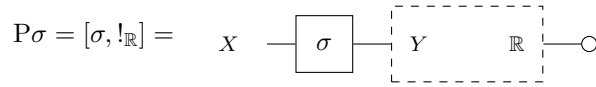
#### 4.4 Bayesian agents

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 to a new posterior;
3. Attempt to maximise their expected utility given their posterior.

**Definition 4.4.1** (Bayesian agent). Let  $X, Y$  be sets. The *Bayesian agent*  $\mathcal{A}_{(X,Y)} : (X, I) \rightarrow (Y, \mathbb{R})$  is the Bayesian atom given by

1.  $\Sigma_{\mathcal{A}} = X \rightarrow D(Y)$ ,
- 2.



3. The selection function  $\varepsilon : \mathbf{Lens}_{\mathbf{Kl}(D)}((I, \mathbb{R}), (X, Y)) \rightarrow \mathcal{P}(X \rightarrow D(Y))$  is given by

$$\varepsilon_{\mathcal{A}}([p, k]) = \left\{ \sigma : X \rightarrow D(Y) \mid \forall x \in \text{supp}(p_2), \right. \\ \left. \sigma(x) \in \arg \max_{\alpha \in D(Y)} \left( \mathbb{E} \left[ k^\dagger(\mathcal{U}_{\Theta}(p)(x), \alpha) \right] \right) \right\}.$$

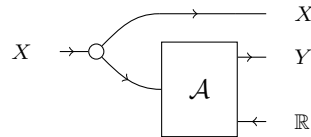
It is worth explaining this last formula in words.  $\mathcal{A}_{(X,Y)}$  represents an agent choosing an element of  $Y$  after observing an element of  $X$ . The context of the decision consists of a set  $\Theta$  of unobservable states, a prior joint distribution  $p : D(\Theta \times X)$  on unobservable and observable states, and a utility function  $k : \Theta \times Y \rightarrow D(\mathbb{R})$  that depends on the unobservable state and the agent's choice. The optimality condition says that for all observations  $x$  that the agent could make with nonzero probability, the strategy  $\sigma(x)$  maximises the expected value of  $k(\vartheta', -)$ , where  $\vartheta' = \mathcal{U}_{\Theta}(p)(x)$  is the posterior distribution on the unobservable state given the observation of  $x$ .

**Lemma 4.4.2.** *The selection 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 (4.2.7).  $\square$

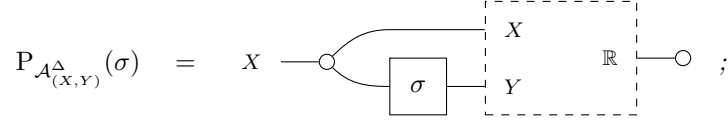
In the next definition we formalise the idea that a player in a game might be assigned a (game-theoretic) type on which their utility function depends. We can do this simply using a Bayesian agent and a copying computation.

**Definition 4.4.3.** Let  $\mathcal{A}_{(X,Y)} : (X, I) \rightarrow (Y, \mathbb{R})$  be a Bayesian agent. Define  $\mathcal{A}_{(X,Y)}^{\Delta} : (X, I) \rightarrow (X \times Y, \mathbb{R})$  to be the Bayesian open game  $(\text{id}_{(X,I)} \otimes \mathcal{A}_{(X,Y)}) \circ \Delta_X$ , or as a string diagram,

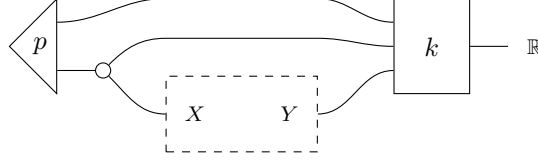


**Lemma 4.4.4.**  $\mathcal{A}_{(X,Y)}^{\Delta}$  is explicitly given, up to isomorphism, by

1.  $\Sigma_{\mathcal{A}_{(X,Y)}^{\Delta}} = X \rightarrow D(Y)$ ;
2.  $P_{\mathcal{A}_{(X,Y)}^{\Delta}} : \Sigma_{\mathcal{A}_{(X,Y)}^{\Delta}} \rightarrow \mathbf{Lens}_{\mathbf{Kl}(D)}((X, I), (X \times Y, \mathbb{R}))$  is given by



3. Let  $[p, k] \in \mathbf{Lens}_{\mathbf{Kl}(D)}((I, \mathbb{R}), (X, X \times Y))$ . Best response is given, up to isomorphism, by  $B_{\mathcal{A}_{(X,Y)}^\Delta}([p, k]) = B_{\mathcal{A}_{(X,Y)}}([p, k]')$  where  $[p, k]'$  is the context given by



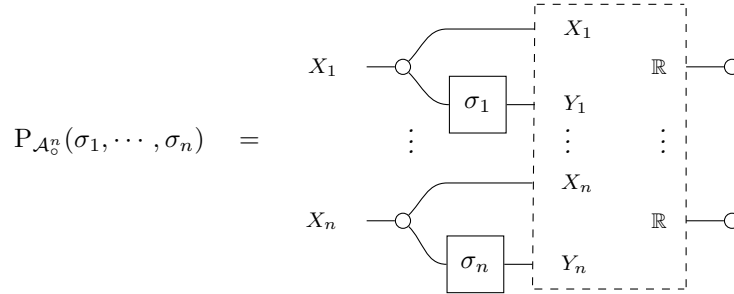
*Proof.* This result follows from definition chasing, noting that  $\mathcal{A}_{(X,Y)}$  is the only component with non-trivial strategy profile set.  $\square$

**Lemma 4.4.5.** Let  $\mathcal{A}_{(X_i, Y_i)}$  be Bayesian agents for  $i \in \{1, \dots, n\}$ . Then  $\mathcal{A}^{\Delta^n} = \bigotimes_{i=1}^n \mathcal{A}_{(X_i, Y_i)}^\Delta$  is explicitly given as follows.

1.  $\Sigma_{\mathcal{A}^{\Delta^n}} = \prod_{i=1}^n (X_i \rightarrow D(Y_i))$ ;
2. The play function

$$P_{\mathcal{A}^{\Delta^n}} : \Sigma_{\mathcal{A}^{\Delta^n}} \rightarrow \mathbf{Lens}_{\mathbf{Kl}(D)} \left( \left( \prod_{i=1}^n X_i, I \right), \left( \prod_{i=1}^n (X_i \times Y_i), \mathbb{R}^n \right) \right)$$

is given by



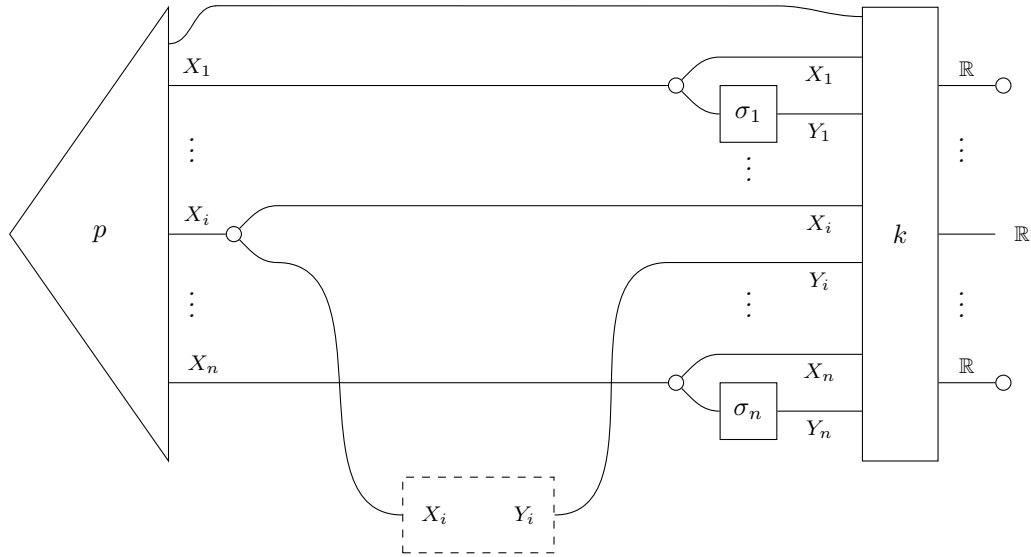
3. Let  $[p, k] \in \mathbf{Lens}_{\mathbf{Kl}(D)}((I, \mathbb{R}^n), (\prod_i X_i, \prod_i X_i \times Y_i))$ . Best response

$$B_{\mathcal{A}^{\Delta^n}}([p, k])(\sigma_1, \dots, \sigma_n)$$

is given, up to isomorphism, by

$$\prod_{i=1}^n \left\{ \sigma_i : X_i \rightarrow D(Y_i) \mid \forall x_i \in \text{supp}(p^{\sigma^{-i}}), \right. \\ \left. \sigma_i(x_i) \in \arg \max_{\alpha_i \in D(Y_i)} \left( \mathbb{E} [k^{\sigma^{-i}}(\mathcal{U}_{\Theta^{-i}}(p^{\sigma^{-i}})(x_i), \alpha_i)] \right) \right\}$$

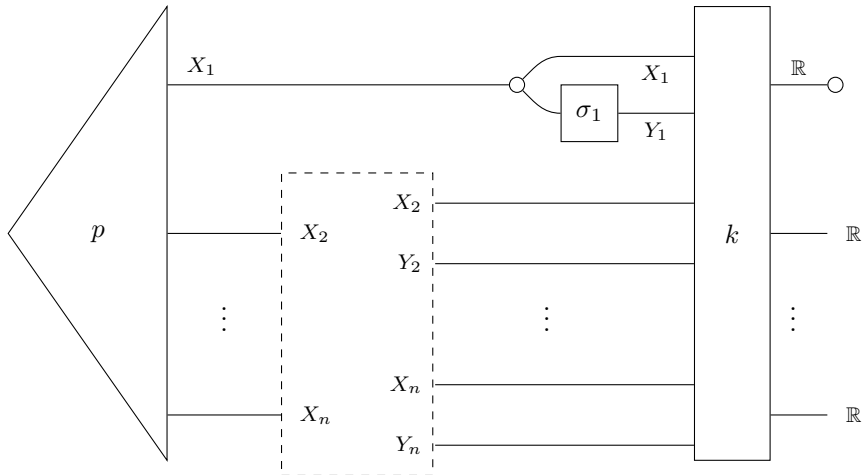
where  $[p^{\sigma^{-i}}, k^{\sigma^{-i}}]$  is the context given by



*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 4.4.4 serves as the base case ( $n = 1$ ) for an induction argument. The result then follows easily by considering that

$$B_{\mathcal{A}^{\Delta^n}}([p, k])(\sigma_1, \dots, \sigma_n) = B_{\mathcal{A}_{(X_1, Y_1)}^{\Delta}}([p, k]_1)(\sigma_1) \times B_{\bigotimes_{i=2}^n \mathcal{A}_{(X_i, Y_i)}^{\Delta}}([p, k]_{-1})(\sigma_{-1})$$

and applying the inductive hypothesis, where  $[p, k]_{-1}$  is the context

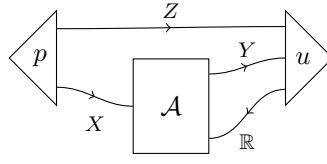


□

### 4.5 Decisions under risk

In this section we introduce another type of situation involving a Bayesian agent that can be modelled 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 are represented by Bayesian open games of the form

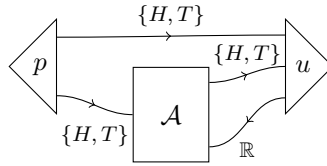


in which an agent  $\mathcal{A}$  attempts to maximise their outcome which is, in part, dependent on the type  $Z$  which  $\mathcal{A}$  does not observe.

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 posterior.

**Example 4.5.1** (Biased coin). 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 applies 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

$$\begin{aligned} u : \{H, T\}^2 &\rightarrow D\mathbb{R} \\ (x, y) &\mapsto \begin{cases} \delta_1 & \text{if } x = y \\ \delta_0 & \text{otherwise.} \end{cases} \end{aligned}$$

Explicitly, the game is given by  $\mathcal{G} := u \circ (\mathcal{A} \otimes \text{id}_{\{H, T\}}) \circ p$ . Note that  $\Sigma_{\mathcal{G}} \cong \Sigma_{\mathcal{A}} = \{H, T\} \rightarrow D(\{H, T\})$ . Also note that there is precisely one context for  $\mathcal{G}$  since its type is  $I \rightarrow I$  (that is,  $(1, 1) \rightarrow (1, 1)$ ) and, moreover, as the best-response functions for  $u$ ,  $\text{id}_{\{H, T\}}$ , and  $p$  are trivial, the best-response function for  $\mathcal{G}$  is isomorphic to the constant relation

$$\begin{aligned} \text{B}_{\mathcal{A}}([p, k]) &= \left\{ \sigma : \{H, T\} \rightarrow D(\{H, T\}) \mid \forall x \in \{H, T\}, \right. \\ &\quad \left. \sigma(x) \in \arg \max_{\alpha \in D(\{H, T\})} \mathbb{E}[k(\mathcal{U}_{\{H, T\}}(p)(x), \alpha)] \right\}. \end{aligned}$$

The posterior  $\mathcal{U}_{\{H,T\}}(p)(H)(H)$  is given by

$$\begin{aligned}\mathcal{U}_{\{H,T\}}(p)(H)(H) &= \frac{p(H, H)}{\sum_{(\vartheta, H) \in \text{supp}(p)} p(\vartheta, H)} \\ &= \frac{5}{8}\end{aligned}$$

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

$$\begin{aligned}\arg \max_{\alpha} \left( \mathbb{E}[k^{\dagger}(\mathcal{U}_{\{H,T\}}(p)(H), \alpha)] \right) &= \{\delta_H\} \\ \arg \max_{\alpha} \left( \mathbb{E}[k^{\dagger}(\mathcal{U}_{\{H,T\}}(p)(T), \alpha)] \right) &= \{\delta_T\}.\end{aligned}$$

Hence  $B_{\mathcal{A}}([p, k])$  is a singleton set containing the strategy  $\sigma$  where  $\sigma(H) = \delta_H$  and  $\sigma(T) = \delta_T$ , as expected.

## 4.6 Relating Bayesian Open Games to Bayesian Games

The example given in the previous section illustrates two important improvements over the version of open games as introduced in [13]. There, everything is deterministic – the environment, the players moves etc. Here, the environment as well as the players' moves can be probabilistic.

Still, we only consider a single player. We could consider examples with more players and with probabilistic behaviour induced by nature or games where probabilistic behaviour is key, such as matching pennies. In [15] a definition of open games is given that can handle mixed strategies.

Instead, we want to focus on another aspect of Bayesian open games, which we believe is much less obvious, particularly to readers with no game theory background: From a game-theoretic perspective, we introduced a new solution concept. For random environments and probabilistic moves, there is no real difference to standard Nash equilibria. However, there are important classes of games for which the new equilibrium notion brings material change.

Reconsider Example 4.5.1. Here, the agent has some prior information about the nature of the coin. Given observations from the coin and the prior information, the agent updates beliefs about the true nature of the coin and maximises accordingly. Thus, we maintain not only an assumption that the agent maximises but also how she deals with information.

The example is very simple. After all, there is only one agent. Our interest lies of course in the interaction of such Bayesian agents. While the strategic reasoning is more complicated in such situations the overall structure is the same as in Example 4.5.1: Each agent has access to some prior information, some (partial) information is revealed, and the agent makes a decision.

In the game-theory literature such a game is called a ‘Bayesian game’ or also ‘game in Bayesian form’ and the solution concept we sketched is called ‘Bayesian Nash Equilibrium’. Next, we introduce the standard economic notion of such games, then provide a simple example, an auction, and followed by a general construction that allows us to translate an arbitrary Bayesian game into a Bayesian open game.

### 4.6.1 Games in Bayesian form

Denote with  $\Gamma^B = (N, (A_i)_{i=1}^N, (\Theta_i)_{i=1}^N, \pi, (u_i)_{i=1}^N)$  a Bayesian game. Before we explain the components we add some useful notation: we write  $\Theta_{-i} = \prod_{j \in N, j \neq i} \Theta_j$  for the possible combinations of  $\Theta_j$  other than  $i$ .

- $N$  denotes the number of players.
- $A_i$  defines a set of actions available to the players.<sup>3</sup>
- $\Theta_i$  defines the a set of (player) types each player  $i$  can have.

<sup>3</sup>Note, we intentionally refer to this as actions and not as strategies. This will be clear once we compare the Bayesian game to a normal-form game.

- $\pi : D\left(\prod_{i=1}^N \Theta_i\right)$  is a joint prior distribution on the types that is common knowledge.
- $u_i : \prod_{i=1}^N \Theta_i \times \prod_{i=1}^N A_i \rightarrow \mathbb{R}$  defines a utility for any profile of player types and any profile of actions.

This definition is rather cryptic at first. So let us try to dissect its components. Then we will compare it to the definition of a normal form which we introduced in 2.12.4. This hopefully further helps to make sense of it.

Let us begin with  $\Theta_i$ , the player type, which is probably most obscure and most central at the same time. One way to think of types is imagining player  $i$  having different realisations or versions, one of which ends up playing the game. Each realisation, or type, summarises information relevant for the game. Often, this information will concern the payoffs.<sup>4</sup> For example, player  $i$  may consider to bid in an auction. His type determines his evaluation for the good, that is, how much he values owning the good. It is easily conceivable that  $i$  may have different evaluations before the auction really takes place.

This example also illustrates another aspect of this structure. While player  $i$  may have different types, before he has to choose his action, he typically will know his type. So, once the bidding starts, he knows how much he values the good. But crucially, he does not know the types of the other players, only a probability distribution – conditional on his own observed type. The last aspect also already alludes to the role of Bayesian updating: Given his own observed type, a player may refine his belief about the other players' types.

The payoff function  $u_i$ , as with a normal form, maps choices of players into a payoff. Here, however, player  $i$ 's type may affect the payoff. Think again about an auction, naturally the bidding behaviour will affect  $i$ 's payoff. But so does his type, the evaluation for the good. Note that the definition above includes the possibility that the whole realisation of types affects  $i$ 's payoff. In the case of bidding for a private good, only  $i$ 's type will typically be relevant.

Lastly,  $A_i$  represents the set of actions player  $i$  can take in the game. Note, that we intentionally refer to this as *actions* and not as strategies. To understand this it is best to compare the above definition to a normal-form game.

Recall its notation:  $\Gamma = (N, (S_i)_{i=1}^N, (u_i)_{i=1}^N)$ . Here,  $S_i$  refers to a strategy which is a complete contingent plan for all possible occasions where  $i$  can make a move. Note that  $S_i$  can be a shorthand for some complicated dynamic structure where  $i$  moves several times.

Analogously,  $A_i$  in the Bayesian game refers to a complete contingent plan *once* the game 'begins', i.e. after  $i$  has learned his type. This could involve a complicated dynamic structure. Crucially, however, this is not the same as a strategy for the Bayesian game. Why? A strategy of the Bayesian game must include a contingent plan also for each realisation of  $i$ 's type!

Formally, a pure strategy for a game in Bayesian form is a mapping  $s_i : \Theta_i \rightarrow A_i$  for all  $\Theta_i$ . A behavioural strategy  $\sigma_i$  for player  $i$  is a mapping  $\sigma_i : \Theta_i \rightarrow D(A_i)$ , i.e for each type of player  $i$  a behavioural strategy assigns a probability distribution over the available actions.<sup>5</sup>

For each player we can now define the (conditional) expected utility for player  $i$  given behavioural strategy profile  $\sigma$  with

$$Eu_i(\sigma | t_i) \equiv \sum_{t_{-i} \in \Theta_{-i}} p_\pi(t_{-i} | t_i) u_i((t_i, t_{-i}), \sigma)$$

Note, that  $p_\pi(\cdot | \cdot)$  updates the prior information  $\pi$  for player  $i$  given Bayes' rule. Equipped with all that we can finally define a Bayesian Nash equilibrium.

<sup>4</sup>The concept of a type is much more expressive. It can express all sorts of uncertainty with respect to a game. We will not discuss this here as it would lead us too far astray. Compare, for instance, Chapters 9-11 in [29] for more background on the notion of type.

<sup>5</sup>In the context of the a game in Bayesian form, similar to a game in normal form, it is more common to use *mixed strategies* of type  $D(S_i) = D(\Theta_i \rightarrow A_i)$ , i.e. distributions on pure strategies. However, as any game in Bayesian form can be equivalently modelled as a game in extensive form, and both, behavioural and mixed strategies lead to the same equilibria (under mild assumptions automatically fulfilled in our setup), the difference is immaterial. The reason for our choice of using behavioural strategies is that open games share properties from both standard formulations and it is easier to connect behavioural strategies to our formulation. More on that in Section 4.7.

**Definition 4.6.2** (Bayesian equilibrium). A (behavioural) strategy profile  $\sigma^* = (\sigma_1^*, \sigma_2^*, \dots, \sigma_N^*)$  is a Bayesian Nash equilibrium if for each player  $i \in N$ , each type  $\vartheta_i \in \Theta_i$ , and each possible action  $a_i \in A_i$  it holds that:

$$Eu_i(\sigma^* \mid \vartheta_i) \geq Eu_i(a_i, \sigma_{-i} \mid \vartheta_i)$$

#### 4.6.3 An auction example

Suppose two agents are bidding for a good that is being sold in a first-price sealed-bid auction. Here ‘first-price’ means that whichever agent bids higher receives the good and pays her/his own bid, with the other agent neither gaining nor losing anything. ‘Sealed-bid’ means neither player can observe the other’s bid, meaning the bidding is effectively simultaneous.

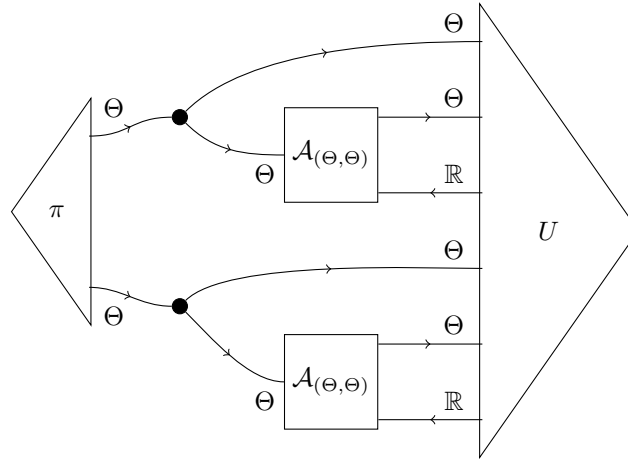
Both players have a private valuation for the good, which are drawn from a joint random distribution  $\pi : D(\Theta \times \Theta)$ , where  $\Theta \subseteq [0, \infty)$  is the range of possible values. As usual, we assume this prior is common knowledge. One consequence thereof is that a player knowing their own valuation can update their beliefs about the other’s valuation. Obviously, whether there is something to learn from one’s own valuation about others’ valuations depends on the type of good being auctioned. For instance, if we are bidding on a construction contract, then the value that contract has for me will be correlated with your evaluation. If, however, we are bidding on a painting by some obscure painter, it is less clear that I can learn something about others’ valuations.

The winning bidder’s utility is given by their valuation of the good minus the bid that they must pay. (This can be negative if the bid is higher than the valuation.) We must also choose a way to resolve a tie if the bids are equal: we assume that the winner is determined by a fair coin flip. Thus the expected utilities, for private valuations  $\vartheta_i \in \Theta$  and bids  $b_i \in \Theta$ , are given by

$$U(\vartheta_1, b_1, \vartheta_2, b_2) = \begin{cases} (\vartheta_1 - b_1, 0) & \text{if } b_1 > b_2 \\ (0, \vartheta_2 - b_2) & \text{if } b_1 < b_2 \\ (\frac{\vartheta_1 - b_1}{2}, \frac{\vartheta_2 - b_2}{2}) & \text{if } b_1 = b_2. \end{cases}$$

In order to formalise this situation as a Bayesian game as defined in the previous section, we take  $N = 2$ ,  $A_1 = A_2 = \Theta$ , and  $\pi$ .

The Bayesian open game describing this situation is given by the string diagram



As a morphism  $\mathcal{G} : I \rightarrow I$  in the category of Bayesian open games, this has  $\Sigma_{\mathcal{G}} = (\Theta \rightarrow D(\Theta)) \times (\Theta \rightarrow D(\Theta))$ , the set of behavioural strategy profiles for the auction, and  $B_{\mathcal{G}}(*)$  the best-response relation for behavioural strategy profiles.

#### 4.6.4 General construction for games in Bayesian form

Generalising the previous example, we can give a construction for converting any game in Bayesian form into a Bayesian open game of type  $I \rightarrow I$  that has the same best-response relation.

Consider a general Bayesian game  $(N, (A_i), (\Theta_i), \pi, U)$ . For each player  $i$ , recall the copying agent  $\mathcal{A}_{(\Theta_i, A_i)}^\Delta : (\Theta_i, 1) \rightarrow (\Theta_i \times A_i, \mathbb{R})$ . Their tensor product has type

$$\bigotimes_{i=1}^N \mathcal{A}_{(\Theta_i, A_i)}^\Delta : \left( \prod_{i=1}^N \Theta_i, 1 \right) \rightarrow \left( \prod_{i=1}^N (\Theta_i \times A_i), \mathbb{R}^N \right).$$

The full game is given by

$$I \xrightarrow{\langle \pi, 1 \rangle} \left( \prod_{i=1}^N \Theta_i, 1 \right) \xrightarrow{\bigotimes_{i=1}^N \mathcal{A}_{(\Theta_i, A_i)}^\Delta} \left( \prod_{i=1}^N (\Theta_i \times A_i), \mathbb{R}^N \right) \xrightarrow{\langle U, \text{id}_{\mathbb{R}^N} \rangle} (\mathbb{R}^N, \mathbb{R}^N) \xrightarrow{\varepsilon_{\mathbb{R}^N}} I.$$

**Proposition 4.6.5.** *Let  $G = (N, (A_i), (T_i), \pi, U)$  be a game Bayesian form, and let*

$$c = (1, \pi, U) : \mathbb{C} \left( \left( \prod_{i=1}^N T_i, 1 \right), \left( \prod_{i=1}^N (T_i \times A_i), \mathbb{R} \right) \right).$$

*Then  $B_{\bigotimes_{i=1}^N \mathcal{A}_{(T_i, A_i)}^\Delta}(c)$  is the best-response relation for behavioural strategy profiles of  $G$ .*

The main part of the proof of this result is contained in Lemma 4.4.5, which characterises the tensor product of copying agents.

## 4.7 Extensive form

The normal-form representation as well as the representation of Bayesian Games really are strategy-centric. They both condense the strategic situation in a way that makes them most amenable to analysis from the point of view of a specific equilibrium concept such as Nash/Bayesian Nash equilibrium.

Open games have baked in a notion of equilibrium and thus certainly share this focus. And for that reason, comparisons to classical game theory in this paper have been limited to these representations.

In classical game theory, an alternative to the normal-form representation of a game is its *extensive form*. This representation can also be used for games in Bayesian form. We will not give a definition here, as standard definitions are cumbersome and will not add much value. What is important though is what the extensive form represents: it provides a detailed account about which player knows what at which point and what actions he has then available. One can think of the extensive form as a detailed account of how a game and the players' information unfolds. Thus, it provides much more information than its corresponding normal form, to which each extensive form can be reduced.

For instance, one reason to consider the extensive form representation of a game is that it explicitly captures temporal information, say, if one player moves before the other player.<sup>6</sup>

Open games as a representation of a strategic situation have much in common with its extensive form representation. After all they provide an expressive language to represent a detailed account of a strategic interaction. In the following, we will make the relationship between Bayesian open games as presented in this paper and the extensive form representation of Bayesian games more concrete.

We begin with a concrete example that illustrates how open games deal with sequential moves of players *in the presence of asymmetric information*.

### 4.7.1 The market for lemons

The following is a simple version of a classic model going back to [1].<sup>7</sup> There are two players, a seller and a buyer. The seller wants to sell a used car to the buyer; he offers the buyer a price

<sup>6</sup>One other important reason to use the extensive form is to consider different, typically refined versions of Nash equilibria which take into account the dynamic dimension of games. But this goes beyond our current paper.

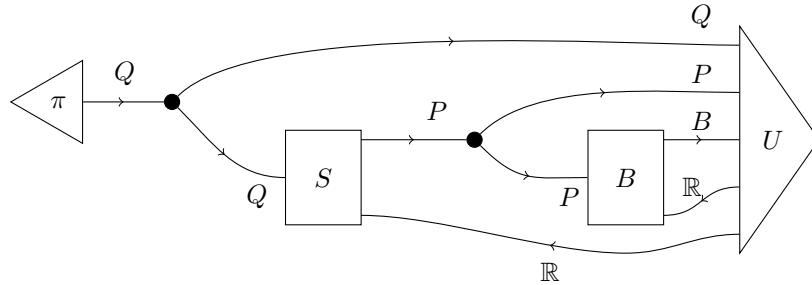
<sup>7</sup>Note that the original model is not game-theoretic in nature. See [28] Chapter 13 for a discussion and a richer game-theoretic model than presented here.



$P$ ; the buyer can either accept or reject the offer. Both players have some evaluation for the car which will affect their profit/utility.

Their evaluation depends on the quality  $Q$  of the car. It is drawn from some fixed distribution  $\pi$ . Critically, the seller observes the quality before offering a price to the buyer. The buyer, however, does not directly observe the quality.

The open game below represents this situation.



As Akerlof [1] observed, due to the asymmetry in information, a transaction may not occur even though such a transaction would be efficient (and would take place if both players were perfectly informed). The reason is simple. For each price the buyer has to take into consideration what quality of car the seller would be willing to offer. As each seller will only offer prices equal to or above his evaluation for the car, a low price indicates low quality. However, a high price does not necessarily reflect high quality; it could just mean that the seller is charging a high price for an actual ‘lemon’. Thus, the buyer, given his prior will update his estimate of the quality for each price offered. And depending on the parameters, transactions may only occur for low quality cars but not for high quality cars.

Note that, similar to our remark in 2.12.3, even though the interaction (and the representation as an open game) is sequential, the equilibrium notion is static and does not consider the timing of moves.<sup>8</sup>

#### 4.7.2 A general construction

To translate an arbitrary extensive form game to a string diagram requires an additional *external choice* operator like the one developed in [22], which is currently not well understood in the Bayesian setting. We will instead give a much simpler translation for a certain subclass of extensive form games, roughly those whose trees are perfectly balanced. Specifically we require:

- All paths from the root to a leaf have the same length
- All information sets intersect only one level of the tree (where the  $i$ th level is the set of all nodes that are  $i$  steps away from the root)
- All information sets on the same level are owned by the same player (the root node is always owned by Nature)
- All nodes on the same level have the same number of outgoing edges, labelled by the same edge labels

Any extensive form game can be converted into one satisfying these conditions that is equivalent for most game-theoretic purposes, by adding (potentially many) dummy nodes and moves leading to redundant subtrees with only strictly dominated payoffs. Obviously this is unsatisfactory, and the construction in this section should be thought of as an argument that Bayesian open games have the same ‘expressive power’ as extensive form, and not as a practical translation.

<sup>8</sup>See remark in Section 4.7. There are refinements of Bayesian Nash Equilibria which take the dynamics into account. In the context of the lemon market this is relevant if one considers extensions of that model. One prominent example is the role of costly signals that the informed side may use in order to reduce the uncertainty of the buyer. Cf. [28] Chapter 13.C.

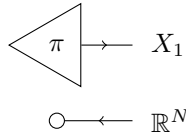
Consider an extensive form game  $G$  satisfying these conditions, with  $N > 0$  players, and  $L > 0$  ‘ordinary’ levels with choice nodes, together with level 0 (the root) and level  $L + 1$  (the leaves).

At an ordinary level  $1 \leq i \leq L$ , suppose the set of nodes is  $X_i$ , the information equivalence relation is  $\sim_i$  (and so the set of information sets at level  $i$  is  $X_i/\sim_i$ ), all information sets are owned by player  $1 \leq P_i \leq N$ , and the set of outgoing edge labels for all nodes is  $A_i$ . We summarise the connectivity between level  $i$  and level  $i + 1$  as a transition function  $q_i : X_i \times A_i \rightarrow X_{i+1}$ .

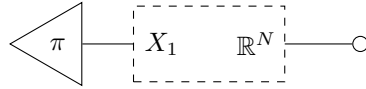
The data at the root node is a probability distribution  $\pi : D(X_1)$  on the Nature moves. Finally,  $X_{L+1}$  is the set of leaf nodes, and the assignment of payoffs at leaf nodes is summarised as a utility function  $U : X_{L+1} \rightarrow \mathbb{R}^N$ .

We translate each level of the tree into a Bayesian open game, which we then sequentially compose together. An ordinary level  $1 \leq i \leq L$  will be translated to an open game of type  $\mathcal{G}_i : (X_i, \mathbb{R}^N) \rightarrow (X_{i+1}, \mathbb{R}^N)$ , level 0 to an open game  $\mathcal{G}_0 : (1, 1) \rightarrow (X_1, \mathbb{R}^N)$  and level  $L + 1$  to an open game  $\mathcal{G}_{L+1} : (X_{L+1}, \mathbb{R}^N) \rightarrow (1, 1)$ . Thus, the final result will be an open game  $\mathcal{G} = \mathcal{G}_{L+1} \circ \dots \circ \mathcal{G}_0 : (1, 1) \rightarrow (1, 1)$ .

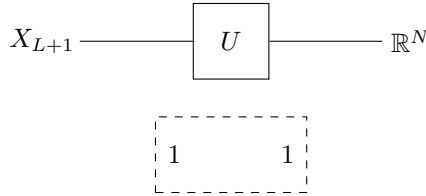
We define  $\mathcal{G}_0$  to be the computation  $\langle \pi, ! \rangle$ , where  $\pi : 1 \rightarrow D(X_1)$  is the prior considered as a state in the Kleisli category, and  $! : \mathbb{R}^N \rightarrow D(1) = 1$  is the discard map. That is to say, it is the atomic open game with  $\Sigma_{\mathcal{G}_0} = \{\pi, !\}$  and  $B_{\mathcal{G}_0}(c) = \{\pi, !\}$  for all contexts  $c \in \mathbb{C}((I, I), (X_1, \mathbb{R}^N))$ .  $\mathcal{G}_0$  is simply depicted by the string diagram



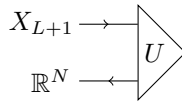
and the optic  $[\pi, !]$  by the coend diagram



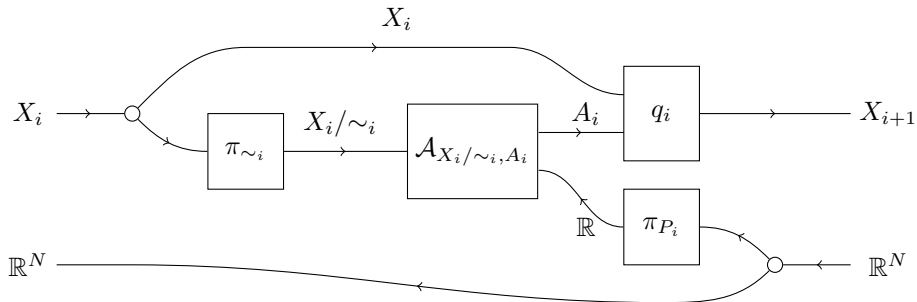
Similarly, we define  $\mathcal{G}_{L+1}$  to be the atomic open game with  $\Sigma_{\mathcal{G}_{L+1}} = \{U\}$  and  $B_{\mathcal{G}_{L+1}}(c) = \{U\}$  for all contexts  $c \in \mathbb{C}((X_{L+1}, \mathbb{R}^N), (I, I))$ , where  $U : (X_{L+1}, \mathbb{R}^N) \rightarrow (I, I)$  is the optic given by  $(\mathbb{R}^N, U, \text{id})$ , with the coend diagram



$\mathcal{G}_{L+1}$  is itself denoted by the string diagram

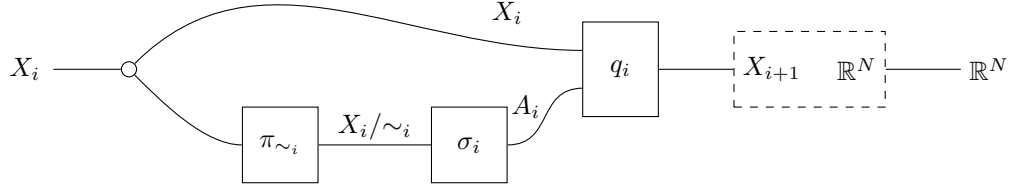


The open game  $\mathcal{G}_i$  for  $1 \leq i \leq L$  is defined by the string diagram



Unwinding this definition, we get that

- $\Sigma_{\mathcal{G}_i} \cong \Sigma_{\mathcal{A}(X_i/\sim_i, A_i)} = X_i/\sim_i \rightarrow D(A_i)$
- For  $\sigma_i : X_i/\sim_i \rightarrow D(A_i)$ ,  $P_{\mathcal{G}_i}(\sigma_i)$  is the optic  $(X_i, \mathbb{R}^n) \rightarrow (X_{i+1}, \mathbb{R}^n)$  given by the coend diagram



- For a context  $c = (\Theta, p, k) : (I, \mathbb{R}^N) \rightarrow (X_i, X_{i+1})$ ,  $B_{\mathcal{G}_i}(c)$  is the constant relation

$$\left\{ \sigma_i : X_i/\sim_i \rightarrow D(A_i) \mid \sigma_i([x_i]_{\sim_i}) \in \arg \max_{a_i \in A_i} \mathbb{E}[\pi_{P_i}(k(p_1, q(p_2, a_i))) \mid p_2 = x_i] \right. \\ \left. \text{for all } x_i \in \text{supp}(p_2) \right\},$$

where  $\mathbb{E}[\pi_{P_i}(k(p_1, q(p_2, a_i))) \mid p_2 = x_i]$  is the conditional expectation of

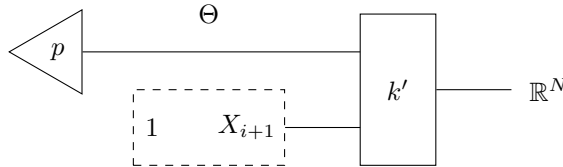
$$I \xrightarrow{p} \Theta \otimes X_i \xrightarrow{\Theta \otimes q(-, a_i)} \Theta \otimes X_{i+1} \xrightarrow{k} \mathbb{R}^N \xrightarrow{\pi_{P_i}} \mathbb{R}$$

given that the right marginal of  $p$  is  $x_i$ .

In order to characterise  $\mathcal{G} = \mathcal{G}_{L+1} \circ \dots \circ \mathcal{G}_0$  inductively we have a choice to either work forwards from  $\mathcal{G}_0$  or backwards from  $\mathcal{G}_{L+1}$ , corresponding roughly to forward induction and backward induction in game theory. We choose the former. The inductive hypothesis is encapsulated in the following lemma.

**Lemma 4.7.3.** *For  $1 \leq i \leq L$ , consider the extensive form game  $G_i$  without specified payoffs given by truncating the original extensive form game  $G$  to the first  $i+1$  levels ( $i$  proper levels plus the root node). Note that the set of leaves of  $G_i$  is (labelled by)  $X_{i+1}$ . Then the open game  $\mathcal{G}_i \circ \dots \circ \mathcal{G}_0 : (1, 1) \rightarrow (X_{i+1}, \mathbb{R}^N)$  is given by the following:*

- $\Sigma_{\mathcal{G}_i \circ \dots \circ \mathcal{G}_0}$  is the set of behavioural strategy profiles of  $G_i$ , namely  $\Sigma_{\mathcal{G}_i \circ \dots \circ \mathcal{G}_0} = \prod_{j=1}^i (X_j/\sim_j \rightarrow D(A_j))$
- $(\mathcal{G}_i \circ \dots \circ \mathcal{G}_0)(\sigma)$  is the optic  $(1, a, !)$ , where  $a : 1 \rightarrow D(X_{i+1})$  is the probability distribution on the leaf nodes of  $G_i$  resulting from playing the behavioural strategy profile  $\sigma$
- Every context of  $\mathcal{G}_i \circ \dots \circ \mathcal{G}_0$  has the form



and thus is equivalent to one of the form  $(1, *, k)$  where  $*$  is the unique distribution on one point and  $k : X_i \rightarrow D(\mathbb{R}^N)$ , by taking  $k$  to be the partial application of  $k' : \Theta \times X_{i+1} \rightarrow D(\mathbb{R}^N)$  to  $p : D(\Theta)$ . Then  $B_{\mathcal{G}_i \circ \dots \circ \mathcal{G}_0}([*, k])(\sigma)$  is the set of best responses to  $\sigma$  in the extensive form game given by  $G_i$  together with payoffs, where the  $i$ th player's payoff from the leaf node  $x \in X_i$  is  $\mathbb{E}[k(x)_i]$  (where  $k(x)_i$  is the  $i$ th marginal of the joint distribution  $k(x) : D(\mathbb{R}^N)$ ).

By taking  $i = L$  in the previous lemma and then post-composing with  $\mathcal{G}_{L+1}$ , we obtain the final result.

**Theorem 4.7.4.**  $\mathcal{G} = \mathcal{G}_{L+1} \circ \dots \circ \mathcal{G}_0 : (1, 1) \rightarrow (1, 1)$  is given by the following:

- The set of strategy profiles of  $\mathcal{G}$  is the set of behavioural strategy profiles of the original extensive form game  $G$
- Every context is equivalent to one of the form  $(1, *, *)$ , and for a behavioural strategy profile  $\sigma$ ,  $B_G(1, *, *) (\sigma)$  is the set of best responses to  $\sigma$  in  $G$

## 5 Conclusion

In this last section we will discuss the wider context of this work, as well as some work in progress and future work.

**Computer support.** Due to the complexity of the definition of Bayesian open games, in practice computer support is necessary to work with real models in the framework. We have created such a software tool in the form of a Haskell library (available at <https://github.com/jules-hedges/open-games-hs>), consisting of a ‘core’ implementation of the monoidal category of Bayesian open games, together with a domain-specific embedded programming language that provides a higher level of description roughly equivalent to string diagrams. We have found this to be a practical tool for modelling, especially for rapid prototyping of models, in a variety of applied domains including auction design, governance modelling and blockchain protocol modelling. The implementation of this tool, and these applied case studies, will be described in a series of future papers.

**Non-finitary probability.** In Section 3 we gave a general theory of open games over a monoidal category. And then, in Section 4 specialised it to the Kleisli category of the finite support probability monad. This restriction to finite support distributions was done for simplicity, but the general machinery we have introduced is applicable to any category of probabilistic functions. Examples include the Kleisli categories of the Giry monads on measurable and Polish spaces [16], the Radon monad on compact Hausdorff spaces [38], and the Kantorovich monad on complete metric spaces [11]. In fact, we believe that Bayesian open games can be formulated over any Markov category [10], which encompasses all of these examples.

**Other solution concepts.** In this paper we have focused on the solution concept of Bayesian Nash equilibrium. It is one of the central solution concepts in applied economic modelling as it allows us to model situations of asymmetric information. It is probably no exaggeration that for most economic situations information is asymmetric and thus the default. Hence, this paper is an important step in making the theory of open games practically useful for a wide range of situations. As a side effect, both ordinary mixed strategy Nash equilibria and correlated equilibria are obtained as special cases of Bayesian Nash equilibrium. However, for dynamic games it is common to use equilibrium refinements such as perfect Bayesian equilibrium or sequential equilibrium, in which individual players’ beliefs are represented explicitly.<sup>9</sup> It is an open question whether it is possible to extend compositional game theory with these stronger solution concepts.

**Behavioural aspects.** A currently-unexplored benefit of Bayesian open games is that neither maximisation of real numbers nor Bayes’ law is involved in the definition of the categorical structure, and instead both enter when decisions are defined. This means that the same framework can equally well accommodate agents who neither perfectly maximise, nor update their beliefs strictly according to Bayes’ law. This could make our framework useful for behavioural game theory, where such alternatives are considered (see, e.g. [4]).

<sup>9</sup>See Chapter 7 in [29] for a general discussion of equilibrium refinements and Chapter 9 in [28] for a discussion of refinements for dynamic games.

## 5.1 Acknowledgements

The text of this paper is closely based on the first author’s PhD thesis, [3]. The authors gratefully thank Mitchell Riley, whose suggestion to apply coend lenses to open games was crucial and very timely.

## References

- [1] George A. Akerlof. The market for ‘lemons’: Quality uncertainty and the market mechanism. *Quarterly Journal of Economics*, 84(3):488–500, 1970. <https://doi.org/10.2307/1879431>. 40, 41
- [2] Jeremy Avigad and Solomon Feferman. Gödel’s functional (“Dialectica”) interpretation. In *Handbook of proof theory*, volume 137 of *Studies in logic and the foundations of mathematics*, pages 337–405. Elsevier, 1998. <https://doi.org/10.2307/420972>. 3
- [3] Joe Bolt. *Probability and nondeterminism in compositional game theory*. PhD thesis, University of Oxford, 2019. 11, 45
- [4] Colin F Camerer. *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press, 2011. 44
- [5] Valeria de Paiva. The dialectica categories. Technical report, University of Cambridge, 1991. 3
- [6] Martin Escardó and Paulo Oliva. Selection functions, bar recursion and backward induction. *Mathematical Structures in Computer Science*, 20(2):127–168, 2010. <https://doi.org/10.1017/S0960129509990351>. 7
- [7] Brendan Fong. *The algebra of open and interconnected systems*. PhD thesis, University of Oxford, 2016. 1
- [8] Brendan Fong, David Spivak, and Rémy Tuyéras. Backprop as functor: A compositional perspective on supervised learning. In *Proceedings of Logic in Computer Science (LiCS) 2019*. ACM, 2019. <https://doi.org/10.1109/LICS.2019.8785665>. 11
- [9] Nate Foster, Michael Greenwald, Jonathan Moore, Benjamin Pierce, and Alan Schmitt. Combinators for bi-directional tree transformations: A linguistic approach to the view update problem. *ACM Transactions on Programming Languages and Systems*, 29(3), 2007. <https://doi.org/10.1145/1232420.1232424>. 3
- [10] Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370, 2020. <https://doi.org/10.1016/j.aim.2020.107239>. 44
- [11] Tobias Fritz and Paolo Perrone. A probability monad as the colimit of spaces of finite samples. *Theory and Applications of Categories*, 34(7), 2019. 44
- [12] Drew Fudenberg and Jean Tirole. *Game theory*. MIT Press, 1991. 1
- [13] Neil Ghani, Jules Hedges, Viktor Winschel, and Philipp Zahn. Compositional game theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 472–481. ACM, 2018. <https://doi.org/10.1145/3209108.3209165>. 1, 2, 37
- [14] Neil Ghani, Clemens Kupke, Alasdair Lambert, and Fredrik Nordvall Forsberg. A compositional treatment of iterated open games. *Theoretical Computer Science*, 741:48 – 57, 2018. ISSN 0304-3975. <https://doi.org/10.1016/j.tcs.2018.05.026>. An Observant Mind : Essays Dedicated to Don Sannella on the Occasion of his 60th Birthday. 14
- [15] Neil Ghani, Clemens Kupke, Alasdair Lambert, and Fredrik Nordvall Forsberg. Compositional game theory with mixed strategies: Probabilistic open games using a distributive law. In *Proceedings of ACT 2019*, volume 323 of *Electronic Proceedings in Theoretical Computer Science*, 2020. <https://doi.org/10.4204/EPTCS.323.7>. 2, 37
- [16] Michèle Giry. A categorical approach to probability theory. *Categorical aspects of topology and analysis*, pages 68–85, 1982. <https://doi.org/10.1007/BFb0092872>. 44
- [17] 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. <https://doi.org/10.1287/mnsc.14.3.159>. 2

- [18] John C Harsanyi. Games with incomplete information played by “Bayesian” players part II. Bayesian equilibrium points. *Management Science*, 14(5):320–334, 1968. <https://doi.org/10.1287/mnsc.14.5.320>.
- [19] John C Harsanyi. Games with incomplete information played by “Bayesian” players, part III. The basic probability distribution of the game. *Management Science*, 14(7):486–502, 1968. <https://doi.org/10.1287/mnsc.14.7.486>. 2
- [20] Jules Hedges. *Towards compositional game theory*. PhD thesis, Queen Mary University of London, 2016. 13
- [21] Jules Hedges. Coherence for lenses and open games. *arXiv:1704.02230*, 2017. <https://doi.org/10.48550/arXiv.1704.02230>. 7
- [22] Jules Hedges. Morphisms of open games. *Electronic Notes in Theoretical Computer Science*, 341:151 – 177, 2018. ISSN 1571-0661. <https://doi.org/10.1016/j.entcs.2018.11.008>. Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV). 11, 41
- [23] Jules Hedges, Paulo Oliva, Evguenia Shprits, Viktor Winschel, and Philipp Zahn. Higher-order decision theory. In *International Conference on Algorithmic Decision Theory*, pages 241–254. Springer, 2017. [https://doi.org/10.1007/978-3-319-67504-6\\_17](https://doi.org/10.1007/978-3-319-67504-6_17). 8
- [24] 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. [https://doi.org/10.1007/978-3-319-51676-9\\_9](https://doi.org/10.1007/978-3-319-51676-9_9). 7, 8
- [25] Julian Hedges. Lenses for philosophers, 2018. Available at <https://julesh.com/2018/08/16/lenses-for-philosophers/>. 3
- [26] Martin Hofmann and Benjamin Pierce. Positive subtyping. *Information and Computation*, 126:11–33, 1996. <https://doi.org/10.1006/inco.1996.0031>. 3
- [27] F. Loregian. *Coend Calculus*, volume 468 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, first edition, 07 2021. <https://doi.org/10.1017/9781108778657>. 2, 15, 27
- [28] Andreu Mas-Colell, Michael D. Whinston, and Jerry Green. *Microeconomic Theory*. Oxford University Press, 1995. ISBN 0195102681. 40, 41, 44
- [29] Michael Maschler, Eilon Solan, and Shmuel Zamir. *Game Theory*. Cambridge University Press, 2013. <https://doi.org/10.1017/CBO9780511794216>. 38, 44
- [30] John Nash. Equilibrium points in  $n$ -person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950. <https://doi.org/10.1073/pnas.36.1.48>. 1
- [31] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton university press, 1944. 1, 2
- [32] Frank Oles. *A category-theoretic approach to the semantics of programming languages*. PhD thesis, Syracuse University, 1982. 3
- [33] Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994. 1
- [34] Luke Palmer. Making Haskell nicer for game programming. Available at <https://lukepalmer.wordpress.com/2007/07/26/making-haskell-nicer-for-game-programming/>, 2007. 3
- [35] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Mathematical Structures in Computer Science*, 7(5):453–468, 1997. <https://doi.org/10.1017/S0960129597002375>. 30
- [36] Mitchell Riley. Categories of optics. *arXiv:1809.00738v2*, 2018. 2, 3, 16, 17, 27, 29
- [37] Mario Román. Open diagrams via coend calculus. In *Proceedings of Applied Category Theory 2020*, volume 333 of *Electronic Proceedings in Theoretical Computer Science*, 2021. <https://doi.org/10.4204/EPTCS.333.5>. 17
- [38] T. Świrszcz. Monadic functors and convexity. *Bulletin de l’Académie Polonaise des Sciences*, 22(1), 1974. 44
- [39] Category Theory Zulip. Change of domain for coends. Archived at <https://mattecapu.github.io/ct-zulip-archive/stream/229199-learning:-questions/topic/topic..22change.20of.20do.main.22.20for.20coends.html>, 2021. 27