Towards compositional game theory

Submitted in partial fulfilment of
the requirements of the degree of
Doctor of Philosophy,
Queen Mary University of London by

Julian Hedges

Statement of originality

I, Julian Hedges, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature:

Date: 25/04/16

Details of collaboration and publications: see §0.6.

Abstract

We introduce a new foundation for game theory based on so-called open games. Unlike existing approaches open games are fully compositional: games are built using algebraic operations from standard components, such as players and outcome functions, with no fundamental distinction being made between the parts and the whole. Open games are intended to be applied at large scales where classical game theory becomes impractical to use, and this thesis therefore covers part of the theoretical foundation of a powerful new tool for economics and other subjects using game theory.

Formally we define a symmetric monoidal category whose morphisms are open games, which can therefore be combined either sequentially using categorical composition, or simultaneously using the monoidal product. Using this structure we can also graphically represent open games using string diagrams. We prove that the new definitions give the same results (both equilibria and off-equilibrium best responses) as classical game theory in several important special cases: normal form games with pure and mixed strategy Nash equilibria, and perfect information games with subgame perfect equilibria.

This thesis also includes work on higher order game theory, a related but simpler approach to game theory that uses higher order functions to model players. This has been extensively developed by Martin Escardó and Paulo Oliva for games of perfect information, and we extend it to normal form games. We show that this approach can be used to elegantly model coordination and differentiation goals of players. We also argue that a modification of the solution concept used by Escardó and Oliva is more appropriate for such applications.

Contents

Foreword

Acknowledgements

Top billing goes to my family (including Jeni, who says hi); my supervisor Paulo Oliva; my second supervisor Edmund Robinson; my closest collaborators Viktor Winschel and Philipp Zahn; my other coauthors Neil Ghani, Mehrnoosh Sadrzadeh and Evguenia Spritz; and my other colleagues in Theory Group at QMUL, especially my fellow PhD students in CS420 that was.

Over the past three and a half years I have had gained from talking with many people. The following list is probably incomplete: Alessandro Abate, Samson Abramsky, Valeria de Paiva, Martin Escardó, Jeremy Gibbons, Helle Hvid Hansen, Peter Hines, Aleks Kissinger, Clemens Kupke, Alexander Kurz, Pierre Lescanne, Adrian Mathias, Arno Pauly, Dusko Pavlović, Alex Smith.

I gratefully acknowledge that my PhD studies were funded by EPSRC doctoral training grant EP/K50290X/1.

A note on style

To begin with, in this introduction I will refer to myself in the first person. Once the thesis proper begins in chapter 1 I will return to using the third person.

I feel very fortunate to be writing a thesis in my native language, and I plan to take advantage of it. The writing throughout is intentionally slightly less formal than would be reasonable in a publication, and in this introduction much less so.

When I read [Ati08] I was overly influenced by point 5 under 'style', namely "Identify papers you have enjoyed reading and imitate their style", even to the expense of the previous point, "Be as clear and succinct as possible while being clear and easy to understand". I immediately thought of [Gir01] and the book [Hoy08], which have both strongly influenced the way I think about logic and programming respectively; both contain a mixture of formal mathematics or computer science with vivid intuitions and downright aggressive personal opinion that borders on philosophy. I have not carried my style to nearly that extreme, but I hope that some of it is visible.

My opinion is that the definition-theorem-proof style of mathematics inherited from Bourbaki will soon (but not quite yet) belong in a past age when theorem provers were not practical to use, and that in the future the style of a typical publication in mathematics will need to change to account for the fact that human-checked proofs are unacceptably unreliable compared with machine-checked proofs. I hope that the outcome is that style will become more informal and focus more on intuition, in contrast to what seems to be happening now (for example, in the homotopy type theory community) with papers written in the ugly syntax of literate proof assistant scripts.

I have intentionally written in continuous prose, rather than dividing into sections labelled 'definition', 'theorem' and 'proof', to reflect the way that mathematics is actually done: plausible definitions, theorems and proof ideas are used to adjust each other until a fixpoint is reached. For example, the proof idea may come first, followed by a definition encapsulating the hypotheses found to be needed to make the proof go through, with the theorem coming last. The proofs in this thesis are indeed checked by hand, although I trust the experimental evidence discussed in §0.2 more than I trust my ability to write correct proofs. Since in some cases several variant or false definitions are given, the 'official' one will always be distinguished by a bold font.

Intended audience

It is very strange to write about 'intended audience' in a thesis, when the rule of thumb is that it will be read by (at most) my supervisor and examiners. However I intend to continue using my thesis as a reference on compositional game theory even after papers are published, in particular because chapter 2 contains far more informal text than would be reasonable in a publication, and I think the informal text is very important. Therefore, I will write here about what background knowledge is assumed.

This thesis contains nontrivial amounts of both pure mathematics (by which I mean the study of mathematical objects for no other reason than their inherent beauty) and applied mathematics (by which I mean mathematics motivated and influenced by modelling problems). An idealised reader has some background knowledge in both game theory and category theory, but I expect that most readers will be familiar with one, but not the other.

The category theory required to read this thesis is mostly monoidal categories, for which the usual reference is [Mac78]. Alternatively, a self-contained introduction to monoidal category theory that emphasises the process-oriented view used in this thesis can be found in [Coe06]. Category theory in this thesis is largely treated as a means to an end, as an axiomatic approach to compositionality and a way to easily prove the soundness of the string diagram language in §2.3. Readers who are category theorists will be able to tell that I am not a category theorist: in particular, no attempt is made to abstractly study the properties of the category of games, the reasoning being very concrete and often by example.

The game theory that is needed as background is also very small, and is discussed in §0.1. A list of topics reads like half of an undergraduate course in game theory: normal form, extensive form, pure and mixed strategies, Nash equilibria, subgame perfect equilibria. It is more important to have an intuition for game theory than any specific piece of mathematical theory, for which a good introduction is [Kre91].

Chapter 0

Introduction

0.1 Background: game theory

There is a tendency for theoretical computer scientists, when writing about game theory, to refer mostly to the oldest references on the subject, such as [vNM44]. To a computer scientist, the term 'game' may mean 'normal form game', or it may mean 'extensive form game', in the latter case often with information sets quietly ignored. In particular, though, the computer scientist will ignore the fact that game theory is a large research areas within economics, which itself is a subject of comparable size to all of computer science. Another mistake that a computer scientist can make, perhaps even simultaneously, is to equate (academic) economics with game theory. These are both errors that I am still in the process of trying to overcome myself.

With that being said, essentially all of the game theory needed to follow this thesis can be found in [vNM44]. For a more concise introduction written by (and therefore readable by) computer scientists I recommend [LBS08]. To computer scientists I would also recommend [Kre91] which, being short on mathematics and long on economic intuition, is likely to be very different to the way they think about the subject. Of the various weighty reference books on game theory, the one I use is [FT91]. Failing that, of course there are endless lecture notes and slides online written for undergraduates in economics, computer science, mathematics, engineering, ...

For the closely related two types of game theory covered in this thesis, namely higher order and compositional, I would like to be clear about how they relate to what I will call classical game theory, by which I mean game theory as covered by these references. The questions are: which features are common? What new features are gained? What features are lost? Which problems are solved, and which are not?

I will begin with the features common to all approaches. A game consists of players or agents, who act in a way that is constrained by some rules or protocol. By this I mean that when each player moves, they have a collection of possible moves, and a collection of possible things they could observe about the past. A strategy for each player is a mapping from possible observations to possible moves, possibly allowing certain side effects such as probabilistic choices or belief updating. Then a 'solution' of the game is an equilibrium, which is a choice of strategy for each player that is stable or non-self-refuting, or equivalently is a fixpoint of a best response function. In general a game may have zero, one or many equilibria. In justifying the solution as a prediction of real world behaviour we assume that the rules of the game and the perfect rationality of the players are common knowledge[1].

A built-in feature of classical game theory is that a choice of strategy for each player will determine a real number for each player called a utility, and the perfect rationality of the players is defined to mean that they act such as to maximise their utility. This is discussed further in §1.2.5. Both higher order and compositional game theory generalise away from this, replacing real numbers with arbitrary objects and allowing rationality to be defined in far more general ways which become part of the specification of a game. I will offer three arguments in favour of doing this. The first is that by abstracting away from a nontrivial but inessential feature, namely real analysis and optimisation, the theory is genuinely simplified, and the significant issues become clearer. The second is that new modelling techniques become available, such as the coordination and differentiation games in §1.2.9 and §2.3.11. The third is that this is a necessary step for compositionality: taking two players who perfectly maximise and composing them together will typically produce a system that does not perfectly maximise.

Since this is in some sense a strict generalisation, we can still revert back to the special case by taking outcomes to be real numbers and considering only agents who maximise real numbers. Indeed, §3.1 and §3.2 of this thesis do exactly that. But what is lost is any piece of theory that begins by assuming that outcomes are utilities and that players maximise. It may even be that we lose the vast majority of all of the literature on game theory this way. To give perhaps the most serious but elementary example, it is impossible to define strategic dominance in general for higher order or compositional games.

Higher order game theory, as a subject which naturally grew out of applications in proof theory, is not really intended to solve any problem in game theory. One feature that stands out, however, is the ability to write very short and elegant functional programs that compute subgame perfect equilibria of perfect information games, including certain infinite games [EO10b, EO12]. The coordination and differentiation games of §1.2.9 and [HOS+15b], in addition, constitute an 'application' of higher order game theory.

Compositional game theory, on the other hand, has been consciously designed as an attack on a specific problem: compositionality. This is the principle that a system should be built by composing together smaller subsystems, and that the behaviour of a system is entirely determined by the behaviour of the subsystems and the ways in which they are composed, and therefore it is possible to reason about the system by structural induction on its decomposition. To a computer scientist compositionality is such a fundamental idea that it is most often not even mentioned. For example, every serious programming language is compositional: a program is built from code blocks composed using sequencing, loops, functions and so on, and the program's behaviour is determined by the behaviour of the code blocks together with the constructs used to compose them. I will discuss the principle of compositionality in considerable detail in §0.4.

[1] The phrase "it is common knowledge that X" means that "all players know that X" together with "all players know that all players know that X", and so on.

Compositionality, however, is an alien concept in game theory, because there is no meaningful formal sense in which a game is built from composing together smaller components. Put bluntly, this is why it is feasible to create a reasonably robust software system containing millions of lines of code, but it is not feasible to work with game theoretic models of comparable scale and complexity. I am not aware of any literature that has come close to identifying the lack of compositionality as a problem in game theory, but nevertheless it is a very serious problem, and it is the problem that is solved in this thesis.

In order to talk systematically about which problems are not solved, I will refer to chapter 5 of [Kre91] for a discussion of the problems of game theory. With the exception of the section titled 'What are a player's payoffs?', which relates to the generalised rationality described above, neither higher order nor compositional game theory contributes anything to these well-known problems. I will divide these into two classes: the problems relating to 'the rules of the game', and the problems relating to equilibrium analysis.

Problems with the 'rules of the game' include:

- A game-theoretic model must have fixed rules, and game theory is unable to model unrestricted negotiation, for example;
- Similarly, it is difficult to model the ability of players to dynamically modify the rules;
- The predictions of the model can be extremely sensitive to apparently small changes in the rules;
- The structure of the game, by default, is common knowledge.

Regarding the last point, in higher order game theory each player's quantifier or selection function is assumed to be common knowledge, and in compositional game theory the entire structure of a string diagram is assumed to be common knowledge. I hope that the usual technique of using Bayesian games and universal type spaces can be generalised to compositional game theory, but that is entirely work for the future.

The central problem of equilibrium analysis is that there is no generally accepted mechanism by which a particular equilibrium can be selected by the players, in some cases even when there is exactly one equilibrium. Both higher order and compositional game theory are fundamentally equilibrium-based, and suffer from the same, familiar problems with equilibrium analysis, and I will say nothing more about it.

0.2 Background: functional programming

Compositional game theory was almost entirely developed by me during an intense two weeks in February 2015 (when I was living in Mannheim), in which time I felt like the Haskell interpreter `ghci` became a sort of extension of my brain. I already had working implementations of the definitions in §2.1 and §2.2, and experimental verification of the results in §3.1 and §3.2, and even more that is not covered in this thesis, before I had even written down the definitions in mathematical language, let alone proved any theorems. The ability to rapidly typecheck and experimentally test many different definitions was crucial in eventually arriving at the definitions that work, and I still genuinely struggle to understand the resulting definitions (particularly those in §2.1) intuitively.

Although the functional programming point of view was built into compositional game theory from the start, I have tried my best to minimise it in the presentation in this thesis, because the intersection of the intended audience with functional programmers might vanish. In particular, monads have mostly been replaced with their Kleisli categories. However, here I will describe my intuitions for the use of functional programming in game theory.

The fundamental intuition I use is **Moggi's thesis** [Mog91], which can be paraphrased as follows: there is a correspondence between

1. Computations that input a value of type X, possibly carry out side effects, and output a value of type Y
2. Functions of type $X \to TY$ for a suitable monad T
3. Morphisms in $\hom_{\mathcal{C}}(X, Y)$ for a suitable category $\mathcal{C}$

The passage from 2 to 3 is to take the Kleisli category $\mathcal{C} = \mathrm{Kl}_T$. The passage from 1 to 2, which involves choosing a suitable monad, is part of the art of functional programming. See also [PP02].

I will give one prototypical example, which is nondeterminism. Nondeterministic choice is the ability of a program to return a result that is not uniquely determined by its input. Instead, for each input the program has a set of outputs that might possibly occur. If this set is empty then the program can never return a result, which can be interpreted either as nontermination or as exceptional termination. The corresponding monad is the powerset monad $\mathscr{P}$, and functions of type $X \to \mathscr{P}Y$ form a useful model of nondeterministic programs. The category that corresponds to this is $\mathrm{Kl}_{\mathscr{P}} = \mathbf{Rel}$, the category of sets and relations, via forward images of relations.

An important idea which is foundational to my research is that side effects, or equivalently monads, are ubiquitous in game theory, and that identifying and classifying them is a useful thing to do. The most important examples are the selection and continuation monads, but they are not intuitive and will be left for chapter 1. More intuitively, consider a player who makes an observation[2] from a type X and then makes a choice from a type Y.

A pure strategy for the player is a function $X \to \mathrm{Id}\, Y$, where Id is the identity monad $\mathrm{Id}\, Y = Y$, whose Kleisli category is **Set**. A mixed strategy is a function $X \to \mathscr{D}Y$, where $\mathscr{D}Y$ is the set of probability distributions on Y. The monad $\mathscr{D}$ is called the probability distribution or (finitary) Giry monad [Gir82], and it is introduced in §2.1.3, along with its Kleisli category **SRel**. The Haskell implementation of probability I use is described in [EK06].

From this viewpoint it is natural to also consider players who make truly nondeterministic choices, without a probability distribution. Such a player has a set of possible choices for each possible observation, and her strategies have type $X \to \mathscr{P}Y$, and are morphisms of **Rel**. Several computer scientists writing

[2] For readers who are game theorists, X is the set of information sets owned by the player, and the strategies we consider are behavioural strategies. The functional programming viewpoint makes behavioural strategies $X \to \mathscr{D}Y$ far more natural to consider than mixed strategies $\mathscr{D}(X \to Y)$, which means that the assumption of perfect recall is essential. Throughout this thesis, the term 'mixed strategy' really means 'behavioural strategy' in the context of dynamic games.

about game theory have independently had the idea of nondeterministic players [LaV06, Pav09, Hed14], although there is little significant theory. In Haskell the most common implementation of nondeterminism uses lists, although there are many alternatives.

If the player makes use of a prior of type A then her strategy is a function of type $X \to \mathrm{Rd}_A\, Y$, where Rd is the reader monad, which acts on sets by $\mathrm{Rd}_A\, Y = A \to Y$. If her strategy moreover has the ability to update the prior with a posterior after making the observation then it has type $X \to \mathrm{St}_A\, Y$, where the state monad St is $\mathrm{St}_A\, Y = A \to Y \times A$.

This Bayesian updating or learning is more complicated than the other examples, because St_A is the only example listed that is a noncommutative monad. In this thesis only commutative monads will be considered, for simplicity. By corollary 4.3 of [PR93], a monad is commutative iff its Kleisli category is symmetric monoidal, which justifies the choice in §2.1.2 to parameterise the definition of open games by an arbitrary symmetric monoidal category. More general premonoidal categories also destroy the connection with string diagrams, although see [Jef97].

This point of view is shared with [Pav09], which moreover refers to Freyd categories [PT99]. The heavier machinery of Freyd categories is avoided in this thesis by assuming that all objects have a comonoid structure (see §2.1.2), which is stronger than necessary but is satisfied by the most important examples.

0.3 Background: logic for social behaviour

In this section I will give a brief literature review of applications of logic and theoretical computer science to game theory. For lack of a better name, I will call this research topic 'logic for social behaviour' after workshops held in Leiden in 2014, Delft in 2015 and Zürich in 2016. Another event that should be mentioned in this context is the 2015 Dagstuhl workshop 'Coalgebraic semantics of reflexive economics' [AKLW15]. Many researchers in this area also consider applications to social choice theory, especially preference aggregation, but I will mention only [Abr15], which links Arrow's famous impossibility theorem with category theory.

A starting point is [Pav09], which proposes to study game theory using ideas from program semantics, in particular viewing games as processes which can have side effects such as state and probabilistic choices. That paper suggests a larger research programme called 'abstract game theory' in which this thesis can be located, although see §2.1.1.

An approach to infinitely repeated games using coinduction was introduced in [LP12] and continued in [AW15] and the working paper [BW13]. Infinite and coalgebraic games are mentioned only briefly in this thesis, in §2.2.1 and the conclusion. However, given that coinduction and bisimulation are the *correct* techniques for reasoning about infinite processes, it is likely that they will continue to be important in game theory. In particular, if trying naively to verify that some strategy of an infinite game is an equilibrium, then infinitely many properties must be checked; however a finitary proof technique based on bisimulation should be expected to work. As yet, coalgebraic game theory has not been connected with the classical approach to repeated games using real analysis, as covered for example in [FT91], and the very extensive literature on repeated games. An unrelated application of coalgebra to game theory is [MI04], which shows that universal Harsanyi type spaces are also final coalgebras.

Another line of work begins with [lR14], connecting two-valued games and determinacy theorems with real-valued games and existence theorems. In [lRP14] this is moreover connected with synthetic topology [Esc04], which is related to ideas I am working on involving computably compact sets of probability distributions in game theory, which are not in the scope of this thesis.

Practical experience of applying functional programming techniques to economic modelling is described in [BMI⁺11], [IJ13] and [BKP15]. More generally, [EK06] describes the application of functional programming to mathematical modelling in biology. I directly quote the last sentence of that paper: "In particular, the high-level abstractions allowed us to quickly change model aspects, in many cases immediately during discussions with biologists about the model." Due to the close connections between game theory and computational effects described in §0.2, I expect the gains of using functional programming to increase in the future.

Finally, algorithmic game theory [NRTV07] is a large topic that studies the computational complexity of Nash equilibria and other constructions in game theory, which began with the result in [DGP06] that computing approximate Nash equilibria is infeasible. Algorithmic game theory can be contrasted with the semantic approach to game theory that this thesis represents, but will likely be an essential ingredient in the research project outlined in the conclusion.

0.4 On compositionality

This section is essentially an essay, loosely based on a talk I gave at Logic for Social Behaviour 2016 in Zürich, after the vast majority of the thesis was written.

The term *compositionality* is commonplace in computer science, but is not well-known in other subjects. Compositionality was defined in §0.1 as the principle that a system should be designed by composing together smaller subsystems, and reasoning about the system should be done recursively on its structure. When I thought more deeply, however, I realised that there is more to this principle than first meets the eye, and even a computer scientist may not be aware of its nuances.

It is worthwhile to spend some time thinking about various natural and artificial systems, and the extent to which they are compositional. To begin with, it is well known that most programming languages are compositional. The behaviour of atomic[3] statements in an imperative language, such as variable assignments and IO actions, is understood. Functions are written by combining atomic statements using constructs such as sequencing (the 'semicolon' in C-like syntax), conditionals and loops, and the behaviour of the whole is understood in terms of the behaviour of the parts together with the ways in which they are combined. This scales sufficiently well that a team of programmers can broadly understand the behaviour of a program consisting of hundreds of millions of individual atomic statements.

When the software industry began software was unstructured, with no intermediate concepts between atomic statements and the entire program, and much of its history has been the creation of finer intermediate concepts: code

[3] The term 'atomic' is used naively here, and does not refer to concurrency.

blocks, functions, classes, modules. Compositionality is not all-nor-nothing, but is slowly increased over time; nor is it entirely well-defined, with many tradeoffs and heated debates in the design and use of different language features. Even with a modern well-designed language it is possible to write bad code which cannot be easily decomposed; and even though there are many design patterns and best practice guidelines, good software design is ultimately an art.

Going beyond software, consider a physical system designed by human engineers, such as an oil refinery. An individual component, such as a pump or a section of pipe, may have a huge amount of engineering built into it, with detailed knowledge of its behaviour in a wide variety of physical situations. It is then possible to connect these components together and reuse knowledge about the components to reason about the whole system. As in software, each component has an 'interface', which is a high level understanding of its behaviour, with unnecessary details being intentionally forgotten.

As a third example, an organisation made of human beings, such as a company or university, is also built in a compositional way, demonstrating that engineering is not a requirement. It is possible to understand the behaviour of a department without knowing the details of how the behaviour is implemented internally. For example, a software engineer can use a computer without knowing the exact process through which the electricity bill is paid, and will probably not even be aware if the electricity provider changes. This is another example of reasoning via an interface.

Clearly interfaces are a crucial aspect of compositionality, and I suspect that interfaces are in fact synonymous with compositionality. That is, compositionality is not just the ability to compose objects, but the ability to work with an object *after intentionally forgetting how it was built*. The part that is remembered is the 'interface', which may be a type, or a contract, or some other high-level description. The crucial property of interfaces is that their complexity stays roughly constant as systems get larger. In software, for example, an interface can be used without knowing whether it represents an atomic object, or a module containing millions of lines of code whose implementation is distributed over a large physical network.

For examples of non-compositional systems, we look to nature. Generally speaking, the reductionist methodology of science has difficulty with biology, where an understanding of one scale often does not translate to an understanding on a larger scale. For example, the behaviour of neurons is well-understood, but groups of neurons are not. Similarly in genetics, individual genes can interact in complex ways that block understanding of genomes at a larger scale.

Such behaviour is not confined to biology, though. It is also present in economics: two well-understood markets can interact in complex and unexpected ways. Consider a simple but already important example from game theory. The behaviour of an individual player is fully understood: they choose in a way that maximises their utility. Put two such players together, however, and there are already problems with equilibrium selection, where the actual physical behaviour of the system is very hard to predict.

More generally, I claim that *the opposite of compositionality is emergent effects*. The common definition of emergence is a system being 'more than the sum of its parts', and so it is easy to see that such a system cannot be understood only in terms of its parts, i.e. it is not compositional. Moreover I claim that *non-compositionality is a barrier to scientific understanding*, because it breaks the reductionist methodology of always dividing a system into smaller components and translating explanations into lower levels.

More specifically, I claim that compositionality is strictly necessary for working at scale. In a non-compositional setting, a technique for a solving a problem may be of *no use whatsoever* for solving the problem one order of magnitude larger. To demonstrate that this worst case scenario can actually happen, consider the theory of differential equations: a technique that is known to be effective for some class of equations will usually be of no use for equations removed from that class by even a small modification. In some sense, differential equations is the ultimate non-compositional theory.

Of course emergent phenomena do exist, and so the challenge is not to avoid them but to *control* them. In some cases, such as differential equations, this is simply impossible due to the nature of what is being studied. The purpose of this thesis is to demonstrate that it is possible to control emergent effects in game theory, although it is far from obvious how to do it. A powerful strategy that is used in this thesis is *continuation passing style*, in which we expand our model of an object to include not only its behaviour in isolation, but also its behaviour in the presence of arbitrary environments. Thus an emergent behaviour of a compound system was already present in the behaviour of each individual component, when specialised to an environment that contains the other components.

As a final thought, I claim that *compositionality is extremely delicate*, and that it is so powerful that it is worth going to extreme lengths to achieve it. In programming languages, compositionality is reduced by such plausible-looking language features as goto statements, mutable global state, inheritance in object-oriented programming, and type classes in Haskell. The demands placed on game theory are extremely strong: seeing a game as something fundamentally different to a component of a game such as a player or outcome function breaks compositionality; so does seeing a player as something fundamentally different to an aggregate of players; so does seeing a player as something fundamentally different to an outcome function. This thesis introduces *open games*, which include all of these as special cases.

0.5 Overview of the thesis

The thesis is divided into three large chapters, each of which is divided into three sections, each of which is divided into many subsections. Each section begins with a 'discussion' subsection that gives motivation and background. The serious part of the thesis consists of chapter 2 and chapter 3, with chapter 1 as a sort of extended introduction.

The subject of this thesis is two new approaches to game theory, which can be called the 'higher order' approach and the 'compositional' approach. Higher order game theory is chronologically prior and much simpler, and can serve as an introduction to the modes of thinking needed for compositional game theory, which is much more complicated and unfamiliar. In principle it should be possible to begin reading at chapter 2 and locally follow hyperlinks back into chapter 1 when necessary to refer to definitions and notations. I do not recommend this, however.

The main objects of study in higher order game theory are so-called quantifiers and selection functions, which are introduced and studied in isolation in §1.1. Simultaneous or normal form higher order games are studied in §1.2, and sequential or perfect information games in §1.3. The contents of §1.2 is closely based on [HOS+15b], and §1.3 is closed based on [EO11] (and therefore is not my own work), with §1.1 being a mixture of the two.

Open games, the objects of study in compositional game theory, are introduced in §2.1. This is a section heavy on definitions, introducing the definitions of open games, decisions, computations and counits. Sequential and parallel composition of open games are studied in §2.2, including some important theorems about how composition behaves. This is applied in §2.3 to give a string diagram language for specifying games. The whole of chapter 2 is based on the preprint [GH16].

The purpose of chapter 3 is to formally connect open games to standard game theory, which previously is done only informally. In §3.1 normal form games, both with pure and mixed strategies, are shown to be a special case of open games. The same is done for perfect information games with pure strategies in §3.2, together with some sketched ideas for extending to imperfect information. Finally, §3.3 returns to more theoretical considerations by exploring a possible solution concept for arbitrary open games, which gives a connection between compositional game theory and higher order game theory.

0.6 Publications

At the time of submission I have three publications: [Hed13], [Hed14] and [Hed15a]. Of these, the last is on a different topic, and this thesis contains little material from the first two.

On the other hand, this thesis does contain large amounts of material from the preprints [HOS+15b] and [GH16]. A third preprint [HS16] is not related to this topic. Two further preprints, [Hed15b] and [HOS+15a], are not currently under review and have not been kept up to date.

0.7 Notation and conventions

Function application is written without brackets whenever it is unambiguous, and the function set arrow $\rightarrow$ associates to the right, so if $f : X \rightarrow Y \rightarrow Z$ then $fxy = (f(x))(y)$. λ-abstractions are denoted $\lambda(x : X).t$, where x is the abstracted variable and X its type. The scope of the abstraction extends as far as possible to the right, and binds tighter than everything except parentheses and the equals sign. The application of a selection function to a λ-abstraction, for example, is written $\varepsilon\lambda(x : X).t$.

Binary and arbitrary products are written $\times$ and $\prod$, binary and arbitrary coproducts are written $+$ and $\sum$, and both bind tighter than $\rightarrow$, so for example $A \times B \rightarrow C + D$ means $(A \times B) \rightarrow (C + D)$. The folded operators bind tighter than their binary equivalents, so for example $\prod_i X_i \times Y$ means $(\prod_i X_i) \times Y$. Projections from a product are written π, and injections into a coproduct are ι. If $x : \prod_{i:I} X_i$ then I write x_i for $\pi_i x : X_i$. Similarly, if $f : X \rightarrow \prod_{i:I} Y_i$ then I write f_i for $\pi_i \circ f : X \rightarrow Y_i$. The type with one element is 1, and its element is $* : 1$. The subscript $-i$, as in x_{-i}, f_{-i}, π_{-i} is used for projection onto $\prod_{j \neq i} X_j$.

The notation (x_i, x_{-i}), common in game theory, additionally uses the natural isomorphism $X_i \times \prod_{j \neq i} X_j \cong \prod_{j:I} X_j$ implicitly.

I distinguish carefully between 'types' X, which are sets, and 'sets' $A \subseteq X$, which are functions $A : X \to \mathbb{B}$ where X is a type and $\mathbb{B}$ is the type of booleans containing $\bot$ and $\top$. To confuse matters, however, I often also use the term 'set' to refer to a type when the difference is harmless, because there is so much social inertia behind the 'set' terminology. If $A \subseteq X$ and $x : X$, then the notation $x \in A$ is shorthand for $Ax = \top$, and $\{x : X \mid \cdots\}$ is shorthand for its characteristic function. $\mathscr{P}X = X \to \mathbb{B}$ is the set of all subsets of the type X.

$\mathbb{P}[\alpha = x] : [0, 1]$ is the probability that a random variable $\alpha : \mathscr{D}X$ is equal to $x : X$. Here $\mathscr{D}$ is the probabilistic analogue of $\mathscr{P}$, defined in §2.1.3. If $\alpha : \mathscr{D}\mathbb{R}$ then $\mathbb{E}[\alpha] : \mathbb{R}$ is the expected value of α.

Some symbols are reserved for particular uses. Types of plays or moves are denoted X, Y, Z and types of outcomes are denoted R, S, T. Σ is the set of strategy profiles of a game, and σ is a strategy profile. $\mathbf{P}$ is a play function, which converts a strategy profile into a play, and $\mathbf{B}$ is a best response function, which takes a strategy profile to its set of best responses. $\mathbf{R}$ is a rationality function (§2.1.7) and $\mathbf{C}$ is a coplay function (§2.1.4), which appear only in the context of open games. ε and δ are single-valued selection functions, E is a multi-valued selection function, φ and ψ are single-valued quantifiers, and Φ is a multi-valued quantifier. q is an outcome function, which takes a play to an outcome. $\mathscr{U}$ is the unilateral deviation operator. $\mathcal{G}$ and $\mathcal{H}$ are open games, and $\mathcal{D}$ is a decision.

Chapter 1

Higher order game theory

1.1 Decision theory

1.1.1 Discussion

The purpose of chapter 1 of this thesis, which is largely based on [HOS$^+$15b] and [EO11], is to introduce the reader to a particular way of thinking about game theory. Although it is logically self-contained, readers unfamiliar with game theory should read it together with another source, such as [LBS08] or [Kre91], that gives a more classical introduction with motivation and standard examples. Readers already familiar with game theory could begin reading at chapter 2 and follow hyperlinked references into this part when necessary.

Suppose we have some situation in which an agent is choosing a move of type X. After the choice is made, she receives some outcome, say of type R. For example, if we are representing preferences by utilities, we would take $R = \mathbb{R}$. The outcome depends not just on the agent's choice, but also on the choices of other agents, the 'rules' of the situation, and the agent's own preferences. The first key concept of this thesis is that these additional dependencies will be abstracted away into a single function $k : X \to R$ mapping choices to outcomes. This function will be called a (strategic) **context**.

If we view an agent as computing a move, the context represents the computation done afterwards by the environment. This leads us to the principle that *strategic contexts are continuations.* Furthermore the structure of game theory, and especially the definition of Nash equilibrium (§1.2.3), is such that many of our definitions have explicit access to the context. A computation which has access to its calling environment by means of a continuation is precisely a continuation passing style computation. Throughout this thesis, and especially in §2.2 we will see that the mathematical structure of game theory can be usefully improved by allowing more things to depend on an arbitrary continuation. This is the principle, perhaps the most important single idea in this thesis, that *game theory wants to be in continuation passing style.*

We will describe agents by their behaviour on each context. There are two options, which lead respectively to 'quantifiers' and 'selection functions': we can map the context to either the good outcomes, or the good moves. We will see in §1.2 that the latter is preferred for technical reasons, and so we will adopt the following slogan: *to know an agent is to know her preferred moves in every* *context.*

1.1.2 Quantifiers

Consider a sentence of predicate logic of the form

$$\exists(x : X).kx$$

Here k is a predicate, which either holds or does not hold for each element of the domain of quantification X. Thus, we can view k as a function $k : X \to \mathbb{B}$, where $\mathbb{B} = \{\bot, \top\}$ is the type of booleans. Since the meaning of our sentence is invariant under α-renaming, it depends only on the value of the function k, and thus could be unambiguously written with the point free syntax $\exists k$. For comparison, there is a familiar example of point free syntax in measure theory:

$$\int_X k \, \mathrm{d}\mu = \int_X k(x) \, \mathrm{d}\mu(x)$$

Since $\exists k$ has a value of type $\mathbb{B}$ and depends only on $k : X \to \mathbb{B}$, we can say that $\exists$ is a particular function

$$\exists : (X \to \mathbb{B}) \to \mathbb{B}$$

To be precise, $\exists$ is the function of this type defined by

$$\exists k = \begin{cases} \top & \text{if } kx = \top \text{ for some } x : X \\ \bot & \text{otherwise} \end{cases}$$

Within an ambient higher-order theory, the sentence $\exists(x : X).kx$ (which involves only a first-order quantifier) is then equivalent to $\exists k$; to be clear, this second $\exists$ is not syntactically speaking a quantifier of the logical theory, but is simply an ordinary higher-order function which behaves as a quantifier. We can do the same thing with the universal quantifier: it is a function

$$\forall : (X \to \mathbb{B}) \to \mathbb{B}$$

defined by

$$\forall k = \begin{cases} \top & \text{if } kx = \top \text{ for all } x : X \\ \bot & \text{otherwise} \end{cases}$$

Abstracting from these two cases leads to the definition of a **generalised quantifier** in [Mos57] as an arbitrary function of type $(X \to \mathbb{B}) \to \mathbb{B}$.

This is generalised one step further in [EO10a], by allowing the type of booleans $\mathbb{B}$ to be replaced by an arbitrary type R. The most important new example that this gains us is maximisation of real-valued functionals $k : X \to \mathbb{R}$: by the same reasoning as before, the expression

$$\max_{x:X} kx$$

can be written as $\max k$, and we can view max as a function

$$\max : (X \to \mathbb{R}) \to \mathbb{R}$$

In summary, we define a **quantifier** on a set X as an arbitrary function

$$\varphi : \mathscr{K}_R X$$

where $\mathscr{K}_R X = (X \to R) \to R$. We will view φ as a function that takes each context $k : X \to R$ to an agent's preferred outcome given that context. We should always think of this as an outcome that is in the image of the context, that is, it is an outcome that can actually occur given that context.

For example, the quantifier $\max : \mathscr{K}_{\mathbb{R}} X$ models a classical economic agent who maximises utility, in the sense that the preferred outcome is the one that is maximal among those that can be attained. In this sense, quantifiers can be seen as a generalisation of utility maximisation, which abstracts away the irrelevant fact that we are working with the ordered real numbers, and allows us to focus on the important structure.

The existential quantifier $\exists : \mathscr{K}_{\mathbb{B}} X$ is also an instance of maximisation, this time over the discrete order $\bot < \top$. We also observe that $\min : \mathscr{K}_{\mathbb{R}} X$ is a quantifier, and that $\forall : \mathscr{K}_{\mathbb{B}} X$ similarly minimises over the order $\bot < \top$.

1.1.3 The continuation monad

In §1.1.2 we introduced the type $\mathscr{K}_R X = (X \to R) \to R$. The operator $\mathscr{K}_R$ is well known in programming language theory where it is called the **continuation monad** [Koc71, Mog91]. This means that we have unit maps $\eta : X \to \mathscr{K}_R X$, and for each function $f : X \to \mathscr{K}_R Y$ a Kleisli extension $f^* : \mathscr{K}_R X \to \mathscr{K}_R Y$. Explicitly, these are given by

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

and

$$f^* \varphi = \lambda(k : Y \to R).\varphi \lambda(x : X).fxk$$

A Kleisli arrow $\varphi : X \to \mathscr{K}_R Y$ is viewed as a computation of type $X \to Y$ in **continuation passing style**. This means that after the function $X \to Y$ has terminated the result is passed to a continuation $Y \to R$, and the computation is allowed to have first class access to its continuation.

An important fact about the continuation monad, proved in [Koc71], is that for an arbitrary strong monad T, the monad morphisms $T \to \mathscr{J}_R$ are exactly in bijection with the T-algebras with carrier R.

The structure of the continuation monad is related to game theory in [EO11, Hed14], but we will not do so in this thesis, because of the argument in [HOS⁺15b] and §1.2 for preferring selection functions to quantifiers.

1.1.4 Selection functions

Just as a quantifier gives the best outcome in a context, so a selection function gives the best move. Thus, a selection function is a function $\varepsilon : \mathscr{J}_R X$ where

$$\mathscr{J}_R X = (X \to R) \to X$$

The selection function corresponding to the existential quantifier is the Hilbert epsilon operator. In the Hilbert calculus $\varepsilon(x : X).kx$ is a term that, by definition, satisfies k if possible. We can informally define $\varepsilon : \mathscr{J}_{\mathbb{B}} X$ by

$$\varepsilon k = \begin{cases} \text{some } x : X \text{ satisfying } kx & \text{if such } x \text{ exists} \\ \text{arbitrary} & \text{otherwise} \end{cases}$$

By the axiom of choice, we can obtain such a function ε satisfying this specification.

Similarly, the selection function corresponding to max is $\arg\max : \mathscr{J}_{\mathbb{R}} X$, which chooses some point at which a function is maximised. This is ordinarily written as

$$\arg\max k = \arg\max_{x:X} kx$$

Again, because a function may attain its maximum at many points, we generally need the axiom of choice to actually obtain arg max as a function.

Although mathematically speaking selection functions are often about *optimality*, it will sometimes be useful to think in anthropomorphic terms of *satisfaction*. To talk about the value φk, we say that if an agent is *choosing in the context* $k : X \to R$, then she is **satisfied** with the outcome φk. Similarly, she is satisfied with making the move εk. If the agent has a quantifier φ that chooses an outcome $\varphi k \notin \mathrm{Im}(k)$ that is not attainable by any move then the agent's preferences in the context k are **unrealistic**, because she will never be satisfied with any outcome that can actually occur.

1.1.5 Attainment

We have given two types of functions that can be used as models of agents, and now we will study the relationship between them. We begin by noticing that a selection function ε defines a quantifier $\overline{\varepsilon}$ by the equation

$$\overline{\varepsilon} k = k(\varepsilon k)$$

This defines a map $\overline{\cdot} : \mathscr{J}_R X \to \mathscr{K}_R X$, which can be proved to be a morphism of monads, see lemma 6.3.6 of [EO10a], where the selection monad $\mathscr{J}_R$ is defined in §1.3.2.

We will say that the selection function ε **attains** the quantifier φ just if $\varphi = \overline{\varepsilon}$, and we will call the quantifier φ **attainable** if it is attained by some selection function.

As a first example, the Hilbert epsilon operator ε attains the existential quantifier. This generalises Hilbert's definition of the existential quantifier as $\exists k = k(\varepsilon k)$. For if $\exists k = \top$ then we have at least one $x : X$ satisfying $kx = \top$, and so $x = \varepsilon k$ has this property. Conversely, if $\exists k = \bot$ then there is no such x, and so $x = \varepsilon k$ is some arbitrary point, which has $kx = \bot$. As a second example, it is easy to see that arg max attains max, essentially by definition.

For a less trivial example, consider the definite integration operator

$$\int : \mathscr{K}_{\mathbb{R}}[0,1]$$

defined by

$$\int k = \int_0^1 kx \, \mathrm{d}x$$

The mean value theorem tells us that if $p : [0,1] \to \mathbb{R}$ is integrable then there is some point $x : [0,1]$ with the property that

$$kx = \int k$$

We can apply the axiom of choice to form a selection function $\varepsilon : \mathscr{J}_{\mathbb{R}}[0,1]$ that takes each k to such an x. Thus, $\int$ is an attainable quantifier. This is given as an example in [EO10a].

For a second interesting example, also from [EO10a], suppose we work in a setting (such as a cartesian closed category of domains) in which every endomorphism $k : X \to X$ has a canonical fixpoint, computed by a function

$$\mathrm{fix} : (X \to X) \to X$$

In this case we have $R = X$, and $\mathscr{J}_X X = \mathscr{K}_X X$, and so fix can be seen as both a quantifier and a selection function. Moreover, because $\mathrm{fix}\, k$ is guaranteed to be a fixpoint of k we have $k(\mathrm{fix}\, k) = \mathrm{fix}\, k$, and hence $\mathrm{fix} : \mathscr{J}_X X$ attains $\mathrm{fix} : \mathscr{K}_X X$.

1.1.6 Multi-valued variants

Quantifiers were generalised yet another step in [EO11], by allowing the quantifier to return a set of results,

$$\Phi : (X \to R) \to \mathscr{P} R$$

We will call a function with this type a **multi-valued quantifier**. Similarly, in [HOS+15b] and [HOS+15a], selection functions were generalised to **multi-valued selection functions**

$$E : (X \to R) \to \mathscr{P} X$$

Multi-valued quantifiers were introduced in [EO11] for applications in game theory and the proof of Bekič's lemma. For example, the max quantifier is single-valued for a total order such as $\mathbb{R}$, but is multi-valued for a preorder such as $\mathbb{R}^n$ with $x \leq_i y \iff x_i \leq y_i$. The step from single-valued to multi-valued selection functions makes it harder to work with sequential games (because we lose the monad structure, see §1.3), but easier to work with simultaneous games. Thus, §1.2 will focus on multi-valued selection functions, but §1.3 on single-valued selection functions.

We will give two important examples of multi-valued selection functions. The first is the multi-valued variant of arg max,

$$\arg\max : (X \to \mathbb{R}) \to \mathscr{P} X$$

which chooses *all* points at which its argument is maximised:

$$\arg\max k = \{x : X \mid kx \geq kx' \text{ for all } x' : X\}$$

Notice that under reasonable hypotheses $\arg\max k$ is nonempty.

The second example is the multi-valued fixpoint operator

$$\mathrm{fix} : (X \to X) \to \mathscr{P} X$$

defined by

$$\operatorname{fix} k = \{x : X \mid x = kx\}$$

The fixpoint operator on sets (as opposed to posets) is naturally multivalued, because a function may have zero, one or many fixpoints, and no preferred fixpoint.

It is a subtle question whether we should allow multi-valued quantifiers and selection functions to return the empty set on any input. For some applications in game theory it is useful to suppose that the sets are always non-empty. We will call such a multi-valued quantifier or selection function **total**, after [Hed13]. For example arg max on a finite set is total. This is the approach taken in [HOS+15b].

On the other hand, there are situations where it is unreasonable to require totality, such as when working with games of imperfect information with pure strategies, so that equilibria may not exist. We will define the **domain** of a multi-valued quantifier or selection function to be the subset of $X \to R$ on which it returns a nonempty set. This will be denoted by $\operatorname{dom}(E)$ or $\operatorname{dom}(\Phi)$. For example, the domain of arg max on a compact space X contains all continuous functions. Of course, a multi-valued quantifier or selection function can be restricted to a total one on its domain. This is the approach taken in [Hed13].

1.1.7 Multi-valued attainment

Multi-valued quantifiers and selection functions also support a concept of attainment, namely that $E : (X \to R) \to \mathscr{P}X$ attains $\Phi : (X \to R) \to \mathscr{P}R$ if for all $k : X \to R$ we have

$$\{kx \mid x \in Ek\} \subseteq \Phi k$$

This definition gives the expected attainments: for example, the multi-valued arg max attains max (where max is a single-valued quantifier viewed as a multi-valued quantifier by returning singletons), and the multi-valued fixpoint operator attains itself.

By analogy to the overline operator $\overline{\cdot} : \mathscr{J}_R \to \mathscr{K}_R$, given a multi-valued selection function $E : (X \to R) \to \mathscr{P}X$ we can define the 'smallest' multi-valued quantifier $\overline{E} : (X \to R) \to \mathscr{P}R$ attained by E, namely

$$\overline{E}k = \{kx \mid x \in Ek\}$$

Equivalently, $\overline{E}k$ is the forward image of Ek under k. However, in the multi-valued case we can also do this in reverse, converting a quantifier Φ into the 'largest' selection function $\overline{\Phi}$ attaining it, namely

$$\overline{\Phi}k = \{x : X \mid kx \in \Phi k\}$$

As suggested, the types $(X \to R) \to \mathscr{P}X$ and $(X \to R) \to \mathscr{P}R$ both carry a partial order structure inherited from the powerset operator. Given multi-valued selection functions $E_1, E_2 : (X \to R) \to \mathscr{P}X$ we will say that E_1 **refines** E_2, and write $E_1 \sqsubseteq E_2$, if for every $k : X \to R$ we have $E_1 k \subseteq E_2 k$. Similarly, for quantifiers we will say that Φ_1 refines Φ_2 and write $\Phi_1 \sqsubseteq \Phi_2$. With this notation, we can say that E attains Φ iff $\overline{E} \sqsubseteq \Phi$.

We can view a single-valued selection function $\varepsilon : \mathscr{J}_R X$ as multi-valued by setting $Ek = \{\varepsilon k\}$, and so we can talk about refinement between single-valued and multi-valued quantifiers. Specifically, we say that ε refines E iff for all k we have $\varepsilon k \in Ek$. By the axiom of choice, a multi-valued selection function has a singled-valued refinement iff it is total. This also applies to quantifiers.

The overline operators define a Galois connection between the refinement orders. Given a selection function $E : (X \to R) \to \mathscr{P}X$ and a quantifier $\Phi : (X \to R) \to \mathscr{P}R$,

$$\overline{E} \sqsubseteq \Phi \iff E \sqsubseteq \overline{\Phi}$$

The proof is straightforward, by showing that both sides are equivalent to the claim that for every $k : X \to R$ and $x : X$, if $x \in Ek$ then $kx \in \Phi k$.

The double overline operator on total quantifiers is the identity, because

$$\overline{\overline{\Phi}}k = \{kx \mid x \in \overline{\Phi}k\} = \{kx \mid kx \in \Phi k\} = \Phi k \cap \operatorname{Im} k$$

On selection functions, following order-theoretic terminology we will think of $\overline{\overline{-}}$ as a *closure operator*, and a selection function E satisfying $E = \overline{\overline{E}}$ will be called **closed**.

1.1.8 Modifying the outcome type

The operator $\mathscr{J}_R$ defined in §1.1.4 is contravariant in the outcome type R, in contrast to the continuation monad $\mathscr{K}_R$, which is not functorial in R. Moreover, the action of $\mathscr{J}_R$ on morphisms of outcomes has a clear game-theoretic reading.

Given a selection function $\varepsilon : \mathscr{J}_R X$ and a function $f : S \to R$, let $f \cdot \varepsilon : \mathscr{J}_S X$ be the selection function

$$f \cdot \varepsilon = \lambda(k : X \to S).\varepsilon(f \circ k)$$

This can easily be shown to commute with the other operations on $\mathscr{J}_R$, including the product of selection functions (§1.3.3).

This also applies to multi-valued selection functions, where the selection function $E : (X \to R) \to \mathscr{P}X$ changes to

$$f \cdot E = \lambda(k : X \to S).E(f \circ k)$$

If we take f to be the projection $\pi_i : \mathbb{R}^N \to \mathbb{R}$ then the multi-valued selection function $\arg\max : (X \to \mathbb{R}) \to \mathscr{P}X$ changes to

$$\pi_i \cdot \arg\max = \lambda(k : X \to \mathbb{R}^N).\{x : X \mid k_i x \geq k_i x' \text{ for all } x' : X\}$$

which is the selection function modelling an agent who maximises the ith coordinate while ignoring the others. This selection function is used in [HOS$^+$15b].

1.2 Normal form games

1.2.1 Discussion

A definition of game was given in [vNM44] that is general enough for most purposes, the so-called extensive form games. At this point we will not introduce extensive form games formally, but we will discuss some of the important concepts.

A game consists of players, who make choices. The choices made by all players, together with the rules of the game, determine an outcome. The choices made by the players are constrained by the fact that each player has (usually different) preferences over the outcomes, and each player acts in such a way as to bring about their preferred outcome. We will model preferences of players by quantifiers or selection functions, which abstracts away more specific definitions such as preference relations or utilities used in standard game theory.

In order to make an informed choice the player needs to know which outcomes will occur for a given choice, but to know this, she needs to know what the other players will choose. However, the other players are reasoning in the same way, and need to know what she will choose, and so we have a circularity. The circularity is resolved by a solution concept, each of which is a proposed definition for what it means for a player to choose rationally, under various assumptions about the other players.

The game may have some dynamic structure in which some players can observe (possibly partial information about) the choices of some other players before making their own choice. A **strategy** for a player is a function that chooses a move, contingent on observed information. Strategies can be used to abstract away the details of how players interact with each other.

Given an arbitrary game, we can define a new game called its **normalisation**. In this new game, the choices are precisely the strategies of the previous game. Given a strategy for each player, we can 'play out' the strategies to determine a choice for each player in the old game, which in turn determines an outcome. The new game is played simultaneously, with no player able to make any direct observations. Thus, games with dynamic structure can be disregarded, and we can focus on simultaneous games only.

For example, suppose the first player chooses $x : X$, and then the second player perfectly observes it and chooses $y : Y$, with the outcome being $q(x, y)$. In the normalisation, simultaneously the first player chooses a strategy $\sigma_1 : X$ and the second player chooses a strategy $\sigma_2 : X \to Y$, with the outcome being $q'(\sigma_1, \sigma_2) = q(\sigma_1, \sigma_2\sigma_1)$.

The most standard solution concept for simultaneous games is called the Nash equilibrium. This is unable to distinguish between an extensive form game and its normalisation, in the sense that it gives the same 'solutions' (rational choices) for both. However there are more refined solution concepts, such as subgame perfect equilibrium, which can distinguish an extensive form game from its normal form. Thus, it is still important to study extensive form games.

This section is based almost entirely on [HOS+15b].

1.2.2 Games, strategies and unilateral continuations

We begin with a collection I of players. Each player $i : I$ has a nonempty set X_i of **moves**. We also have a set R of outcomes, and an **outcome function**

$$q : \prod_{i:I} X_i \to R$$

Since the game is played simultaneously, a **strategy** for each player is simply a move $\sigma_i : X_i$. The function q (together with its type) completely specifies the *rules* of a normal form game. We will leave the specification of the *players* until §1.2.3.

A **strategy profile** is a tuple of strategies for each player,

$$\sigma : \prod_{i:I} X_i$$

A **play** is a tuple of moves (which has the same type) and playing the strategy profile σ results in the play $\mathbf{P}\sigma = \sigma$, so strategies are 'played out' by the **play function**

$$\mathbf{P} = \mathrm{id} : \prod_{i:I} X_i \to \prod_{i:I} X_i$$

The play $\mathbf{P}\sigma = \sigma$ is called the **strategic play** of the strategy σ, and is the play that results from all players playing according to σ. Each strategy σ additionally determines an outcome $q(\mathbf{P}\sigma) = q\sigma$.

We will next define **unilateral continuations**, which have proved to be a useful tool for reasoning about higher order games. They were introduced in [Hed13], in which it is shown that the majority of the proof of Nash's theorem amounts to showing that the unilateral continuations satisfy certain topological properties. They are also heavily used in [HOS+15b].

Suppose we have a fixed strategy profile σ. We can now define unilateral continuations in which all but one player use σ, and the remaining player **unilaterally deviates** to some other move. The ith player's unilateral continuation from σ is the function

$$\mathscr{U}_i^q \sigma : X_i \to R$$

given by

$$\mathscr{U}_i^q \sigma x_i = q(x_i, \sigma_{-i})$$

The notation (x_i, σ_{-i}), which we will now introduce here, is standard in game theory and useful for reasoning about unilateral deviation. The subscript in σ_{-i} means that we project σ onto the subspace $\prod_{j \neq i} X_j$. We will sometimes write the projection operator as

$$\pi_{-i} : \prod_{j=1}^{N} X_j \to \prod_{\substack{1 \leq j \leq N \\ j \neq i}} X_j$$

The notation (x_i, σ_{-i}) fills the 'missing' ith entry with x_i, and is defined by the equation

$$(x_i, \sigma_{-i})_j = \begin{cases} x_i & \text{if } i = j \\ \sigma_j & \text{otherwise} \end{cases}$$

Although this could be ambiguous and is often disliked by those who strive for type safety, this notation will come into its own in §3.1 and §3.2.

The purpose of a unilateral continuation is that the behaviour of all other players has been abstracted into a single function, allowing us to reduce a game-theoretic problem to a decision-theoretic one.

1.2.3 Nash equilibria of normal form games

The preferences of the player $i : I$ can be modelled either using a multi-valued quantifier

$$\Phi_i : (X_i \to R) \to \mathscr{P} R$$

or a multi-valued selection function

$$E_i : (X_i \to R) \to \mathscr{P} X_i$$

In any context $k : X_i \to R$, the value of the quantifier $\Phi_i k$ is the set of outcomes that player i considers to be good in the context k. Similarly, if we use selection functions then $E_i k$ is the set of moves that player i considers to be good in the context.

The unilateral continuation $\mathscr{U}_i^q \sigma : X_i \to R$ is the context in which player i is unilaterally deviating from the strategy profile σ. If the player is implemented by a multi-valued quantifier then the set of outcomes that the player considers good, and can be attained by unilaterally deviating, is $\Phi_i(\mathscr{U}_i^q \sigma)$. The outcome that actually occurs if the player does not deviate is $q(\mathbf{P}\sigma) = q\sigma$. Therefore, if

$$q\sigma \in \Phi_i(\mathscr{U}_i^q \sigma)$$

then the player is already satisfied with the outcome and has no incentive to unilaterally deviate. If this condition holds for each player $i : I$ then we will call σ a **quantifier equilibrium**.

On the other hand, if we implement players using multi-valued selection functions, then we have a set of good moves $E_i(\mathscr{U}_i^q \sigma)$. This is the set of moves which it would be rational for player i to choose, given that all other players use σ. The actual move chosen by player i is σ_i, so player i has no incentive to unilaterally deviate if

$$\sigma_i \in E_i(\mathscr{U}_i^q \sigma)$$

If this condition holds for all $i : I$ then we call σ a **selection equilibrium**.

If we could model the players either by quantifiers Φ_i, or by selection functions E_i that attain Φ_i (in the sense of §1.1.7), then every selection equilibrium is a quantifier equilibrium. To see this, if $\sigma_i \in E_i(\mathscr{U}_i^q \sigma)$ and E_i attains Φ_i, then

$$q\sigma = \mathscr{U}_i^q \sigma \sigma_i \in \Phi_i(\mathscr{U}_i^q \sigma)$$

We can therefore say that the selection equilibrium is an **equilibrium refinement** of the quantifier equilibrium. In particular, if we have players modelled by selection functions E_i then we can define quantifier equilibria using the quantifiers $\overline{E_i}$.

The converse of this holds, and hence quantifier and selection equilibria coincide, if our selection functions are of the form $E_i = \overline{\Phi_i}$, where the overline operator is the one defined in §1.1.7. To see this, suppose σ is a quantifier equilibrium, so we have

$$q\sigma = \mathscr{U}_i^q \sigma \sigma_i \in \Phi_i(\mathscr{U}_i^q \sigma)$$

Then $\sigma \in \overline{\Phi_i}(\mathscr{U}_i^q \sigma)$ and hence we have a selection equilibrium for the selection functions $\overline{\Phi}_i$.

Given a strategy profile σ, we will say that player i is **non-pivotal** if the unilateral continuation $\mathscr{U}_i^q \sigma : X_i \to R$ is constant. If σ is a quantifier equilibrium then any deviation (x_i, σ_{-i}) by a non-pivotal player results in another quantifier equilibrium. Selection equilibria in general do not share this property.

1.2.4 Best responses

An important concept in the foundations of game theory is that of a **best response function**. This is a function $\mathbf{B} : \Sigma \to \mathscr{P}\Sigma$, where Σ is the type of strategy profiles of a game, such that the fixpoints of $\mathbf{B}$ pick out some solution concept. The informal specification of a game's best response function is that $\mathbf{B}\sigma$ should be the set of strategy profiles σ' such that for each player i, it is rational for player i to use the strategy σ'_i, in the sense of having no incentive to unilaterally deviate, under the assumption that the other players are using σ_{-i}.

For a normal form game as defined in §1.2.2, we can define the best response function

$$\mathbf{B} : \prod_{i:I} X_i \to \mathscr{P} \prod_{i:I} X_i$$

in two different ways, which lead to quantifier and selection equilibria. If our players are defined by multi-valued quantifiers $\Phi_i : (X_i \to R) \to \mathscr{P}R$ then we can use the definition

$$\mathbf{B}\sigma = \left\{ \sigma' : \prod_{i:I} X_i \;\middle|\; \mathscr{U}_i^q \sigma \sigma'_i \in \Phi_i(\mathscr{U}_i^q \sigma) \text{ for all } i : I \right\}$$

A fixpoint of this $\mathbf{B}$ is a strategy profile satisfying

$$\mathscr{U}_i^q \sigma \sigma_i \in \Phi_i(\mathscr{U}_i^q \sigma)$$

for each $i : I$. Since $\mathscr{U}_i^q \sigma \sigma_i = q\sigma$, this is precisely the definition of a quantifier equilibrium.

Alternatively, if our players are specified by multi-valued selection functions $E_i : (X_i \to R) \to \mathscr{P}X_i$, we define the best response function by

$$\mathbf{B}\sigma = \prod_{i:I} E_i(\mathscr{U}_i^q \sigma)$$

A fixpoint $\sigma \in \mathbf{B}\sigma$ of this satisfies

$$\sigma_i \in E_i(\mathscr{U}_i^q \sigma)$$

for each $i : I$, and so is a selection equilibrium.

There are two reasons for focussing on best response functions. A technical reason is that it is sometimes possible to prove existence theorems for a solution concept by applying a fixpoint theorem to a suitable best response function. Nash's theorem has this form, but this is not something that will be emphasised in this thesis.

A second, more philosophical reason is that having the same best response function seems to be a necessary condition to consider two games to be *equivalent*. Although there is no formal theory of this, in §3.1 and §3.2 we will use equality of best response functions as an informal criterion for two games to be considered the same.

1.2.5 Classical games

In classical normal form games, each play determines a real number for each player called that player's **utility**. The utility is a numerical rating of the player's preference for that play, and players with rational preferences, in the sense of the von Neumann-Morgenstern utility theorem [vNM44], act *as if* they maximise some real utility. It is important to realise that utilities need not *physically* exist, and in particular utility should not be identified with monetary profit (which is usually called 'payoff'). This point is made strongly in section 1.1 of [Kre90].

In this setting, the outcome of a play consists of a real number for each player, and so we take $R = \mathbb{R}^I$, with the outcome function having type

$$q : \prod_{i:I} X_i \to \mathbb{R}^I$$

We can think of each player $i : I$ as having her own personal outcome function

$$q_i : \prod_{j:I} X_j \to \mathbb{R}$$

defined by $q_i = \pi_i \circ q$ where π_i is the ith projection $\mathbb{R}^I \to \mathbb{R}$.

Player i acts to maximise the ith coordinate, and so we could model her with the maximising quantifier

$$\Phi_i : (X_i \to \mathbb{R}^I) \to \mathscr{P}\mathbb{R}^I$$

given by

$$\Phi_i k = \{kx \mid k_i x \geq k_i x' \text{ for all } x' : X_i\}$$

or alternatively with the maximising selection function

$$E_i : (X_i \to \mathbb{R}^I) \to \mathscr{P} X_i$$

given by

$$E_i k = \{x : X_i \mid k_i x \geq k_i x' \text{ for all } x' : X_i\}$$

(see §1.1.8).

Notice that $E_i = \overline{\Phi_i}$, and hence quantifier and selection equilibria coincide for a classical game. An equilibrium σ (of either kind) satisfies the conditions, for each $i : I$, that

$$q_i \sigma \geq \mathscr{U}_i^{q_i} \sigma x'$$

for each possible unilateral deviation $x' : X_i$. Since $\mathscr{U}_i^{q_i} \sigma x' = q_i(x', \sigma_{-i})$, we see that this is precisely the ordinary definition of a pure strategy Nash equilibrium [LBS08]. Thus, our two solution concepts both coincide with the ordinary one in the case of classical games.

The best response function for a classical normal form game is

$$\mathbf{B}\sigma = \left\{ \sigma' : \prod_{i:I} X_i \;\middle|\; q_i(\sigma'_i, \sigma_{-i}) \geq q_i(x'_i, \sigma_{-i}) \text{ for all } i : I \text{ and } x'_i : X_i \right\}$$

We get this irrespective of whether we use the quantifier or selection function forms, and fixpoints of this $\mathbf{B}$ give precisely the pure strategy Nash equilibria.

The same reasoning applies if we replace $\mathbb{R}^I$ with an arbitrary set R with a **rational preference relation** $\succeq$. A rational preference relation is another name for a total preorder, that is, a relation that is transitive and total but not necessarily antisymmetric. We will write $a \approx b$ if we have $a \preceq b$ and $b \preceq a$ for some $a \neq b : R$. For example, if we take $\mathbb{R}^I$ with the order $a \preceq_i b$ iff $a_i \leq b_i$, then $\preceq_i$ is a rational preference relation which has $a \approx b$ iff $a_i = b_i$.

A rational preference relation on R is the same as a total order on $\approx$-equivalence classes. Thus, given a continuation $k : X \to R$, we have a set of optimal outcomes, namely the maximal equivalence class

$$\Phi k = \{kx \mid kx \succeq kx' \text{ for all } x' : X\}$$

This defines a multi-valued quantifier, and similarly we have a multi-valued selection function attaining it.

1.2.6 Mixed strategies

In this section we will turn aside from the development of higher order game theory in order to introduce mixed strategies, which will be used in §3.1. It is still an open problem how mixed strategies should be modelled in the higher order framework.

Consider a classical normal-form game with a finite number of players, labelled $1 \leq i \leq N$, so the outcome function has type

$$q : \prod_{i=1}^{N} X_i \to \mathbb{R}^N$$

We will also assume that the sets X_i are finite. A **mixed strategy** for player i is a probability distribution on X_i. We will write this as $\sigma_i : \mathscr{D} X_i$, where the probability distribution operator $\mathscr{D}$ is properly introduced in §2.1.3. A **mixed strategy profile** is a tuple

$$\sigma : \prod_{i=1}^{N} \mathscr{D} X_i$$

Given a mixed strategy profile σ, for each player i we obtain a probability distribution on utilities. This can be used to obtain an **expected utility** for the ith player, which is given by

$$\mathbb{E}[q_i \sigma] = \sum_{x : \prod_{i=1}^{N} X_i} \left(q_i x \cdot \prod_{i=1}^{N} \mathbb{P}\left[\sigma_i = x_i\right] \right) : \mathbb{R}$$

where we use the fact that the X_i are finite to ensure convergence. The notation $q_i \sigma$ means the application of the non-stochastic function q_i to the random variable σ. Formally, the functor $\mathscr{D}$ is acting on q_i (see §2.1.3).

A **mixed strategy Nash equilibrium** is a mixed strategy profile in which no player can increase her expected utility by unilaterally deviating to some other move. Equivalently, a mixed strategy Nash equilibrium is a fixpoint of the best response function

$$\mathbf{B} : \prod_{i=1}^{N} \mathscr{D} X_i \to \mathscr{P} \prod_{i=1}^{N} \mathscr{D} X_i$$

given by

$$\mathbf{B}\sigma = \left\{ \sigma' : \prod_{i=1}^{N} \mathscr{D}X_i \;\middle|\; \mathbb{E}[q_i(\sigma'_i, \sigma_{-i})] \geq \mathbb{E}[q_i(x'_i, \sigma_{-i})] \right.$$
$$\left. \text{for all } 1 \leq i \leq N \text{ and } x'_i : X_i \right\}$$

Nash's famous existence theorem, namely that mixed strategy equilibria always exist under only the finiteness condition we have given, is proved in [Nas51] by applying the Kakutani fixpoint theorem [Kak41] to this function **B**. A generalisation of this method for certain higher order games was given in [Hed13]. However, there is no known sense in which we can consider mixed equilibria of an arbitrary higher order game. This will be called the **problem of mixed extensions**. For example, the passage from the deterministic to the probabilistic arg max operator appears not to be categorically natural, and for other selection functions it is unclear what the probabilistic equivalent should even be.

1.2.7 Voting games

We are now going to introduce an extended example from [HOS+15b], which illustrates the use of selection functions that are different from arg max, and argues that selection equilibrium (as supposed to quantifier equilibrium) is the *correct* solution concept for higher order games in normal form.

We will consider an election with three voters and two candidates. The set of candidates is $X = \{a, b\}$. We will consider the election as a game in which the three voters simultaneously choose a candidate. The outcome of the game is precisely the candidate who received the most votes. Thus, we take the set of outcomes to also be X, and the outcome function to be the majority function $q : X^3 \to X$.

We will denote the three voters by $i = 1, 2, 3$. The preferences of the voters are defined by multi-valued selection functions $E_i : (X \to X) \to \mathscr{P}X$, or by the multi-valued quantifiers $\overline{E_i}$. We will investigate how the selection and quantifier equilibria vary as we choose different combinations of selection functions.

If player i is *rational*, then by definition she has a rational preference relation $\succeq_i$ on X. Since X contains two elements, there are precisely three rational preference relations, where respectively a is preferred to b, b is preferred to a, and both are equally preferred. The last case, when player i is indifferent between a and b, is described by the multi-valued selection function

$$E_{a \approx b} k = X$$

that always returns the set of all moves. For the remaining two cases, we can fix an ordering $a \succeq b$ and write the corresponding selection functions as arg max and arg min with respect to this ordering. We can get all possible rational behaviours by choosing E_1, E_2 and E_3 from arg max, arg min and $E_{a \approx b}$.

As a first example, take the three selection functions to be $E_1 = \arg\max$, $E_2 = \arg\min$ and $E_3 = E_{a \approx b}$, so the first judge prefers a, the second prefers b and the third is indifferent. Consider a strategy profile $\sigma = (\sigma_1, \sigma_2, \sigma_3) : X^3$. For each of these, we can calculate for $i = 1, 2, 3$ the unilateral contexts $\mathscr{U}_i^q\sigma : X \to X$, namely

$$\mathscr{U}_1^q\sigma x_1 = q(x_1, \sigma_2, \sigma_3)$$
$$\mathscr{U}_2^q\sigma x_2 = q(\sigma_1, x_2, \sigma_3)$$
$$\mathscr{U}_3^q\sigma x_3 = q(\sigma_1, \sigma_2, x_3)$$

Using the definition in §1.2.3, we see that σ is a selection equilibrium iff

$$\sigma_1 \in \arg\max(\mathscr{U}_1^q\sigma) = \arg\max_{x_1 : X} q(x_1, \sigma_2, \sigma_3)$$

$$\sigma_2 \in \arg\min(\mathscr{U}_2^q\sigma) = \arg\min_{x_2 : X} q(\sigma_1, x_2, \sigma_3)$$

$$\sigma_3 \in E_{a \approx b}(\mathscr{U}_3^q\sigma) = X$$

where the third condition is trivial.

Since there are only 8 strategy profiles, we can simply enumerate them and calculate the equilibria by brute force. We will begin by checking that the obvious strategies (a, b, a) and (a, b, b), where each player votes for her preferred candidate, are selection equilibria. In this case the unilateral contexts are

$$\mathscr{U}_1^q(a, b, a)x_1 = q(x_1, b, a) = x_1$$

$$\mathscr{U}_2^q(a, b, a)x_2 = q(a, x_2, a) = a$$

Applying the selection functions gives

$$\arg\max(\mathscr{U}_1^q(a, b, a)) = \arg\max_{x_1 : X} x_1 = \{a\}$$

$$\arg\min(\mathscr{U}_2^q(a, b, a)) = \arg\min_{x_2 : X} a = \{a, b\}$$

Since $a \in \arg\max(\mathscr{U}_1^q(a, b, a))$ and $b \in \arg\min(\mathscr{U}_2^q(a, b, a))$, we have a selection equilibrium. We can similarly verify that (a, b, b) is a selection equilibrium.

1.2.8 Modelling with selection functions

However, this game has additional equilibria that are less plausible. Consider for example the strategy (a, a, a). Classically, this is an equilibrium because the second player, who is the only player who might have incentive to deviate, is in fact not pivotal and so cannot increase her utility with any deviation. This is also captured by the selection equilibrium, because the second player's unilateral context is the constant function

$$\mathscr{U}_2^q(a, a, a)x_2 = q(a, x_2, a) = a$$

The minimum value that $q(a, x_2, a)$ can take is a, and this minimum is attained at $x_2 = a, b$. Thus we have $\arg\min(\mathscr{U}_2^q(a, a, a)) = \{a, b\}$, and since $a \in \arg\min(\mathscr{U}_2^q(a, a, a))$ we see that player 2 has no incentive to unilaterally deviate.

If we as a modeller decide that this is implausible, one classical method is to modify the preferences of the players. In the initial formulation, the preferences of player 2 are given by a rational preference relation $a \preceq_2 b$ on the type of outcomes $R = X$. However, we can pass from this to a rational preference relation $\preceq'_2$ on the type of plays $X \times X \times X$, given by

$$(x_1, x_2, x_3) \preceq'_2 (x'_1, x'_2, x'_3) \iff q(x_1, x_2, x_3) \preceq_2 q(x'_1, x'_2, x'_3)$$

Thus, a play is preferred iff it leads to a preferred outcome. In this preference order, we have implausible equivalences such as $(a, a, a) \approx'_2 (a, b, a)$. The solution is to manually modify the preference relation. For example, we could replace $\preceq'_2$ with a lexicographic relation $\preceq''_2$ which first takes preferred outcomes, and then if the outcomes are equivalent, uses the preference on moves $b \succeq_2 a$. This models a player who will always vote b, except in the strange context in which voting a leads b to win the contest, and voting b leads a to win.

When using selection functions, the idiomatic way to approach this problem is to use selection refinements (§1.1.7). Specifically, we refine player 2's multi-valued selection function $\arg\min : (X \to X) \to \mathscr{P}X$ to the single-valued selection function

$$\varepsilon_2 k = \begin{cases} b & \text{if } b \in \arg\min k \\ a & \text{otherwise} \end{cases}$$

Much more detail about this is given in [HOS+15b]. A particular advantage of higher order game theory that can be exploited is that we never need to assume that the notion of rationality implemented by a selection function is equivalent to maximising over a rational preference relation, so rational and non-rational behaviour can be treated on an equal footing.

1.2.9 Coordination and differentiation

The main contribution of [HOS+15b] is the demonstration that coordinating and differentiating behaviour of players in the voting game can be modelled by selection functions that do not arise from rational preference relations, specifically fixpoint and non-fixpoint operators. Specifically, coordinating behaviour is modelled by the selection function $E_{\text{fix}} : (X \to X) \to \mathscr{P}X$ given by

$$E_{\text{fix}} k = \begin{cases} \{x : X \mid x = kx\} & \text{if nonempty} \\ X & \text{otherwise} \end{cases}$$

and differentiating behaviour by

$$E_{\text{nonfix}} k = \begin{cases} \{x : X \mid x \neq kx\} & \text{if nonempty} \\ X & \text{otherwise} \end{cases}$$

Notice that $\{x : X \mid x \neq kx\}$ is empty iff k is the identity function $X \to X$.

In the scenario of the voting game, a player modelled by E_{fix} aims to *vote for the winner*, and as such they are a **Keynesian agent**, after [Key36, chapter 12]. Keynes' example of coordinating preferences in economics is as follows: suppose for simplicitly that the financial success of a company is dependent only on the number of investors it attracts. An investor, therefore, has no preferences over different companies, but rather must aim to go with the majority of other investors. Similarly, a player modelled by E_{nonfix} is a *punk*, whose only aim is to vote against the majority.

These selection functions are total variants of their refinements

$$E'_{\text{fix}}k = \{x : X \mid x = kx\}$$

$$E'_{\text{nonfix}}k = \{x : X \mid x \neq kx\}$$

which are not total (see §1.1.6). The intuition behind returning X in the exceptional case, rather than $\varnothing$ or some arbitrary default value such as $\{a\}$, is that in this case the player is indifferent and hence should be satisfied with any choice. In practice this has been checked by brute force, by showing that these exact definitions of E_{fix} and E_{nonfix} agree with the informal specification of a player who aims to vote with or against the majority.

Unlike arg max and arg min, the selection functions E_{fix} and E_{nonfix} are not closed (see §1.1.7). To give a specific counterexample, consider the context $k : X \to X$ given by $kx = a$. The fixpoints of k are $E_{\text{fix}}k = \{a\}$. Then

$$\overline{E_{\text{fix}}}k = \{kx \mid x \in \{a\}\} = \{a\}$$

and

$$\overline{\overline{E_{\text{fix}}}}k = \{x : X \mid kx \in \{a\}\} = \{a, b\}$$

Thus $E_{\text{fix}}k \neq \overline{\overline{E_{\text{fix}}}}k$. Similarly,

$$E_{\text{nonfix}}k = \{b\} \neq \{a, b\} = \overline{\overline{E_{\text{nonfix}}}}k$$

This relates to the choices not being pivotal in the constant context (see §1.2.3).

1.2.10 Illustrating the solution concepts

As a second example, we will take $E_1 = E_{\text{fix}}$, $E_2 = E_{\text{nonfix}}$ and $E_3 = \arg\max$. Thus we have a Keynesian agent, a punk, and a rational player who prefers a. This game has 4 selection equilibria, namely

$$(a, a, b), (a, b, a), (b, b, a), (b, b, b)$$

Of these, (a, a, b) and (b, b, b) are implausible and can be ruled out by refining arg max as described in §1.2.8. With (a, b, a) all three voters have achieved their aim: the first is in the majority, the second is in the minority and the third has her preferred outcome. In (b, b, a) player 2 is unable to be in the minority because she is pivotal, and so she is satisfied with the choice of b. Player 2's unilateral context is

$$\mathscr{U}_2^q(b, b, a)x_2 = q(b, x_2, a) = x_2$$

and $\mathscr{U}_2^q(b, b, a)$ has no non-fixpoints, and so $E_{\text{nonfix}}(\mathscr{U}_2^q(b, b, a)) = \{a, b\}$ by definition. This selection equilibrium could be ruled out if desired out by refining E_{nonfix} to return the actual set of non-fixpoints, even if empty.

Now, suppose we define the preferences instead by the multi-valued quantifiers $\overline{E_1}$, $\overline{E_2}$, $\overline{E_3}$. The four selection equilibria remain as quantifier equilibria (see §1.2.3), but there are two additional quantifier equilibria which are not selection equilibria, namely (a, a, a) and (b, a, a). We will focus on the latter as an example. The unilateral context for the first player with this strategy profile is

$$\mathscr{U}_1^q(b, a, a)x_1 = q(x_1, a, a) = a$$

The strategy profile is not a selection equilibrium because player 1's choice is not a fixpoint, in other words,

$$b \notin \{a\} = E_{\text{fix}}(\mathscr{U}_1^q(b, a, a))$$

However, the quantifier equilibrium uses outcomes rather than choices, and the outcome is a fixpoint:

$$q(b, a, a) = a \in \{a\} = \overline{E_{\text{fix}}}(\mathscr{U}_1^q(b, a, a))$$

According to the definition in §1.2.3, taking $\Phi_i = \overline{E_i}$, this makes (b, a, a) a quantifier equilibrium. Alternatively, this can be seen as player 1 not being pivotal (§1.2.3) and deviating from the more plausible equilibrium (a, a, a).

For a third example, we will take $E_1 = E_2 = E_3 = E_{\text{fix}}$, giving a game in which no players have preferences over the candidates but all aim to vote with the majority. Intuitively the equilibria of this game should be (a, a, a) and (b, b, b), and indeed these are exactly the selection equilibria. However, every strategy profile of this game is a quantifier equilibrium. This is essentially a tautology: no matter how the players vote, the majority choice is in the majority. This is an example where the selection equilibrium makes a useful prediction that agrees with intuition, but the quantifier equilibrium makes no prediction.

For a final example, take $E_1 = E_2 = E_3 = E_{\text{nonfix}}$. This models a population consisting entirely of punks. Of course, in any particular play a majority of the punks will always fail in their aim to be in the minority. The selection equilibria of this game are precisely the 'maximally differentiated' strategy profiles, namely all strategy profiles except for (a, a, a) and (b, b, b). Once again, every strategy profile is a quantifier equilibrium, because

$$\overline{E_{\text{nonfix}}} k = \begin{cases} \{kx \mid x \neq kx\} & \text{if nonempty} \\ \operatorname{Im} k & \text{otherwise} \end{cases}$$

If we take for example player 1 with the strategy profile (a, a, a), the unilateral context is $\mathscr{U}_1^q(a, a, a)x_1 = a$, and so

$$\overline{E_{\text{nonfix}}}(\mathscr{U}_1^q(a, a, a)) = \{\mathscr{U}_1^q(a, a, a)b\} = \{q(b, a, a)\} = \{a\}$$

This gives us a second example in which the selection equilibrium makes a useful and intuitive prediction, whereas the quantifier equilibrium makes no prediction.

On the basis of these examples, we choose to use selection equilibria rather than quantifier equilibria as the default solution concept for higher order games. This is essentially the argument of [HOS+15b]. This approach is more general than [EO11], which (naively) takes the quantifier equilibrium as its solution concept.

1.3 Sequential games

1.3.1 Discussion

The theory of sequential games defined by selection functions was the starting point of higher order game theory. It was first developed in [EO10a], with the presentation in this section based on [EO11]. In particular it is not the work of the author, but is included for completeness.

The type of single-valued selection functions, introduced in §1.1.4, carries the structure of a strong monad. Moreover the type of single-valued quantifiers carries a different strong monad structure, namely the continuation monad (see §1.1.3). The key result about higher order games is that the monad operations on these types are compatible with the game-theoretic interpretation, and in particular there is a monoidal product operator that implements the backward induction algorithm (and which moreover extends to unbounded games), which is the standard method in game theory to calculate subgame perfect equilibria of games of perfect information.

We will use the term **sequential game** to refer to games of perfect information with the added restriction that the player choosing at each stage, and the set of moves available to her, is independent of the previously chosen moves. This rules out standard examples such as the market entry game.[1]

In a sequential game the players are ordered, with each player being able to observe the choices made by previous players before making her own choice. We can easily adapt the unilateral continuations from §1.2.2 in order to define Nash equilibria for sequential games. However, sequential games can have implausible Nash equilibria in which players make so-called incredible threats, in which a player can rule out branches of the game tree by contingently making a mutually destructive move. Subgame perfect equilibria rule this out, by forcing play to be rational even in branches that are not reached by playing the strategy. Every sequential game with finitely many players has at least one subgame perfect equilibrium, a result that is proved constructively using backward induction.

The method in this chapter of computing subgame perfect equilibria using the product of selection functions generalises to games with infinitely many players, so long as the outcome function is topologically continuous. These 'unbounded sequential games' are crucial to the application of selection functions in proof theory [Pow13, OP14, OP15], and are discussed from a game-theoretical point of view in [EO12]. However, backward induction for unbounded games relies crucially on subtle aspects of higher type computability, namely bar recursion, and for simplicity in this thesis we will focus only on games with finitely many players.

1.3.2 The category of selection functions

Just as quantifiers form the continuation monad $\mathscr{K}_R$, so selection functions form the **selection monad** $\mathscr{J}_R$ where

$$\mathscr{J}_R X = (X \to R) \to X$$

Explicitly, the unit maps $\eta : X \to \mathscr{J}_R X$ are given by

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

and the Kleisli extension of $f : X \to \mathscr{J}_R Y$ is

$$f^* \varepsilon = \lambda(k : Y \to R).f(\varepsilon\lambda(x : X).k(fxk))k$$

[1] Generally speaking, type theoretic approaches to game theory have difficulty with this sort of example. Multi-agent influence diagrams [KM03], for example, cannot represent the market entry game. This exact issue is the focus of [BIB13].

It is difficult to give a direct proof that $\mathscr{J}_R$ is a monad, and the proof in [EO10a], which also applies to the selection monad transformer in [Hed14], uses the fact that the monad laws are equivalent to the category axioms for the Kleisli category. This proof is simple enough to reproduce here.

Given a locally small category $\mathcal{C}$ and a fixed object R, we will define a new category $\mathscr{J}_R\mathcal{C}$ called a **selection category**. The objects of $\mathscr{J}_R\mathcal{C}$ are exactly the objects of $\mathcal{C}$, and the hom-sets are given by

$$\hom_{\mathscr{J}_R\mathcal{C}}(X, Y) = \hom_{\mathcal{C}}(Y, R) \to \hom_{\mathcal{C}}(X, Y)$$

In particular, using the fact that **Set** is enriched in itself gives

$$\hom_{\mathscr{J}_R\mathbf{Set}}(X, Y) = (Y \to R) \to (X \to Y) \cong X \to \mathscr{J}_R Y$$

The identity on X is given by

$$\mathrm{id}_X k = \mathrm{id}_X$$

where the left hand id_X is the identity in $\hom_{\mathscr{J}_R\mathcal{C}}(X, X)$, and the right hand is in $\hom_{\mathcal{C}}(X, X)$. The composition of $\varepsilon : \hom_{\mathcal{C}}(Y, R) \to \hom_{\mathcal{C}}(X, Y)$ and $\delta : \hom_{\mathcal{C}}(Z, R) \to \hom_{\mathcal{C}}(Y, Z)$ is given by

$$(\delta \circ \varepsilon)k = \delta k \circ \varepsilon(k \circ \delta k)$$

The category axioms can be proved very compactly. The left unit law is

$$(\mathrm{id}_Y \circ \varepsilon)k = \mathrm{id}_Y k \circ \varepsilon(k \circ \mathrm{id}_Y k) = id_Y \circ \varepsilon(k \circ \mathrm{id}_Y) = \varepsilon k$$

The right unit law is

$$(\varepsilon \circ \mathrm{id}_X)k = \varepsilon k \circ \mathrm{id}_X(k \circ \varepsilon k) = \varepsilon k \circ \mathrm{id}_X = \varepsilon k$$

and the associativity law is

$$\begin{aligned}
(\gamma \circ (\delta \circ \varepsilon))k &= \gamma k \circ (\delta \circ \varepsilon)(k \circ \gamma k) \\
&= \gamma k \circ \delta(k \circ \gamma k) \circ \varepsilon(k \circ \gamma k \circ \delta(k \circ \gamma k)) \\
&= (\gamma \circ \delta)k \circ \varepsilon(k \circ (\gamma \circ \delta)k) \\
&= ((\gamma \circ \delta) \circ \varepsilon)k
\end{aligned}$$

Notice that the selection category does not capture the strong monad structure of $\mathscr{J}_R$ (although this could probably be achieved by using enriched category theory more carefully). However, $\mathscr{J}_R\mathcal{C}$ can also be given a premonoidal product, which we will now introduce.

1.3.3 The product of selection functions

In §1.1.4 we introduced single-valued selection functions as elements of the type

$$\varepsilon : \mathscr{J}_R X = (X \to R) \to X$$

In §1.3.2 we proved that $\mathscr{J}_R$ is a monad. In [EO10a] it is moreover proved that $\mathscr{J}_R$ is a strong monad. In particular this means that it is a monoidal monad, in the sense that it has a premonoidal[2] product operator

$$\ltimes : \mathscr{J}_R X \times \mathscr{J}_R Y \to \mathscr{J}_R (X \times Y)$$

This operator is called the binary **product of selection functions**, and is explicitly defined by

$$(\varepsilon \ltimes \delta)k = (a, ba)$$

where

$$\begin{aligned} a &= \varepsilon \lambda(x : X).k(x, bx) \\ bx &= \delta \lambda(y : Y).k(x, y) \end{aligned}$$

This can be folded to any finite number of selection functions $\varepsilon_j : \mathscr{J}_R X_j$, by

$$\bigotimes_{j=i}^{N} \varepsilon_j = \varepsilon_i \ltimes \bigotimes_{j=i+1}^{N} \varepsilon_j$$

with the base case $\bigotimes_{j=N}^{N} \varepsilon_j = \varepsilon_N$.

1.3.4 Sequential games

Since our players now choose sequentially, we will number them $1, \ldots, N$. Just as for normal form games (§1.2.2), a **sequential game** consists of choice types X_i, and an outcome function

$$q : \prod_{i=1}^{N} X_i \to R$$

where R is a type of outcomes.

We will, similarly, model the players by multi-valued selection functions

$$E_i : (X_i \to R) \to \mathscr{P} X_i$$

This is the role taken by single-valued quantifiers in [EO10a] and by multi-valued quantifiers in [EO11]. Since the same reasoning about coordination and differentiation games applies as in §1.2.10 and [HOS+15b], we will use multi-valued selection functions rather than quantifiers.

The first difference between sequential and normal form games is that strategies become nontrivial. When player i makes her choice, she can directly observe the choices made by all players $j < i$. Thus, player i's strategy is a function

$$\sigma_i : \prod_{j<i} X_j \to X_i$$

and the type of strategy profiles is

$$\Sigma = \prod_{i=1}^{N} \left(\prod_{j<i} X_j \to X_i \right)$$

[2] The product of selection functions is written $\otimes$ elsewhere in the literature, but we will write it as $\ltimes$ to emphasise that it is noncommutative, and avoid confusion with the tensor product of open games introduced in §2.2.7, which is very different. This notation for a premonoidal product is from [PR93].

A play, however, is still a tuple in $\prod_{i=1}^{N} X_i$.

Given a strategy, we can play it to obtain its **strategic play**. Notice that if $\sigma : \Sigma$ then σ_1 is simply a choice $\sigma_1 : X_1$, because the tuple $\prod_{j<1} X_j$ is the unit type. Similarly $\sigma_2 : X_1 \to X_2$, so we can apply σ_2 to σ_1 to obtain the second player's choice, $\sigma_2 \sigma_1 : X_2$. This can be extended by course-of-values recursion:

$$\mathbf{P} : \Sigma \to \prod_{i=1}^{N} X_i$$
$$(\mathbf{P}\sigma)_i = \sigma_i((\mathbf{P}\sigma)_1, \ldots, (\mathbf{P}\sigma)_{i-1})$$

1.3.5 Subgame perfection

A **partial play** is defined to be a tuple

$$x_1, \ldots, x_{i-1} : \prod_{j<i} X_j$$

for some $1 \leq i \leq N$. Notice that the empty sequence is considered a partial play, but a play is not considered a partial play. Partial plays are in bijection with **subgames**, which are games that can be obtained by replacing some initial segment of players by fixed choices.

Given a strategy profile σ, we can extend a partial play to a play called its **strategic extension** by σ. This is similar to a unilateral deviation, except that instead of a single player i deviating, all players $j \leq i$ deviate. In the strategic extension ν_x^σ of $x = x_1, \ldots, x_{i-1}$ by σ, the first $i-1$ players are forced to use the partial play, but subsequent players use σ. The definition is by the course-of-values recursion

$$(\nu_x^\sigma)_j = \begin{cases} x_j & \text{if } j < i \\ \sigma_j((\nu_x^\sigma)_1, \ldots, (\nu_x^\sigma)_{j-1}) & \text{if } j \geq i \end{cases}$$

If $i = 1$ then x is the (unique) partial play of length 0, in which case $\mathbf{P}\sigma = \nu^\sigma$.

We will now modify the unilateral continuations introduced in §1.2.2 for use with sequential games. Instead of simply having a single function

$$\mathscr{U}_i^q : \Sigma \to X_i \to R$$

for the ith player's unilateral continuation, we must instead have one for each partial play of length $i-1$. This will be defined by

$$\mathscr{U}_x^q \sigma x_i = q(\nu_{x,x_i}^\sigma)$$

for $x = x_1, \ldots, x_{i-1}$.

A **subgame perfect equilibrium** is a strategy profile σ that is an equilibrium in every subgame. That is, for every partial play $x = x_1, \ldots, x_{i-1}$, the next move played by σ, namely $\sigma_i x$, is rational according to the selection function E_i, in the context in which player i unilaterally deviates in the subgame induced by x. Explicitly, σ must satisfy the conditions

$$\sigma_i x \in E_i(\mathscr{U}_x^q \sigma)$$

for all partial plays $x = x_1, \ldots, x_{i-1}$.

Equivalently, a subgame perfect equilibrium is a fixpoint of the best response function $\mathbf{B} : \Sigma \to \mathscr{P}\Sigma$, where

$$\mathbf{B}\sigma = \{\sigma' : \Sigma \mid \sigma'_i x \in E_i(\mathscr{U}^q_x \sigma) \text{ for all } x = x_1, \ldots, x_{i-1}\}$$

1.3.6 Backward induction

We will now prove the theorem that deserves to be called the *fundamental theorem of higher order game theory*, that the product of selection functions computes a play that is rational according to subgame perfect equilibrium.

Suppose each E_i is refined by a single-valued selection function $\varepsilon_i : \mathscr{J}_R X_i$, in the sense of §1.1.7, so $\varepsilon_i k \in E_i k$ for all $k : X_i \to R$. Then

$$\left(\bigotimes_{i=1}^{N} \varepsilon_i \right) q$$

is the strategic play of a subgame perfect equilibrium.

This has been proven in [EO10a, EO11], but we will give the proof again here, because the technique is important and will be used again in §3.2.4 and §3.3.4. Specifically, we will prove that the strategy profile σ defined by the course-of-values recursion

$$\sigma_i(x_1, \ldots, x_{i-1}) = \varepsilon_i(\mathscr{U}^q_{x_1,\ldots,x_{i-1}} \sigma)$$

is subgame perfect, and that its strategic play is $\left(\bigotimes_{i=1}^{N} \varepsilon_i \right) q$. Subgame perfection is immediate, because for every partial play $x_1, \ldots, x_{i-1}$ we have

$$\sigma_i(x_1, \ldots, x_{i-1}) = \varepsilon_i(\mathscr{U}^q_{x_1,\ldots,x_{i-1}} \sigma) \in E_i(\mathscr{U}^q_{x_1,\ldots,x_{i-1}} \sigma)$$

For the second part, we will prove the stronger fact that for any partial play $x_1, \ldots, x_{i-1}$,

$$\left(\bigotimes_{j=i}^{N} \varepsilon_j \right) q_{x_1,\ldots,x_{i-1}} = (\nu^\sigma_{x_1,\ldots,x_{i-1}})_{j=i}^N$$

by strong induction on $N - i$, where

$$q_{x_1,\ldots,x_{i-1}} : \prod_{j=i}^{N} X_j \to R$$

is defined by

$$q_{x_1,\ldots,x_{i-1}}(x_i, \ldots, x_N) = q(x_1, \ldots, x_{i-1}, x_i, \ldots, x_N)$$

The original claim follows, because the empty partial play has the property

$$(\nu^\sigma)_{j=1}^N = \nu^\sigma = \mathbf{P}\sigma$$

This is the characteristic of the proof technique we are using: to prove a property of a sequential game, work by bar induction on the tree of subgames. For games with finitely many players, this is equivalent to fixing a partial play of length

$i-1$ and working by strong induction on $N-i$, ending eventually with the empty partial play, whose induced subgame is the game itself.

In the base case, we have a partial play $x_1, \ldots, x_{N-1}$, and

$$\left(\bigotimes_{j=N}^{N} \varepsilon_j \right) q_{x_1,\ldots,x_{N-1}} = \varepsilon_N q_{x_1,\ldots,x_{N-1}} = (\nu^\sigma_{x_1,\ldots,x_{N-1}})_{j=N}^N$$

because by the definition of $(\nu^\sigma_x)_j$ in §1.3.5,

$$\begin{aligned}(\nu^\sigma_{x_1,\ldots,x_{N-1}})_N &= \sigma_N((\nu^\sigma_{x_1,\ldots,x_{N-1}})_1, \ldots, (\nu^\sigma_{x_1,\ldots,x_{N-1}})_{N-1}) \\ &= \sigma_N(x_1, \ldots, x_{N-1})\end{aligned}$$

By the definition of σ this is $\varepsilon_N(\mathscr{U}^q_{x_1,\ldots,x_{N-1}}\sigma)$. Finally, we have

$$\mathscr{U}^q_{x_1,\ldots,x_{N-1}}\sigma = q_{x_1,\ldots,x_{N-1}}$$

because for all $x_N : X_N$,

$$\begin{aligned}\mathscr{U}^q_{x_1,\ldots,x_{N-1}}\sigma x_N &= q(\nu^\sigma_{x_1,\ldots,x_N}) \\ &= q((\nu^\sigma_{x_1,\ldots,x_N})_1, \ldots, (\nu^\sigma_{x_1,\ldots,x_N})_N) \\ &= q(x_1, \ldots, x_N) \\ &= q_{x_1,\ldots,x_{N-1}} x_N\end{aligned}$$

This completes the base case of the proof.

1.3.7 The inductive step

For the inductive step, we take a partial play $x_1, \ldots, x_{i-1}$. Unfolding the product of selection functions once gives

$$\left(\bigotimes_{j=i}^{N} \varepsilon_j \right) q_{x_1,\ldots,x_{i-1}} = \left(\varepsilon_i \ltimes \bigotimes_{j=i+1}^{N} \varepsilon_j \right) q_{x_1,\ldots,x_{i-1}} = (a, ba)$$

where

$$\begin{aligned}a &= \varepsilon_i \lambda(x_i : X_i).q_{x_1,\ldots,x_{i-1},x_i}(bx_i) \\ bx_i &= \left(\bigotimes_{j=i+1}^{N} \varepsilon_j \right) q_{x_1,\ldots,x_{i-1},x_i}\end{aligned}$$

The inductive hypothesis gives us

$$bx_i = (\nu^\sigma_{x_1,\ldots,x_{i-1},x_i})_{j=i+1}^N$$

for all $x_i : X_i$. We must prove that

$$(a, ba) = (\nu^\sigma_{x_1,\ldots,x_{i-1}})_{j=i}^N$$

which is to say that

$$\begin{aligned}a &= (\nu^\sigma_{x_1,\ldots,x_{i-1}})_i \\ ba &= (\nu^\sigma_{x_1,\ldots,x_{i-1}})_{j=i+1}^N\end{aligned}$$

We will first prove that for all $x_i : X_i$,

$$\mathscr{U}^q_{x_1,\ldots,x_{i-1}} \sigma x_i = q_{x_1,\ldots,x_{i-1},x_i}(bx_i)$$

The left hand side by definition is

$$\mathscr{U}^q_{x_1,\ldots,x_{i-1}} \sigma x_i = q(\nu^\sigma_{x_1,\ldots,x_{i-1},x_i})$$

and the right hand side, by the inductive hypothesis, is

$$q_{x_1,\ldots,x_{i-1},x_i}(bx_i) = q_{x_1,\ldots,x_{i-1},x_i}((\nu^\sigma_{x_1,\ldots,x_{i-1},x_i})^N_{j=i+1})$$

and these are equal because

$$\begin{aligned} q_{x_1,\ldots,x_{i-1},x_i}((\nu^\sigma_{x_1,\ldots,x_{i-1},x_i})^N_{j=i+1}) &= q(x_1,\ldots,x_{i-1},x_i,(\nu^\sigma_{x_1,\ldots,x_{i-1},x_i})^N_{j=i+1}) \\ &= q(\nu^\sigma_{x_1,\ldots,x_{i-1},x_i}) \end{aligned}$$

From this we immediately get

$$\begin{aligned} (\nu^\sigma_{x_1,\ldots,x_{i-1}})_i &= \sigma_i((\nu^\sigma_{x_1,\ldots,x_{i-1}})_1,\ldots,(\nu^\sigma_{x_1,\ldots,x_{i-1}})_{i-1}) \\ &= \sigma_i(x_1,\ldots,x_{i-1}) \\ &= \varepsilon_i(\mathscr{U}^q_{x_1,\ldots,x_{i-1}}\sigma) \\ &= \varepsilon_i \lambda(x_i : X_i).q_{x_1,\ldots,x_{i-1},x_i}(bx_i) \\ &= a \end{aligned}$$

which is the first of the two conditions to be proved.

In order to prove the second condition

$$ba = (\nu^\sigma_{x_1,\ldots,x_{i-1}})^N_{j=i+1}$$

we use the inductive hypothesis to get

$$ba = (\nu^\sigma_{x_1,\ldots,x_{i-1},a})^N_{j=i+1}$$

and so we will prove by strong induction on $j \geq i$ that

$$(\nu^\sigma_{x_1,\ldots,x_{i-1}})_j = (\nu^\sigma_{x_1,\ldots,x_{i-1},a})_j$$

(Notice that this also trivially holds for $j < i$.) In the base case we have

$$(\nu^\sigma_{x_1,\ldots,x_{i-1},a})_i = a = (\nu^\sigma_{x_1,\ldots,x_{i-1}})_i$$

by the previous result. For the inductive step,

$$\begin{aligned} (\nu^\sigma_{x_1,\ldots,x_{i-1},a})_j &= \sigma_j((\nu^\sigma_{x_1,\ldots,x_{i-1},a})_1,\ldots,(\nu^\sigma_{x_1,\ldots,x_{i-1},a})_{j-1}) \\ &= \sigma_j((\nu^\sigma_{x_1,\ldots,x_{i-1}})_1,\ldots,(\nu^\sigma_{x_1,\ldots,x_{i-1}})_{j-1}) \\ &= (\nu^\sigma_{x_1,\ldots,x_{i-1}})_j \end{aligned}$$

This completes the entire proof.

Chapter 2

The algebra and geometry of games

2.1 Open games

2.1.1 Discussion

In this section we define open games, our objects of study. The barrier to entry is high: this section is quite abstract, and although we focus on examples and intuitions inline with the theory, we must wait until §2.3.9 before giving even the most trivial examples from game theory textbooks. Moreover the definitions we develop here bear no apparent relation whatsoever to the usual, familiar definitions of game due to von Neumann [LBS08], or the higher order games introduced in chapter 1, and a large part of this thesis will be spent developing the reader's intuition for how these objects behave and how they should be used in mathematical modelling. To some extent the investment in abstraction will not pay off in the scope of this thesis, which is setting the groundwork for serious applications.

There are two key pieces of intuition that should be understood before beginning this section. The first is that a game should be seen as a *process*. In particular a game should be a process that maps observations to choices: a game should input whatever information is observable, compute a decision and then output that decision. We remain agnostic about what is meant by 'computation' using the familiar technique of modelling a computation as a morphism of a suitable axiomatically-defined category. In this informal games-as-processes description we can already see that a game may consist of an *aggregate* of players: the game-process may distribute its input to a collection of sub-processes, each of which has only partial access to the true input, and which run in parallel to compute choices.

The idea of viewing games as processes, and hence as morphisms of a suitable category, first appears (to the author's knowledge) in [Pav09]. Our implementation, however, is quite different. With the theory here we are able to take the logical next step and view sequential and parallel play of games as sequential and parallel composition of processes. This brings game theory into line with other areas in which 'process' can be viewed as meaning 'morphism of a monoidal category', especially in physics [Coe06].

The second key piece of intuition, which is shared with higher order game theory and which we reiterate from §1.1.1, is that all of our computation is done relative to a continuation. A game is a process that runs in a calling environment, and after a process has terminated with a decision the calling environment will use that decision to compute an outcome for the process. This computation taking decisions to outcomes is the process' continuation, and it is one of the most fundamental principles of game theory that this computation is known to the players, hence the processes run in continuation passing style. We will uniformly use the term 'continuation' to refer to what is variously called a 'utility function', 'payoff function', 'payoff matrix' or 'outcome function'. However continuations are more general, and include for example contexts in which other players are making rational choices. This ability to reduce multi-player situations to single-player situations, by abstracting away the other players into something akin to an outcome function, is a crucial ingredient of compositional game theory. The outcomes-as-continuations view strongly informed the definitions in this section, and other parts of the theory.

The entirety of chapter 2 is essentially based on [GH16].

2.1.2 The underlying model of computation

We will now introduce our model of computation: **symmetric monoidal categories**. The definitions in this section will be made with respect to an arbitrary symmetric monoidal category, which will be instantiated with particular examples in later sections. This should be seen in the context of premonoidal categories and the computational λ-calculus [Mog89], restricted to commutative effects.

We will begin with a category with finite products, which for simplicity we will take to be the category **Set** of sets and functions. Given a monad $T : \mathbf{Set} \to \mathbf{Set}$, we can form the Kleisli category $\mathrm{Kl}(T)$, whose objects are sets and whose morphisms are Kleisli arrows,

$$\hom_{\mathrm{Kl}(T)}(X, Y) = X \to TY$$

The unit morphisms of $\mathrm{Kl}(T)$ are the units of T, and the composition of morphisms in $\mathrm{Kl}(T)$ is given by Kleisli extension.

A **monoidal monad** [Koc72] is, intuitively, a monad T equipped with a product operator

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

(For a full definition see [Sea13].) If the diagram

$$\begin{tikzcd} TX \times TY \arrow[r, "\otimes"] \arrow[d, "\sigma_{TX,TY}"] & T(X \times Y) \arrow[d, "T\sigma_{X,Y}"] \\ TY \times TX \arrow[r, "\otimes"] & T(Y \times X) \end{tikzcd}$$

commutes then T is called a **commutative monad**, and this can be used to make $\mathrm{Kl}(T)$ into a symmetric monoidal category [PR93]. The unit object of $\mathrm{Kl}(T)$ is $I = 1$, the set with one element, and the object $X \otimes Y$ is the cartesian product $X \times Y$.

Categories of the form $\mathrm{Kl}(T)$ have some additional structure we will need, namely that every object X canonically has the structure of a cocommutative comonoid: that is, we have deleting morphisms

$$!_X : \hom_{\mathrm{Kl}(T)}(X, I)$$

and copying morphisms

$$\Delta_X : \hom_{\mathrm{Kl}(T)}(X, X \otimes X)$$

given by composing the unique maps in **Set** with the units of T. Notice that in general, however, the monoidal product is not a cartesian product. As a consequence, we also get canonical projections

$$\pi_1 : X \otimes Y \xrightarrow{X \otimes !_Y} X \otimes I \xrightarrow{\rho_X} X$$

and

$$\pi_2 : X \otimes Y \xrightarrow{!_X \otimes Y} I \otimes Y \xrightarrow{\lambda_Y} Y$$

The justification and intuition for considering such a category as a 'model of computation' is described in §0.2. For the remainder of this chapter, $\mathcal{C}$ is going to refer to an arbitrary category in which every object carries the structure of a cocommutative comonoid. One example is **Set**, which translates into game-theoretic terms as pure strategies. Another example is **SRel**, which gives mixed strategies and which we will now describe.

2.1.3 The category of stochastic relations

We begin with the **finitary probability distribution monad** $\mathscr{D} : \mathbf{Set} \to \mathbf{Set}$. The underlying functor acts on sets by

$$\mathscr{D}X = \left\{ \alpha : X \to [0,1] \;\middle|\; \mathrm{supp}(\alpha) \text{ is finite}, \sum_{x \in \mathrm{supp}(\alpha)} \alpha x = 1 \right\}$$

where $\mathrm{supp}(\alpha) = \{x : X \mid \alpha x \neq 0\}$. We consider values $\alpha : \mathscr{D}X$ as random variables of type X, and use the notation $\mathbb{P}[\alpha = x]$ for $\alpha x : [0,1]$. The action on morphisms $f : X \to Y$ is given by

$$\mathbb{P}[\mathscr{D} f \alpha = y] = \sum_{\substack{x \in \mathrm{supp}(\alpha) \\ fx = y}} \mathbb{P}[\alpha = x]$$

This makes $\mathscr{D}$ into a functor.

The monad unit $\delta : X \to \mathscr{D}X$ creates unit mass distributions,

$$\mathbb{P}\left[\delta x = x'\right] = \begin{cases} 1 & \text{if } x = x' \\ 0 & \text{otherwise} \end{cases}$$

The Kleisli extension of a function $f : X \to \mathscr{D}Y$ is given by

$$\mathbb{P}\left[f^* \alpha = y\right] = \sum_{x \in \mathrm{supp}(\alpha)} \mathbb{P}\left[f x = y\right]$$

This makes $\mathscr{D}$ into a monad on **Set**. Finally, we have a monoidal product $\otimes : \mathscr{D}X \times \mathscr{D}Y \to \mathscr{D}(X \times Y)$ given by

$$\mathbb{P}\left[\alpha \otimes \beta = (x, y)\right] = \mathbb{P}\left[\alpha = x\right] \cdot \mathbb{P}\left[\beta = y\right]$$

With these operations, $\mathscr{D}$ is a commutative monad on **Set**, and so its Kleisli category is symmetric monoidal. We will refer to the Kleisli category as **SRel**, the category of sets and **stochastic relations**. This is a variant of the usual notion of stochastic relations, which is defined in [Pan99] using subprobability rather than probability distributions. The distribution monad used here is the same one used in the study of convex sets [Fri09, JWW15], and is a finitary version of the Giry monad on the category of measure spaces [Gir82]. Although subprobability distributions might be interesting in this setting, allowing choices to fail with some probability, we will use true probability distributions to remain close to Nash's original assumptions.

2.1.4 Open games

Open games are defined with respect to a category $\mathcal{C}$ carrying the structure described in §2.1.2, namely a symmetric monoidal category in which every object X carries the structure of a cocommutative comonoid $(X, !_X, \Delta_X)$. More explicitly, $\mathcal{C}$ can be thought of as the kleisli category of a commutative monad on **Set**.

We can now give the definition of an open game. A **type** of an open game is of the form

$$\mathcal{G} : (X, S) \to (Y, R)$$

where X, Y, R, S are objects of $\mathcal{C}$. For now this is purely formal; later we will make open games into the morphisms of a category whose objects are pairs. In §2.3.5 we will introduce an alternative notation for this, namely

$$\mathcal{G} : X \otimes S^* \to Y \otimes R^*$$

Each of the four objects of $\mathcal{C}$ should be read in a different way. We view X as the type of **observations** that can be made by $\mathcal{G}$, and Y as the type of **moves** or **choices**, hence a game maps observations to choices. The types R and S, on the other hand, are 'dual' or 'contravariant' types. A useful intuition is that a type appearing in the right component of a pair represents a type whose elements are elements of R in the future, and which a rational player is reasoning about. We view R as the type of **outcomes** of $\mathcal{G}$, that is, the type of values about which the players in $\mathcal{G}$ have preferences. The type S, dually, represents outcomes that are 'generated' by $\mathcal{G}$ and returned to the calling environment.

We now finally arrive at the central definition of this thesis, that of an **open game**, which we will often simply call a **game**. An open game of type

$$\mathcal{G} : (X, S) \to (Y, R)$$

is, by definition, a 4-tuple

$$\mathcal{G} = (\Sigma_\mathcal{G}, \mathbf{P}_\mathcal{G}, \mathbf{C}_\mathcal{G}, \mathbf{B}_\mathcal{G})$$

where

- $\Sigma_\mathcal{G}$ is a set, called the set of **strategy profiles** of $\mathcal{G}$
- $\mathbf{P}_\mathcal{G} : \Sigma_\mathcal{G} \to \hom_\mathcal{C}(X, Y)$ is called the **play function** of $\mathcal{G}$
- $\mathbf{C}_\mathcal{G} : \Sigma_\mathcal{G} \to \hom_\mathcal{C}(X \otimes R, S)$ is called the **coplay function** of $\mathcal{G}$
- $\mathbf{B}_\mathcal{G} : \hom_\mathcal{C}(I, X) \times \hom_\mathcal{C}(Y, R) \to \Sigma_\mathcal{G} \to \mathscr{P}\Sigma_\mathcal{G}$ is the **best response function** of $\mathcal{G}$

In general we impose no conditions whatsoever on these components. In practice, however, we will restrict to games which are freely generated by the constructions considered in this section and §2.2, which gives some implicit restrictions.

Notice the unusual mixture of 'internal' and 'external' parts of this definition. It would be simple to directly generalise this to the setting in which $\mathcal{C}$ is enriched over another category $\mathcal{V}$; the definition given is the case when $\mathcal{V} = \mathbf{Set}$. If $\mathcal{C}$ is enriched over itself (as is the case when $\mathcal{C} = \mathbf{Set}$), the definition is equivalent to a purely internal one. If $\mathcal{C}$ is the kleisli category of some monad, it could also be given a purely internal definition in terms of that monad.

The most straightforward parts of this definition are the first two components. It is intuitive that a game has a set of strategy profiles and that, given a strategy profile and an observation, we can run the strategy profile on the observation to obtain a choice. To give a simple concrete example with $\mathcal{C} = \mathbf{Set}$, suppose $Y = A \times B$, and define

$$\Sigma = (X \to A) \times (X \times A \to B)$$

and

$$\mathbf{P}(\sigma_1, \sigma_2)x = (\sigma_1 x, \sigma_2(x, \sigma_1 x))$$

This represents a two-player game of perfect information: first the value x is input, then the first player observes this and chooses a, and then the second player observes both x and a and chooses b, and finally the pair (a, b) is output.

Probably the most mysterious part of the definition is the coplay function. The basic idea is that the coplay function takes a utility in the future and transforms it to a utility less far in the future (this idea will be made more explicit in the conclusion). It is completely unclear, however, why this should depend on an observation and a strategy. The only explanation that will be given is that the dependence on the observation is used only to define the counit game in §2.1.9, which transforms an observation into a utility. The dependence on a strategy profile is a consequence of this, since to define coplay for an aggregate it is necessary to have access to the play functions of the components, which requires a strategy profile.

2.1.5 The best response function

The final part of the definition of an open game, and the part which needs the most explanation, is the best response function. In classical game theory the best response function of a game is a function $\mathbf{B} : \Sigma \to \mathscr{P}\Sigma$, where $\sigma' \in \mathbf{B}\sigma$ means that each player i would be satisfied to play her component σ'_i in response to the situation in which every other player j plays the strategy σ_j. When this happens we say that σ' is a 'best response' to σ. An equilibrium can be defined as a strategy which is a best response to itself, that is to say, a fixpoint of the multi-valued function $\mathbf{B}$. The best response functions in §1.2.4 and §1.3.5 behave in this way.

We can quite easily replace $\mathbf{B}_\mathcal{G}$ for open games with an 'equilibrium set function'

$$\mathbf{E}_\mathcal{G} : \hom_\mathcal{C}(I, X) \times \hom_\mathcal{C}(Y, R) \to \mathscr{P}\Sigma_\mathcal{G}$$

by defining $\mathbf{E}_\mathcal{G}(h, k)$ to be the set of fixpoints of $\mathbf{B}_\mathcal{G}(h, k)$. If we do this, the remaining definitions in this section can be made correctly, without reference to best responses. However we choose to always carry around the best response function, which after all contains more information, for the reasons given in §1.2.4: it provides a method to prove existence theorems , and it gives a finer notion of equivalence between games. If we were studying 'abstract game theory' more seriously we could investigate the idea that the string diagrams in §2.3.5 can be interpreted in two different categories, and the best response function is forgotten by an identity-on-objects functor.

One of the fundamental ideas of this thesis (see §1.1.1 and §2.1.1) is that games should only be defined relative to a continuation. This leads to the idea of allowing a continuation as an additional parameter to the best response function, namely

$$\mathbf{B}_\mathcal{G} : \hom_\mathcal{C}(Y, R) \to \Sigma_\mathcal{G} \to \mathscr{P}\Sigma_\mathcal{G}$$

Here we see in full generality what is meant by the term 'continuation' in this thesis: it is nothing but a function from choices to outcomes. The importance of working relative to a truly *arbitrary* continuation, rather than an arbitrary-but-fixed continuation such as an outcome function, is that when we define $\mathbf{B}_\mathcal{G} k\sigma$ in terms of the best response functions of its components, we will use continuations k' that actually depend on the strategy profile σ as well as on k. Thus, allowing continuations to vary is at least as important as allowing strategies to vary.

If we define the best response function in this way we again obtain a logically sound theory. However this definition contains a serious error, because nontrivial examples can only be written if the best response function is also allowed to depend on the observation made. A pair

$$(h, k) : \hom_\mathcal{C}(I, X) \times \hom_\mathcal{C}(Y, R)$$

will be called a **context** for the game $\mathcal{G}$, and we can roughly think of h and k as the past and future behaviour of the calling environment. There is another major subtlety here, namely that the best response function does not depend on a pure observation but on the *computation of an observation*, which may have side effects. If T is a commutative monad on the category of sets, and $\mathcal{C}$ is its (symmetric monoidal) Kleisli category, then the observation is a value of type TX rather than X. For example if we have probabilistic choice as a side effect by taking $\mathcal{C} = \mathbf{SRel}$ (see §2.1.3) then we need to consider probability distributions over possible observations, because elements of $\hom_\mathbf{SRel}(I, X)$ are random variables of type X. If we try to use pure observations instead, we find that we cannot define categorical composition in §2.2.3.

Finally, we come to a family of variants that are much more important. The definition of best response as a multivalued function and an equilibrium as a fixpoint is classical, and is sufficient for working with pure, mixed and nondeterministic strategies. This idea, however, does not stand up to more complex side effects such as learning. If we write the multivalued function instead as

$$\mathbf{B}_\mathcal{G} : \hom_\mathcal{C}(I, X) \times \hom_\mathcal{C}(Y, R) \to \Sigma_\mathcal{G} \to \Sigma_\mathcal{G} \to \mathbb{B}$$

then the better definition is to replace the booleans $\mathbb{B}$ with some other (in general noncommutative) algebraic structure B. If we only care about Nash equilibria then we can take B to be a monoid, however for reasoning about subgame perfection it seems that B should be some kind of 'noncommutative complete semilattice'. A typical example would be of the form $B = T'\mathbb{B}$, where T' is another (strong, not necessarily commutative) monad on the category of sets, which can be different to T, the monad with $\mathcal{C} = \mathrm{Kl}(T)$. Since it is still unclear what the appropriate definition should be we will use the simpler one by default.

2.1.6 Closed games

The reason for the terminology *open game* introduced in §2.1.4 is that an open game is *open* to its environment, in the sense that information can pass back and forth. However, if one of X, Y, R, S in $\mathcal{G} : (X, S) \to (Y, R)$ is equal to the tensor unit I in $\mathcal{C}$, then the flow of information is restricted, and if all are equal to I then no information can pass between the game and its environment. A **closed game** is therefore defined[1] to be an open game of type $\mathcal{G} : (I, I) \to (I, I)$.

Looking ahead to the result in §2.2.10, that open games are morphisms of a monoidal category whose objects are pairs of sets, and whose tensor unit is (I, I), a closed game can equivalently be called an **abstract scalar**. This terminology is from [AC04, Abr05], and we will use it interchangeably with 'closed game'. See also [KL80].

Closed games are particularly interesting when I is terminal in $\mathcal{C}$. The categories **Set** and **SRel** both have this property, although **Rel** (which models nondeterminism) does not. In this case, closed games have a very simple formulation: the play and coplay functions become trivial, and the game is described entirely by its set $\Sigma_\mathcal{G}$ of strategy profiles, and its best response function, whose type reduces to

$$\mathbf{B}_\mathcal{G} : \Sigma_\mathcal{G} \to \mathscr{P}\Sigma_\mathcal{G}$$

A game in the intuitive, informal sense can be described by this data (see §1.2.4), and correspondingly there is no notion of information being shared with the environment in classical game theory, which suggests that games in the usual sense should be thought of as closed games.

A closed game $\mathcal{G}$ over any category has a canonical history and a canonical continuation, namely both being given by id_I. Thus we can give a solution concept for closed games: an **equilibrium** of $\mathcal{G}$ is a strategy profile $\sigma : \Sigma_\mathcal{G}$ with $\sigma \in \mathbf{B}_\mathcal{G}(\mathrm{id}_I, \mathrm{id}_I)\sigma$. This solution concept will be used in §3.1 and §3.2, where we see that it includes pure and mixed Nash and subgame perfect equilibria as special cases.

[1] Games are not doors, or topologies: every closed game is open, although it is reasonable to think of closed games as a degenerate case of open games.

2.1.7 Decisions

In this section we see our first examples of open games, the **decisions**, which are one-player games. Every nontrivial game contains decisions as components, representing the players of the game.

Decisions, by definition, are games of the form

$$\mathcal{D} : (X, I) \to (Y, R)$$

where the player makes an observation from X and makes a choice from Y, with preferences over outcomes in R. Formally, the requirements for a game with this type to be a decision are

- $\Sigma_{\mathcal{D}} = \hom_{\mathcal{C}}(X, Y)$
- $\mathbf{P}_{\mathcal{D}} \sigma = \sigma : \hom_{\mathcal{C}}(X, Y)$
- $\mathbf{C}_{\mathcal{D}} \sigma = !_{X \otimes R} : \hom_{\mathcal{C}}(X \otimes R, I)$
- $\mathbf{B}_{\mathcal{D}}(h, k) : \hom_{\mathcal{C}}(X, Y) \to \mathscr{P} \hom_{\mathcal{C}}(X, Y)$ is constant

Consequently, to define a decision is to define a function

$$\mathbf{R} : \hom_{\mathcal{C}}(I, X) \times \hom_{\mathcal{C}}(Y, R) \to \mathscr{P} \hom_{\mathcal{C}}(X, Y)$$

Such a function will be called a **rationality function**, and defines the best response function of a decision by

$$\mathbf{B}_{\mathcal{D}}(h, k)\sigma = \mathbf{R}(h, k)$$

Since a decision always consists of a single player and never an aggregate, a strategy is simply a mapping from observations to outcomes. The fact that the strategy is a morphism of $\mathcal{C}$ means that the strategy can have effects, such as probabilistic choice (see §1.2.6). The condition on $\mathbf{P}_{\mathcal{D}}$ is simply that playing the strategy involves applying it as a function to the observation.

The only possibly unexpected condition is that on $\mathbf{B}_{\mathcal{D}}$. The idea of best responses (see §1.2.4) is that $\mathbf{B}(h, k)\sigma$ should be the set of strategy profiles σ' such that, for each player, playing using σ' is a rational response to the situation in which every other player is playing using σ. Since a decision involves only one player any property of 'every other player' is vacuous, and so σ is not used in the definition.

The definition of a decision is already at the right level of generality that an arbitrary multi-valued selection function (§1.1.6) can be considered as a decision. First, notice that a decision $\mathscr{D} : (I, I) \to (Y, R)$ over $\mathcal{C} = \mathbf{Set}$ is defined by a rationality function

$$\mathbf{R} : (Y \to R) \to \mathscr{P} Y$$

which is precisely the type of a multi-valued selection function (see §1.1.6). More generally, a multi-valued selection function $E : (Y \to R) \to \mathscr{P} Y$ can be converted into a decision $\mathcal{D}_E : (X, I) \to (Y, R)$ for an arbitrary set X, using the response function

$$\mathbf{R}(x, k) = \{\sigma : X \to Y \mid \sigma x \in Ek\}$$

The resulting decision models a player whose rationality is defined by E.

2.1.8 Preliminary examples of decisions

We will give two example families of decisions representing classically rational players, respectively with pure and mixed strategies. In the first case, we take $\mathcal{C} = \mathbf{Set}$. We will fix a rational preference relation $\preceq$ on a set R, representing a player's preferences for different outcomes in R, where $r \succeq r'$ means that the outcome r is considered at least as good as the outcome r'. We will also fix an arbitrary set X and a finite set Y. To define a decision

$$\mathcal{D} : (X, 1) \to (Y, R)$$

is to define a rationality function

$$\mathbf{R} : X \times (Y \to R) \to \mathscr{P}(X \to Y)$$

We define a particular family of decisions by the response functions

$$\mathbf{R}(x, k) = \{\sigma : X \to Y \mid k(\sigma x) \succeq ky \text{ for all } y : Y\}$$

We will now explicitly give the data defining $\mathcal{D}$. Its strategies are $\Sigma_{\mathcal{D}} = X \to Y$, and its play function $\mathbf{P}_{\mathcal{D}} : (X \to Y) \to (X \to Y)$ is $\mathbf{P}_{\mathcal{D}}\sigma x = \sigma x$. Its coplay function $\mathbf{C}_{\mathcal{D}} : (X \to Y) \to (X \times R \to 1)$ is $\mathbf{C}_{\mathcal{D}}(\sigma, (x, r)) = *$. Its best response function

$$\mathbf{B}_{\mathcal{D}} : X \times (Y \to R) \to (X \to Y) \to \mathscr{P}(X \to Y)$$

is given by

$$\mathbf{B}_{\mathcal{D}}(x, k)\sigma = \{\sigma' : X \to Y \mid k(\sigma' x) \succeq ky \text{ for all } y : Y\}$$

For our second family of examples we will use $\mathcal{C} = \mathbf{SRel}$ (defined in §2.1.3). For an arbitrary set X and finite set Y we define a decision

$$\mathcal{D} : (X, 1) \to (Y, \mathbb{R})$$

modelling a player who maximises expected utility (see §1.2.6). To specify such a decision is to specify a response function

$$\mathbf{R} : \mathscr{D}X \times (Y \to \mathscr{D}\mathbb{R}) \to \mathscr{P}(X \to \mathscr{D}Y)$$

where $\mathscr{D}$ is the distribution monad (see §2.1.3). We use the particular function

$$\mathbf{R}(h, k) = \{\sigma : X \to \mathscr{D}Y \mid \mathbb{E}[k^*(\sigma^* h)] \geq \mathbb{E}[ky] \text{ for all } y : Y\}$$

where $\mathbb{E} : \mathscr{D}\mathbb{R} \to \mathbb{R}$ is the expectation operator defined by

$$\mathbb{E}[\alpha] = \sum_{x \in \mathrm{supp}(\alpha)} \mathbb{P}[\alpha = x] \cdot x$$

and $-^*$ is the Kleisli extension of $\mathscr{D}$. Explicitly, we have the strategy set $\Sigma_{\mathcal{D}} = X \to \mathscr{D}Y$. The play function

$$\mathbf{P}_{\mathcal{D}} : (X \to \mathscr{D}Y) \to (X \to \mathscr{D}Y)$$

is given by $\mathbb{P}[\mathbf{P}_{\mathcal{D}}\sigma x = y] = \mathbb{P}[\sigma x = y]$. The coplay function

$$\mathbf{C}_{\mathcal{D}} : (X \to \mathscr{D}Y) \to (X \times \mathbb{R} \to \mathscr{D}1)$$

is given by $\mathbb{P}[\mathbf{C}_{\mathcal{D}}\sigma(x, r) = *] = 1$. The best response function

$$\mathbf{B}_{\mathcal{D}} : \mathscr{D}X \times (Y \to \mathscr{D}\mathbb{R}) \to (X \to \mathscr{D}Y) \to \mathscr{P}(X \to \mathscr{D}Y)$$

is given by

$$\mathbf{B}_{\mathcal{D}}(h, k)\sigma = \{\sigma' : X \to \mathscr{D}Y \mid \mathbb{E}[k^*(\sigma'^* h)] \geq \mathbb{E}[ky] \text{ for all } y : Y\}$$

2.1.9 Computations and counit

Besides decisions, the other atomic games we will consider are computations and counits. These components are unable to make strategic decisions, and in particular only have one strategy, which behaves trivially. Their purpose, however, is to control the information flow in a game. Examples of computations include outcome functions, identities, copying and deleting. The counits serve to connect forward-flowing to backward-flowing data, that is, identifying a particular forward-flowing value as the value that some player is reasoning about.

Formally, a **computation** is a game

$$(f, g) : (X, S) \to (Y, R)$$

defined by morphisms $f : \hom_{\mathcal{C}}(X, Y)$ and $g : \hom_{\mathcal{C}}(R, S)$. The definition is given by

- $\Sigma_{(f,g)} = 1 = \{*\}$
- $\mathbf{P}_{(f,g)} * = f : \hom_{\mathcal{C}}(X, Y)$
- $\mathbf{C}_{(f,g)} * : \hom_{\mathcal{C}}(X \otimes R, S)$ is given by the composition

$$X \otimes R \xrightarrow{\pi_2} R \xrightarrow{g} S$$

- $\mathbf{B}_{(f,g)}(h, k)* = \{*\}$ for all h and k

The conditions $\Sigma_{(f,g)} = 1$ and $\mathbf{B}_{(f,g)}(h, k)* = \{*\}$ implement the idea that (f, g) is 'strategically trivial'. The fact that $* \in \mathbf{B}_{(f,g)}(h, k)*$ means that $*$ is always an equilibrium of a computation. The idea behind this is that a strategy for an aggregate game should never *fail* to be an equilibrium because of a computation, because a computation has no preferences. Rather, if a strategy fails to be an equilibrium, it should always be because some *player* has incentive to deviate. We expand on this idea in §2.2.11.

The conditions on $\mathbf{P}_{(f,g)}$ and $\mathbf{C}_{(f,g)}$ determine the information flow, which is explained in more detail in §2.2.14. When $\mathcal{C} = \mathbf{Set}$, the coplay function of a computation is simply

$$\mathbf{C}_{(f,g)} * (x, r) = gr$$

Next, for each object X of $\mathcal{C}$ we define a **counit** $\varepsilon_X : (X, X) \to (I, I)$ by

- $\Sigma_{\varepsilon_X} = 1 = \{*\}$
- $\mathbf{P}_{\varepsilon_X} * = !_X : \hom_{\mathcal{C}}(X, I)$
- $\mathbf{C}_{\varepsilon_X} * = \rho_X : \hom_{\mathcal{C}}(X \otimes I, X)$
- $\mathbf{B}_{\varepsilon_X}(h, k)* = \{*\}$ for all h and k

The intuition behind this definition is similar to that for computations. If $\mathcal{C} = \mathbf{Set}$, the coplay function is $\mathbf{C}_{\varepsilon_X} * (x, *) = x$.

2.2 The category of games

2.2.1 Discussion

In §2.1 we introduced many definitions, especially of open games (§2.1.4), decisions (§2.1.7), computations and counits (§2.1.9). In this section we will add algebraic structure, giving two ways to compose games: categorical composition, which is a primitive form of sequential play of games, and tensor product, which is a primitive form of simultaneous play.

As a matter of fact, in this section we define *two different* categorical composition operators. This is essentially a historical accident, and the state of affairs is discussed in the conclusion. These two categorical composition operators are called N-composition and SP-composition, and they correspond roughly to a choice of solution concept between Nash equilibrium and subgame perfect equilibrium.

In this section we will additionally prove that N-composition and tensor product together obey the axioms of a symmetric monoidal category. In §2.3 this algebraic structure will be investigated further. The appendix, which closely follows this section (minus the explanatory text) does the same for SP-composition, proving that it forms a symmetric *pre*monoidal category with the same definition of tensor product. Both operators will be used in chapter 3.

Readers who are not category theorists can treat composition and tensor product as rather like sequential and parallel composition operators in process algebra, similar to those in [BW13], with the axioms we check simply being technical conditions for the string diagram language introduced in §2.3.5 to be sound.

One important direction that we will not consider in this thesis (although it is discussed in the appendix) is to additionally consider morphisms between games, so that we have a monoidal bicategory [SP09]. This would allow us to give some canonicity to the operators defined in this section (for example, characterising them as universal properties), and more importantly, when considering infinite games such as repeated games, a natural approach is to define the game as a terminal coalgebra of a functor which precomposes a finite approximation to the repeated game with one additional stage. Because a terminal coalgebra is defined by a universal property of objects, this requires morphisms between games. There are several possible definitions that could be used, however, and at the present time it is not clear which are useful. One possibility that would be elegant, but is only speculation, is that solutions of games (see §3.3.2) are global points, and that backward induction of solutions (§3.3.4) is an instance of horizontal composition of 2-cells. A bicategory of games would also provide a formal theory of equivalences and refinements between games, as discussed in §1.2.4.

2.2.2 Equivalences of games

For technical reasons, we must quotient the class of open games by isomorphisms of strategy sets. This is necessary to obtain a category because, for example, we would otherwise find that the compositions $\mathcal{I} \circ (\mathcal{H} \circ \mathcal{G})$ and $(\mathcal{I} \circ \mathcal{H}) \circ \mathcal{G}$ are not equal (as required by the axioms of a category) because their underlying sets of strategies are not *equal*, merely naturally isomorphic. By taking morphisms of the category to be suitable equivalence classes, this problem is avoided.

Given two open games $\mathcal{G}, \mathcal{G}' : (X, S) \to (Y, R)$, we will say that $\mathcal{G}$ and $\mathcal{G}'$ are **equivalent**, and write $\mathcal{G} \sim \mathcal{G}'$, if there exists an isomorphism $i : \Sigma_\mathcal{G} \cong \Sigma_{\mathcal{G}'}$ which commutes with the play, coplay and best response functions. That is, we demand that

- $\mathbf{P}_\mathcal{G}\sigma = \mathbf{P}_{\mathcal{G}'}(i\sigma) : \hom_\mathcal{C}(X, Y)$
- $\mathbf{C}_\mathcal{G}\sigma = \mathbf{C}_{\mathcal{G}'}(i\sigma) : \hom_\mathcal{C}(X \otimes R, S)$
- $\sigma' \in \mathbf{B}_\mathcal{G}(h, k)\sigma \iff i\sigma' \in \mathbf{B}_{\mathcal{G}'}(h, k)(i\sigma)$

always hold.

In the remainder of this section, we must check that everything we define is independent of the choice of representative of the equivalence class.

2.2.3 Categorical composition of games

Given a pair of games $\mathcal{G} : (X, T) \to (Y, S)$ and $\mathcal{H} : (Y, S) \to (Z, R)$, we need to define a composition $\mathcal{H} \circ_N \mathcal{G} : (X, T) \to (Z, R)$. This is a primitive form of composition in which the intermediate choice at Y is hidden; the more intuitive sequential composition that produces plays of type $X \otimes Y$ will be recovered using tensor product and identities in §3.2.2. The game $\mathcal{H} \circ_N \mathcal{G}$ is an aggregate whose players consist of the players of $\mathcal{G}$ together with the players of $\mathcal{H}$. This means that a strategy profile for $\mathcal{H} \circ_N \mathcal{G}$ should consist of a strategy profile for $\mathcal{G}$ together with a strategy profile for $\mathcal{H}$, that is, $\Sigma_{\mathcal{H} \circ_N \mathcal{G}} = \Sigma_\mathcal{G} \times \Sigma_\mathcal{H}$.

Since the play is sequential, the observation that is made by $\mathcal{H}$ should be the choice that is made by $\mathcal{G}$. This motivates the definition of the play function of $\mathcal{H} \circ_N \mathcal{G}$, namely that $\mathbf{P}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau)$ should be the composition

$$X \xrightarrow{\mathbf{P}_\mathcal{G}\sigma} Y \xrightarrow{\mathbf{P}_\mathcal{H}\tau} Z$$

Using the categorical composition in $\mathcal{C}$ appropriately sequences the effects that can be used by the two components. For example if $\mathcal{G}$ and $\mathcal{H}$ both contain players who can make probabilistic choices, the play function of $\mathcal{H} \circ_N \mathcal{G}$ gives the appropriate probability distribution on Z taking into account the probability distribution on Y.

The coplay function of $\mathcal{H} \circ_N \mathcal{G}$, on the other hand, is hard or impossible to justify on intuitive grounds. We define $\mathbf{C}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau)$ to be the composition

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_\mathcal{G}\sigma \otimes R} X \otimes Y \otimes R \xrightarrow{X \otimes \mathbf{C}_\mathcal{H}\tau} X \otimes S \xrightarrow{\mathbf{C}_\mathcal{G}\sigma} T$$

(where we are assuming that $\mathcal{C}$ is strict monoidal). When $\mathcal{C} = \mathbf{Set}$, this is given explicitly by

$$\mathbf{C}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau)(x, r) = \mathbf{C}_\mathcal{G}\sigma(x, \mathbf{C}_\mathcal{H}\tau(\mathbf{P}_\mathcal{G}\sigma x, r))$$

This will be discussed in §2.2.14.

For now we will also give the best response function without motivation, but it will be discussed in detail in the next section. The definition is

$$\mathbf{B}_{\mathcal{H} \circ_N \mathcal{G}}(h, k)(\sigma, \tau) = \mathbf{B}_\mathcal{G}(h, k_{\tau\sigma}) \times \mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma \circ h, k)\tau$$

where $k_{\tau\circ}$ is the composition

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes \mathbf{P}_{\mathcal{H}}\tau} Y \otimes Z \xrightarrow{Y \otimes k} Y \otimes R \xrightarrow{\mathbf{C}_{\mathcal{H}}\tau} S$$

With this definition we obtain a category $\mathbf{Game}_N(\mathcal{C})$, whose objects are pairs of objects of $\mathcal{C}$, and whose morphisms are equivalence classes of open games, with the identity on (X, R) being the computation $(\mathrm{id}_X, \mathrm{id}_R)$. The axioms of a category will be proved in §2.2.5 and §2.2.6.

We must now prove that $\mathcal{H} \circ_N \mathcal{G}$ is well-defined, i.e. if $\mathcal{G} \sim \mathcal{G}'$ and $\mathcal{H} \sim \mathcal{H}'$ then $\mathcal{H} \circ_N \mathcal{G} \sim \mathcal{H}' \circ_N \mathcal{G}'$. We are given isomorphisms $i_\mathcal{G} : \Sigma_\mathcal{G} \to \Sigma_{\mathcal{G}'}$ and $i_\mathcal{H} : \Sigma_\mathcal{H} \to \Sigma_{\mathcal{H}'}$, and we define a new isomorphism $i_{\mathcal{H} \circ_N \mathcal{G}}(\sigma_1, \sigma_2) = (i_\mathcal{G}\sigma_1, i_\mathcal{H}\sigma_2)$. For the play function,

$$\begin{aligned} \mathbf{P}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau) &= \mathbf{P}_\mathcal{H}\tau \circ \mathbf{P}_\mathcal{G}\sigma \\ &= \mathbf{P}_{\mathcal{H}'}(i_\mathcal{H}\tau) \circ \mathbf{P}_{\mathcal{G}'}(i_\mathcal{G}\sigma) \\ &= \mathbf{P}_{\mathcal{H}' \circ_N \mathcal{G}'}(i_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau)) \end{aligned}$$

Similarly for coplay,

$$\begin{aligned} \mathbf{C}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau) &= \mathbf{C}_\mathcal{G}\sigma \circ (X \otimes \mathbf{C}_\mathcal{H}\tau) \circ (X \otimes \mathbf{P}_\mathcal{G}\sigma \otimes R) \circ (\Delta_X \otimes R) \\ &= \mathbf{C}_{\mathcal{G}'}(i_\mathcal{G}\sigma) \circ (X \otimes \mathbf{C}_{\mathcal{H}'}(i_\mathcal{H}\tau)) \circ (X \otimes \mathbf{P}_{\mathcal{G}'}(i_\mathcal{G}\sigma) \otimes R) \circ (\Delta_X \otimes R) \\ &= \mathbf{C}_{\mathcal{H}' \circ_N \mathcal{G}'}(i_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau)) \end{aligned}$$

For the best response functions, we first check that

$$\begin{aligned} k_{\tau\circ} &= \mathbf{C}_\mathcal{H}\tau \circ (Y \otimes k) \circ (Y \otimes \mathbf{P}_\mathcal{H}\tau) \circ \Delta_Y \\ &= \mathbf{C}_{\mathcal{H}'}(i_\mathcal{H}\tau) \circ (Y \otimes k) \circ (Y \otimes \mathbf{P}_{\mathcal{H}'}(i_\mathcal{H}\tau)) \circ \Delta_Y \\ &= k_{i_\mathcal{H}\tau\circ} \end{aligned}$$

Then

$$\mathbf{B}_{\mathcal{H} \circ_N \mathcal{G}}(h, k)(\sigma, \tau) = \mathbf{B}_\mathcal{G}(h, k_{\tau\circ})\sigma \times \mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma \circ h, k)\tau$$

and

$$\mathbf{B}_{\mathcal{H}' \circ_N \mathcal{G}'}(h, k)(i_\mathcal{G}\sigma, i_\mathcal{H}\tau) = \mathbf{B}_{\mathcal{G}'}(h, k_{i_\mathcal{H}\tau\circ})(i_\mathcal{G}\sigma) \times \mathbf{B}_{\mathcal{H}'}(\mathbf{P}_{\mathcal{G}'}(i_\mathcal{G}\sigma) \circ h, k)(i_\mathcal{H}\tau)$$

Because $k_{\tau\circ} = k_{i_\mathcal{H}\tau\circ}$ we immediately get $\mathbf{B}_\mathcal{G}(h, k_{\tau\circ})\sigma = \mathbf{B}_{\mathcal{G}'}(h, k_{i_\mathcal{H}\tau\circ})(i_\mathcal{G}\sigma)$. Similarly, since $\mathbf{P}_\mathcal{G}\sigma \circ h = \mathbf{P}_{\mathcal{G}'}(i_\mathcal{G}\sigma) \circ h$, we also have

$$\mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma \circ h, k)\tau = \mathbf{B}_{\mathcal{H}'}(\mathbf{P}_{\mathcal{G}'}(i_\mathcal{G}\sigma) \circ h, k)(i_\mathcal{H}\tau)$$

and we are done.

2.2.4 Best response for sequential compositions

In order to give a definition of $\mathbf{B}_{\mathcal{H} \circ_N \mathcal{G}}(h, k)(\sigma, \tau)$, we need to define what it means for a strategy profile (σ', τ') to be a best response to (σ, τ), in the context consisting of a history $h : \hom_\mathcal{C}(I, X)$ and a continuation $k : \hom_\mathcal{C}(Z, R)$.

The idea is that we have a fixed situation in which the strategy profile played is (σ, τ), and then one player unilaterally deviates to a component of σ' or τ'.

This means that if we have a player in $\mathcal{G}$ deviating to a component of σ' we know that every player in $\mathcal{H}$ is playing τ, and vice versa.

The first condition is that σ' should be a best response to σ, in a modified context in which we extend the continuation k backwards in time using the fact that we know that $\mathcal{H}$ will be played using τ. Therefore we define the extended continuation $k_{\tau\circ}$ to be the composition

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes \mathbf{P}_{\mathcal{H}}\tau} Y \otimes Z \xrightarrow{Y \otimes k} Y \otimes R \xrightarrow{\mathbf{C}_{\mathcal{H}}\tau} S$$

For the second condition we need that τ' is a best response to τ in an appropriate context. The same continuation k can be used without modification, and the history can be extended forwards using the fact that $\mathcal{G}$ will be played with σ, namely as the composition

$$I \xrightarrow{h} X \xrightarrow{\mathbf{P}_{\mathcal{G}}\sigma} Y$$

This leads to the definition

$$\mathbf{B}_{\mathcal{H}\circ_N\mathcal{G}}(h,k)(\sigma,\tau) = \left\{ (\sigma',\tau') : \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}} \;\middle|\; \begin{array}{l} \sigma' \in \mathbf{B}_{\mathcal{G}}(h,k_{\tau\circ})\sigma \text{ and} \\ \tau' \in \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma \circ h, k)\tau \end{array} \right\}$$

which can be written equivalently as

$$\mathbf{B}_{\mathcal{H}\circ_N\mathcal{G}}(h,k)(\sigma,\tau) = \mathbf{B}_{\mathcal{G}}(h,k_{\tau\circ})\sigma \times \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma \circ h, k)\tau$$

This is precisely the definition of N-composition.

Assuming that when a player in $\mathcal{H}$ is deviating to τ', the players in $\mathcal{G}$ use σ, leads to a theory of games supporting a solution concept based on Nash equilibrium. However our games are dynamic (that is to say, they have non-trivial temporal structure in which players can observe events that happened in the past), and it is widely recognised in game theory that Nash equilibrium is not a suitable solution concept for dynamic games. In particular, in a Nash equilibrium earlier players can make so-called 'non-credible threats' to later players. The category $\mathbf{Game}_N(\mathcal{C})$ behaves in this way, in that it allows strategies making non-credible threats to be equilibria (that is, fixpoints of the best response function).

The usual solution to this problem is to use subgame-perfect equilibria (see §1.3.5). This is an equilibrium refinement of Nash: every subgame-perfect equilibrium is a Nash equilibrium, but not vice versa. In classical game theory, the usual method is to define 'subgames' as subtrees of the game tree, and define a subgame-perfect equilibrium to be a strategy which induces a Nash equilibrium when restricted to any subgame.

One possibility, therefore, would be to extend the data specifying a game with a recursively-built collection of subgames, and quantify over this collection when defining the best-response function. However, a simpler solution is possible. An apparent alternative would be to quantify over the type of histories, and say that a best response should be valid for every possible history, not just the ones that arise from the strategy profile. For classical game theory this works, but for us it is too strong. Consider, for example, a game defined (over the category of sets, for simplicity) as a composition of the form

$$(X, I) \xrightarrow{(\Delta_X, \mathrm{id}_I)} (X \otimes X, I) \xrightarrow{\mathcal{D}} (Y, R)$$

where $\mathcal{D}$ is a decision. Now the earlier component does not contain a player and cannot make strategic decisions, so our intuition about threats is no longer valid. No matter what the starting history is, and no matter what this game is precomposed with, the player making the decisions can only ever observe histories of the form (x, x). It is therefore too strong to require that the player's strategy should be rational for histories not of this form.

This example, however, suggests the solution: the image of the play function, as the strategy profile of the first component varies, gives precisely those histories for the second component that can possibly arise. Now we can return to the original problem: defining what it means for (σ', τ') to be a best response to (σ, τ) in the context (h, k). The second condition becomes that τ' should be a best response to τ now in a variety of contexts: all those of the form (h', k), where h' is of the form $\mathbf{P}_\mathcal{G}\sigma'' \circ h$, where σ'' is an arbitrary strategy profile for $\mathcal{G}$. Putting this together and rewriting it, we get

$$\mathbf{B}_{\mathcal{H}\circ_{SP}\mathcal{G}}(h, k)(\sigma, \tau) = \mathbf{B}_\mathcal{G}(h, k_{\tau\circ})\sigma \times \bigcap_{\sigma' : \Sigma_\mathcal{G}} \mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma' \circ h, k)\tau$$

We will define the SP-composition operator $\circ_{SP}$ to be identical to $\circ_N$ for strategy profiles, play and coplay, but to behave this way instead for best responses. We define another category, called $\mathbf{Game}_{SP}(\mathcal{C})$, whose composition operator is $\circ_{SP}$ but in every other way is defined identically to $\mathbf{Game}_N(\mathcal{C})$. Proofs about $\mathbf{Game}_{SP}(\mathcal{C})$ are given in the appendix.

The reason that we retain both definitions, rather than considering only SP-composition, is that N-composition is better behaved with respect to the tensor operation that we will define later in this section.

We must still prove that this definition respects equivalence. As before, we define $i_{\mathcal{H}\circ_{SP}\mathcal{G}}(\sigma, \tau) = (i_\mathcal{G}\sigma, i_\mathcal{H}\tau)$. We have

$$\mathbf{B}_{\mathcal{H}\circ_{SP}\mathcal{G}}(h, k)(\sigma, \tau) = \mathbf{B}_\mathcal{G}(h, k_{\tau\circ})\sigma \times \bigcap_{\sigma' : \Sigma_\mathcal{G}} \mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma' \circ h, k)\tau$$

and

$$\mathbf{B}_{\mathcal{H}'\circ_{SP}\mathcal{G}'}(h, k)(i_\mathcal{G}\sigma, i_\mathcal{H}\tau) = \mathbf{B}_{\mathcal{G}'}(h, k_{i_\mathcal{H}\tau\circ})(i_\mathcal{G}\sigma) \times \bigcap_{\sigma' : \Sigma_{\mathcal{G}'}} \mathbf{B}_{\mathcal{H}'}(\mathbf{P}_{\mathcal{G}'}\sigma' \circ h, k)(i_\mathcal{H}\tau)$$

The first terms are equal as for N-composition. For the second part, take some

$$\tau' \in \bigcap_{\sigma' : \Sigma_\mathcal{G}} \mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma'' \circ h, k)\tau$$

Then for each $\sigma' : \Sigma_{\mathcal{G}'}$ we have

$$\tau' \in \mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}(i_\mathcal{G}^{-1}\sigma'') \circ h, k)\tau$$

and hence

$$i_\mathcal{H}\tau' \in \mathbf{B}_{\mathcal{H}'}(\mathbf{P}_{\mathcal{G}'}\sigma'' \circ h, k)(i_\mathcal{H}\tau)$$

The converse is symmetrical.

2.2.5 The identity laws

We will now prove that $\mathbf{Game}_N(\mathcal{C})$, as defined in §2.2.3, is a category, beginning with the identity laws. Let $\mathcal{G} : (X, S) \to (Y, R)$ be a game. We first prove that $(\mathrm{id}_Y, \mathrm{id}_R) \circ_N \mathcal{G} \sim \mathcal{G}$, and hence they are equal after quotienting.

For the strategy sets,

$$\Sigma_{(\mathrm{id}_Y\,,\,\mathrm{id}_R) \circ_N \mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{(\mathrm{id}_Y\,,\,\mathrm{id}_R)} = \Sigma_{\mathcal{G}} \times 1$$

and so we take the isomorphism $i : \Sigma_{\mathcal{G}} \times 1 \to \Sigma_{\mathcal{G}}$. For the play function,

$$\mathbf{P}_{(\mathrm{id}_Y\,,\,\mathrm{id}_R) \circ_N \mathcal{G}}(i\sigma) = \mathbf{P}_{(\mathrm{id}_Y\,,\,\mathrm{id}_R)} * \circ \mathbf{P}_{\mathcal{G}}\sigma = \mathrm{id}_Y \circ \mathbf{P}_{\mathcal{G}}\sigma = \mathbf{P}_{\mathcal{G}}\sigma$$

For coplay, by definition $\mathbf{C}_{(\mathrm{id}_Y\,,\,\mathrm{id}_R) \circ_N \mathcal{G}}(i\sigma)$ is the composition

$$\begin{aligned} X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{\mathcal{G}}\sigma \otimes R} X \otimes Y \otimes R \\ \xrightarrow{X \otimes \mathbf{C}_{(\mathrm{id}_Y\,,\,\mathrm{id}_R)} *} X \otimes R \xrightarrow{\mathbf{C}_{\mathcal{G}}\sigma} S \end{aligned}$$

which, expanding the definitions further, is

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{\mathcal{G}}\sigma \otimes R} X \otimes Y \otimes R \xrightarrow{X \otimes \pi_2} X \otimes R \xrightarrow{\mathbf{C}_{\mathcal{G}}\sigma} S$$

The first part of this is the identity on $X \otimes R$, so it is equal to $\mathbf{C}_{\mathcal{G}}\sigma$.

For best response,

$$\begin{aligned} & i\sigma' \in \mathbf{B}_{(\mathrm{id}_Y,\mathrm{id}_R) \circ_N \mathcal{G}}(h, k)(i\sigma) \\ \iff & (\sigma', *) \in \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \times \mathbf{B}_{(\mathrm{id}_Y,\mathrm{id}_R)}(\mathbf{P}_{\mathcal{G}}(i\sigma) \circ h, k)* \\ \iff & \sigma' \in \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \end{aligned}$$

The continuation $k_{*\circ}$ is given by

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes \mathbf{P}_{(\mathrm{id}_Y\,,\,\mathrm{id}_R)} *} Y \otimes Y \xrightarrow{Y \otimes k} Y \otimes R \xrightarrow{\mathbf{C}_{(\mathrm{id}_Y\,,\,\mathrm{id}_R)} *} R$$

Expanding the definitions and simplifying, this is

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes k} Y \otimes R \xrightarrow{\pi_2} R$$

which is equal to k, and we are done.

For the other identity law, we will prove that $\mathcal{G} \circ_N (\mathrm{id}_X, \mathrm{id}_S) \sim \mathcal{G}$. For the strategy sets,

$$\Sigma_{\mathcal{G} \circ_N (\mathrm{id}_X\,,\,\mathrm{id}_S)} = \Sigma_{(\mathrm{id}_X\,,\,\mathrm{id}_S)} \times \Sigma_{\mathcal{G}} = 1 \times \Sigma_{\mathcal{G}} \cong \Sigma_{\mathcal{G}}$$

now with the isomorphism $i : \Sigma_{\mathcal{G}} \to 1 \times \Sigma_{\mathcal{G}}$. For the play function,

$$\mathbf{P}_{\mathcal{G} \circ_N (\mathrm{id}_X\,,\,\mathrm{id}_S)}(i\sigma) = \mathbf{P}_{\mathcal{G}}\sigma \circ \mathbf{P}_{(\mathrm{id}_X\,,\,\mathrm{id}_S)} * = \mathbf{P}_{\mathcal{G}}\sigma \circ \mathrm{id}_X = \mathbf{P}_{\mathcal{G}}\sigma$$

For coplay we have that $\mathbf{C}_{\mathcal{G} \circ_N (\mathrm{id}_X\,,\,\mathrm{id}_S)}(i\sigma)$ is the composition

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{(\mathrm{id}_X\,,\,\mathrm{id}_S)} * \otimes R} X \otimes X \otimes R$$

$$\xrightarrow{X \otimes \mathbf{C}_\mathcal{G}\sigma} X \otimes S \xrightarrow{\mathbf{C}_{(\mathrm{id}_X, \mathrm{id}_S)}*} S$$

Expanding and simplifying, this is

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{C}_\mathcal{G}\sigma} X \otimes S \xrightarrow{\pi_2} S$$

which is equal to $\mathbf{C}_\mathcal{G}\sigma$.

For best response we have

$$\begin{aligned}
& i\sigma' \in \mathbf{B}_{\mathcal{G} \circ_N (\mathrm{id}_X, \mathrm{id}_S)}(h, k)(i\sigma) \\
\iff & (*, \sigma') \in \mathbf{B}_{(\mathrm{id}_X, \mathrm{id}_S)}(h, k_{\sigma\circ})* \times \mathbf{B}_\mathcal{G}(\mathbf{P}_{(\mathrm{id}_X, \mathrm{id}_S)}* \circ h, k)\sigma \\
\iff & \sigma' \in \mathbf{B}_\mathcal{G}(\mathbf{P}_{(\mathrm{id}_X, \mathrm{id}_S)}* \circ h, k)\sigma \\
\iff & \sigma' \in \mathbf{B}_\mathcal{G}(h, k)\sigma
\end{aligned}$$

because $\mathbf{P}_{(\mathrm{id}_X, \mathrm{id}_S)}* = \mathrm{id}_X$.

In summary, we have proved that the identity for composition on an object (X, R) is $(\mathrm{id}_X, \mathrm{id}_R)$, in the category $\mathbf{Game}_N(\mathcal{C})$.

2.2.6 Associativity

Consider games

$$(W, U) \xrightarrow{\mathcal{G}} (X, T) \xrightarrow{\mathcal{H}} (Y, S) \xrightarrow{\mathcal{I}} (Z, R)$$

We will prove that

$$\mathcal{I} \circ_N (\mathcal{H} \circ_N \mathcal{G}) \sim (\mathcal{I} \circ_N \mathcal{H}) \circ_N \mathcal{G}$$

The two sets of strategy profiles are $\Sigma_{\mathcal{I} \circ_N (\mathcal{H} \circ_N \mathcal{G})} = (\Sigma_\mathcal{G} \times \Sigma_\mathcal{H}) \times \Sigma_\mathcal{I}$ and $\Sigma_{(\mathcal{I} \circ_N \mathcal{H}) \circ_N \mathcal{G}} = \Sigma_\mathcal{G} \times (\Sigma_\mathcal{H} \times \Sigma_\mathcal{I})$, so we take the isomorphism

$$i : (\Sigma_\mathcal{G} \times \Sigma_\mathcal{H}) \times \Sigma_\mathcal{I} \to \Sigma_\mathcal{G} \times (\Sigma_\mathcal{H} \times \Sigma_\mathcal{I})$$

The case for the play function follows immediately from associativity of composition in the underlying category $\mathcal{C}$:

$$\begin{aligned}
\mathbf{P}_{\mathcal{I} \circ_N (\mathcal{H} \circ_N \mathcal{G})}((\sigma, \tau), \upsilon) &= \mathbf{P}_\mathcal{I}\upsilon \circ \mathbf{P}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau) \\
&= \mathbf{P}_\mathcal{I}\upsilon \circ \mathbf{P}_\mathcal{H}\tau \circ \mathbf{P}_\mathcal{G}\sigma \\
&= \mathbf{P}_{\mathcal{I} \circ_N \mathcal{H}}(\tau, \upsilon) \circ \mathbf{P}_\mathcal{G}\sigma \\
&= \mathbf{P}_{(\mathcal{I} \circ_N \mathcal{H}) \circ_N \mathcal{G}}(i((\sigma, \tau), \upsilon))
\end{aligned}$$

For coplay, by definition $\mathbf{C}_{\mathcal{I} \circ_N (\mathcal{H} \circ_N \mathcal{G})}((\sigma, \tau), \upsilon)$ is the composition

$$W \otimes R \xrightarrow{\Delta_W \otimes R} W \otimes W \otimes R \xrightarrow{W \otimes \mathbf{P}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau) \otimes R} W \otimes Y \otimes R$$

$$\xrightarrow{W \otimes \mathbf{C}_\mathcal{I}\upsilon} W \otimes S \xrightarrow{\mathbf{C}_{\mathcal{H} \circ_N \mathcal{G}}(\sigma, \tau)} U$$

which is

$$W \otimes R \xrightarrow{\Delta_W \otimes R} W \otimes W \otimes R \xrightarrow{W \otimes \mathbf{P}_\mathcal{G}\sigma \otimes R} W \otimes X \otimes R$$

$$\xrightarrow{W\otimes\mathbf{P}_\mathcal{H}\tau\otimes R} W\otimes Y\otimes R \xrightarrow{W\otimes\mathbf{C}_\mathcal{I}\upsilon} W\otimes S \xrightarrow{\Delta_W\otimes S} W\otimes W\otimes S$$
$$\xrightarrow{W\otimes\mathbf{P}_\mathcal{G}\sigma\otimes S} W\otimes X\otimes S \xrightarrow{W\otimes\mathbf{C}_\mathcal{H}\tau} W\otimes T \xrightarrow{\mathbf{C}_\mathcal{G}\sigma} U$$

On the other hand $\mathbf{C}_{(\mathcal{I}\circ_N\mathcal{H})\circ_N\mathcal{G}}(i((\sigma,\tau),\upsilon))$ is

$$W\otimes R \xrightarrow{\Delta_W\otimes R} W\otimes W\otimes R \xrightarrow{W\otimes\mathbf{P}_\mathcal{G}\sigma\otimes R} W\otimes X\otimes R$$
$$\xrightarrow{W\otimes\mathbf{C}_{\mathcal{I}\circ_N\mathcal{H}}(\tau,\upsilon)} W\otimes T \xrightarrow{\mathbf{C}_\mathcal{G}\sigma} U$$

which is

$$W\otimes R \xrightarrow{\Delta_W\otimes R} W\otimes W\otimes R \xrightarrow{W\otimes\mathbf{P}_\mathcal{G}\sigma\otimes R} W\otimes X\otimes R$$
$$\xrightarrow{W\otimes\Delta_X\otimes R} W\otimes X\otimes X\otimes R \xrightarrow{W\otimes X\otimes\mathbf{P}_\mathcal{H}\tau\otimes R} W\otimes X\otimes Y\otimes R$$
$$\xrightarrow{W\otimes X\otimes\mathbf{C}_\mathcal{I}\upsilon} W\otimes X\otimes S \xrightarrow{W\otimes\mathbf{C}_\mathcal{H}\tau} W\otimes T \xrightarrow{\mathbf{C}_\mathcal{I}\sigma} U$$

and these two morphisms are equal by the comonoid laws for Δ. For another perspective on this part of the proof, see §2.2.14.

For best response, we have

$$\begin{aligned}
&((\sigma',\tau'),\upsilon') \in \mathbf{B}_{\mathcal{I}\circ_N(\mathcal{H}\circ_N\mathcal{G})}(h,k)((\sigma,\tau),\upsilon)\\
\iff\ &((\sigma',\tau'),\upsilon') \in \mathbf{B}_{\mathcal{H}\circ_N\mathcal{G}}(h,k_{\upsilon\circ})(\sigma,\tau)\times\mathbf{B}_\mathcal{I}(\mathbf{P}_{\mathcal{H}\circ_N\mathcal{G}}(\sigma'',\tau'')\circ h,k)\upsilon\\
\iff\ &(\sigma',\tau',\upsilon') \in \mathbf{B}_\mathcal{G}(h,(k_{\upsilon\circ})_{\tau\circ})\sigma\\
&\times\mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma\circ h,k_{\upsilon\circ})\tau\times\mathbf{B}_\mathcal{I}(\mathbf{P}_\mathcal{H}\tau\circ\mathbf{P}_\mathcal{G}\sigma\circ h,k)\upsilon
\end{aligned}$$

and

$$\begin{aligned}
&i((\sigma',\tau'),\upsilon') \in \mathbf{B}_{(\mathcal{I}\circ_N\mathcal{H})\circ_N\mathcal{G}}(h,k)(i((\sigma,\tau),\upsilon))\\
\iff\ &(\sigma',(\tau',\upsilon')) \in \mathbf{B}_\mathcal{G}(h,k_{(\tau,\upsilon)\circ})\sigma\times\mathbf{B}_{\mathcal{I}\circ_N\mathcal{H}}(\mathbf{P}_\mathcal{G}\sigma\circ h,k)(\tau,\upsilon)\\
\iff\ &(\sigma',\tau',\upsilon') \in \mathbf{B}_\mathcal{G}(h,k_{(\tau,\upsilon)\circ})\sigma\\
&\times\mathbf{B}_\mathcal{H}(\mathbf{P}_\mathcal{G}\sigma\circ h,k_{\upsilon\circ})\tau\times\mathbf{B}_\mathcal{I}(\mathbf{P}_\mathcal{H}\tau\circ\mathbf{P}_\mathcal{G}\sigma\circ h,k)\upsilon
\end{aligned}$$

Here $k_{\upsilon\circ}$ is the composition

$$Y \xrightarrow{\Delta_Y} Y\otimes Y \xrightarrow{Y\otimes\mathbf{P}_\mathcal{I}\upsilon} Y\otimes Z \xrightarrow{Y\otimes k} Y\otimes R \xrightarrow{\mathbf{C}_\mathcal{I}\upsilon} S$$

and $(k_{\upsilon\circ})_{\tau\circ}$ is the composition

$$X \xrightarrow{\Delta_X} X\otimes X \xrightarrow{X\otimes\mathbf{P}_\mathcal{H}\tau} X\otimes Y \xrightarrow{X\otimes k_{\upsilon\circ}} X\otimes S \xrightarrow{\mathbf{C}_\mathcal{H}\tau} T$$

which expands to

$$X \xrightarrow{\Delta_X} X\otimes X \xrightarrow{X\otimes\mathbf{P}_\mathcal{H}\tau} X\otimes Y \xrightarrow{X\otimes\Delta_Y} X\otimes Y\otimes Y$$
$$\xrightarrow{X\otimes Y\otimes\mathbf{P}_\mathcal{I}\upsilon} X\otimes Y\otimes Z \xrightarrow{X\otimes Y\otimes k} X\otimes Y\otimes R \xrightarrow{X\otimes\mathbf{C}_\mathcal{I}\upsilon} X\otimes S \xrightarrow{\mathbf{C}_\mathcal{H}\tau} T$$

On the other hand $k_{(\tau,\upsilon)\circ}$ is the composition

$$X \xrightarrow{\Delta_X} X\otimes X \xrightarrow{X\otimes\mathbf{P}_{\mathcal{I}\circ\mathcal{H}}(\tau,\upsilon)} X\otimes Z \xrightarrow{X\otimes k} X\otimes R \xrightarrow{\mathbf{C}_{\mathcal{I}\circ\mathcal{H}}(\tau,\upsilon)} T$$

which expands to

$$X \xrightarrow{\Delta_X} X\otimes X \xrightarrow{X\otimes\mathbf{P}_\mathcal{H}\tau} X\otimes Y \xrightarrow{X\otimes\mathbf{P}_\mathcal{I}\upsilon} X\otimes Z \xrightarrow{X\otimes k} X\otimes R$$
$$\xrightarrow{\Delta_X\otimes R} X\otimes X\otimes R \xrightarrow{X\otimes\mathbf{P}_\mathcal{H}\tau\otimes R} X\otimes Y\otimes R \xrightarrow{\mathbf{P}_\mathcal{I}\upsilon} X\otimes S \xrightarrow{\mathbf{P}_\mathcal{H}\tau} T$$

Then $(k_{\upsilon\circ})_{\tau\circ} = k_{(\tau,\upsilon)\circ}$, and we are done.

2.2.7 Tensor product of games

Besides the two variants of categorical composition we have introduced, the other aggregation operator we will consider is a tensor (or monoidal) product. Given an arbitrary pair of games

$$\mathcal{G} : (X_1, S_1) \to (Y_1, R_1)$$

and

$$\mathcal{H} : (X_2, S_2) \to (Y_2, R_2)$$

we can form their tensor product

$$\mathcal{G} \otimes \mathcal{H} : (X_1 \otimes X_2, S_1 \otimes S_2) \to (Y_1 \otimes Y_2, R_1 \otimes R_2)$$

Since the tensor product $\otimes$ of $\mathcal{C}$ can be extended componentwise to pairs of objects, this will give us a product-like operator on $\mathbf{Game}_N(\mathcal{C})$. Whereas both variants of $\mathcal{H} \circ \mathcal{G}$ behaves like the sequential play of $\mathcal{G}$ and $\mathcal{H}$, the purpose of $\mathcal{G} \otimes \mathcal{H}$ is to behave like the simultaneous play.

Just as for sequential composition, a strategy profile for an aggregate of two games consists of a strategy profile for each game, thus

$$\Sigma_{\mathcal{G} \otimes \mathcal{H}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$$

The play function can be defined using the tensor product of morphisms in $\mathcal{C}$, by

$$\mathbf{P}_{\mathcal{G} \otimes \mathcal{H}}(\sigma, \tau) = \mathbf{P}_{\mathcal{G}}\sigma \otimes \mathbf{P}_{\mathcal{H}}\tau$$

Similarly, the coplay function is

$$\mathbf{C}_{\mathcal{G} \otimes \mathcal{H}} : \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}} \to \hom_{\mathcal{C}}(X_1 \otimes X_2 \otimes R_1 \otimes R_2, S_1 \otimes S_2)$$

where $\mathbf{C}_{\mathcal{G} \otimes \mathcal{H}}(\sigma, \tau)$ is given by

$$X_1 \otimes X_2 \otimes R_1 \otimes R_2 \xrightarrow{\cong} X_1 \otimes R_1 \otimes X_2 \otimes R_2 \xrightarrow{\mathbf{C}_{\mathcal{G}}\sigma \otimes \mathbf{C}_{\mathcal{H}}\tau} S_1 \otimes S_2$$

To define the best response function

$$\begin{aligned} \mathbf{B}_{\mathcal{G} \otimes \mathcal{H}} : \hom_{\mathcal{C}}(I, X_1 \otimes X_2) \times \hom_{\mathcal{C}}(Y_1 \otimes Y_2, R_1 \otimes R_2) \to \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}} \\ \to \mathscr{P}(\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \end{aligned}$$

essentially the same reasoning applies as in §2.2.4, except that we are now working with simultaneous games, and with selection equilibria (in the sense of §1.2.3). Given $h : \hom_{\mathcal{C}}(I, X_1 \otimes X_2)$, we can take the canonical projections of it to form the component histories $h_1 : \hom_{\mathcal{C}}(I, X_1)$ and $h_2 : \hom_{\mathcal{C}}(I, X_2)$. More importantly, given a continuation $k : \hom_{\mathcal{C}}(Y_1 \otimes Y_2, R_1 \otimes R_2)$, a history $h : \hom_{\mathcal{C}}(I, X_1 \otimes X_2)$ and a strategy profile $(\sigma, \tau) : \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$, we can form smaller continuations in which each component may deviate from the strategy, while the other stays fixed. The continuation for the first player is denoted $k_{\otimes \tau(h_2)} : \hom_{\mathcal{C}}(Y_1, R_1)$, and is defined by the composition

$$Y_1 \xrightarrow{\cong} Y_1 \otimes I \xrightarrow{Y_1 \otimes h_2} Y_1 \otimes X_2 \xrightarrow{Y_1 \otimes \mathbf{P}_{\mathcal{H}}\tau} Y_1 \otimes Y_2 \xrightarrow{k} R_1 \otimes R_2 \xrightarrow{\pi_1} R_1$$

Similarly, the continuation $k_{\sigma(h_1)\otimes} : \hom_{\mathcal{C}}(Y_2, R_2)$ for the second player is

$$Y_2 \xrightarrow{\cong} I \otimes Y_2 \xrightarrow{h_1 \otimes Y_2} X_1 \otimes Y_2 \xrightarrow{\mathbf{P}_{\mathcal{G}}\sigma \otimes Y_2} Y_1 \otimes Y_2 \xrightarrow{k} R_1 \otimes R_2 \xrightarrow{\pi_2} R_2$$

Given these definitions, the best response function for the tensor product is given by

$$\mathbf{B}_{\mathcal{G}\otimes\mathcal{H}}(h, k)(\sigma, \tau) = \mathbf{B}_{\mathcal{G}}(h_1, k_{\otimes\tau(h_2)})\sigma \times \mathbf{B}_{\mathcal{H}}(h_2, k_{\sigma(h_1)\otimes})\tau$$

We will now prove that the tensor product is well-defined on equivalence classes. Take equivalences $\mathcal{G} \sim \mathcal{G}'$ and $\mathcal{H} \sim \mathcal{H}'$, so we have isomorphisms $i_{\mathcal{G}} : \Sigma_{\mathcal{G}} \to \Sigma_{\mathcal{G}'}$ and $i_{\mathcal{H}} : \Sigma_{\mathcal{H}} \to \Sigma_{\mathcal{H}'}$. We will prove that there is an equivalence $\mathcal{G} \otimes \mathcal{H} \sim \mathcal{G}' \otimes \mathcal{H}'$ given by the isomorphism $i_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau) = (i_{\mathcal{G}}\sigma, i_{\mathcal{H}}\tau)$. For the play function,

$$\mathbf{P}_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau) = \mathbf{P}_{\mathcal{G}}\sigma \otimes \mathbf{P}_{\mathcal{H}}\tau = \mathbf{P}_{\mathcal{G}'}(i_{\mathcal{G}}\sigma) \otimes \mathbf{P}_{\mathcal{H}'}(i_{\mathcal{H}}\tau) = \mathbf{P}_{\mathcal{G}'\otimes\mathcal{H}'}(i_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau))$$

For coplay,

$$\begin{aligned}
\mathbf{C}_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau) &= (\mathbf{C}_{\mathcal{G}}\sigma \otimes \mathbf{C}_{\mathcal{H}}\tau) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= (\mathbf{C}_{\mathcal{G}'}(i_{\mathcal{G}}\sigma) \otimes \mathbf{C}_{\mathcal{H}'}(i_{\mathcal{H}}\tau)) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= \mathbf{C}_{\mathcal{G}'\otimes\mathcal{H}'}(i_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau))
\end{aligned}$$

For best reponse, similarly to §2.2.3 we first check that there is an equality of continuations

$$\begin{aligned}
k_{\otimes\tau(h_2)} &= \pi_1 \circ k \circ (Y_1 \otimes \mathbf{P}_{\mathcal{H}}\tau) \circ (Y_1 \otimes h_2) \circ \rho_{Y_1}^{-1} \\
&= \pi_1 \circ k \circ (Y_1 \otimes \mathbf{P}_{\mathcal{H}'}(i_{\mathcal{H}}\tau)) \circ (Y_1 \otimes h_2) \circ \rho_{Y_1}^{-1} \\
&= k_{\otimes i_{\mathcal{H}}\tau(h_2)}
\end{aligned}$$

and similarly $k_{\sigma(h_1)\otimes} = k_{i_{\mathcal{G}}\sigma(h_1)\otimes}$. Then

$$\begin{aligned}
& i_{\mathcal{G}\otimes\mathcal{H}}(\sigma', \tau') \in \mathbf{B}_{\mathcal{G}'\otimes\mathcal{H}'}(h, k)(i_{\mathcal{G}\otimes\mathcal{H}}(\sigma, \tau)) \\
\iff & (i_{\mathcal{G}}\sigma', i_{\mathcal{H}}\tau') \in \mathbf{B}_{\mathcal{G}'}(h_1, k_{\otimes i_{\mathcal{H}}\tau(h_2)})(i_{\mathcal{G}}\sigma) \times \mathbf{B}_{\mathcal{H}'}(h_2, k_{i_{\mathcal{G}}\sigma(h_1)\otimes})(i_{\mathcal{H}}\tau) \\
\iff & (\sigma', \tau') \in \mathbf{B}_{\mathcal{G}}(h_1, k_{\otimes\tau(h_2)})\sigma \times \mathbf{B}_{\mathcal{H}}(h_2, k_{\sigma(h_1)\otimes})\tau \\
\iff & (\sigma', \tau') \in \mathbf{B}_{\mathcal{G}\otimes\mathcal{H}}(h, k)(\sigma, \tau)
\end{aligned}$$

2.2.8 Functoriality of the tensor product

We will now prove that $\otimes$ makes $\mathbf{Game}_N(\mathcal{C})$ into a symmetric monoidal category. The first step of this is to prove that $\otimes$ is a bifunctor.

The action of the monoidal product on objects is to pairwise apply the monoidal product of $\mathcal{C}$, so

$$(X_1, R_1) \otimes (X_2, R_2) = (X_1 \otimes X_2, R_1 \otimes R_2)$$

The action on morphisms, which are games, is exactly the construction given in §2.2.7. The monoidal unit is (I, I), where I is the monoidal unit of $\mathcal{C}$. The symmetric monoidal category axioms underly the string diagram language for $\mathbf{Game}_N(\mathcal{C})$ introduced in §2.3.5 and used in the remainder of this thesis.

We will first prove the identity law, namely

$$\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)} \sim \mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}$$

Since $\Sigma_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}} = 1 \times 1$ and $\Sigma_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}} = 1$, we take the isomorphism $i(*,*) = *$. For the play function,

$$\begin{aligned}
\mathbf{P}_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}}(*,*) &= \mathbf{P}_{\mathrm{id}_{(X_1,R_1)}} * \otimes \mathbf{P}_{\mathrm{id}_{(X_2,R_2)}} * \\
&= \mathrm{id}_{X_1} \otimes \mathrm{id}_{X_2} \\
&= \mathrm{id}_{X_1 \otimes X_2} \\
&= \mathbf{P}_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}} *
\end{aligned}$$

For coplay,

$$\begin{aligned}
&\mathbf{C}_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}}(*,*) \\
&= (\mathbf{C}_{\mathrm{id}_{(X_1,R_1)}} * \otimes \mathbf{C}_{\mathrm{id}_{(X_2,R_2)}} *) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= ((\mathrm{id}_{R_1} \circ \pi_{X_1 \otimes X_2 \to R_1}) \otimes (\mathrm{id}_{R_2} \circ \pi_{X_2 \otimes R_2 \to R_2})) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= (\mathrm{id}_{R_1} \otimes \mathrm{id}_{R_2}) \circ (\pi_{X_1 \otimes R_1 \to R_1} \otimes \pi_{X_2 \otimes R_2 \to R_2}) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= \mathrm{id}_{R_1 \otimes R_2} \circ \pi_{X_1 \otimes X_2 \otimes R_1 \otimes R_2 \to R_1 \otimes R_2} \\
&= \mathbf{C}_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}} *
\end{aligned}$$

where the canonical projections have been labelled with their types for clarity. For best response, we note that

$$* \in \mathbf{B}_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}}(h,k)*$$

always holds, and so does

$$\begin{aligned}
(*,*) &\in \mathbf{B}_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}}(h,k)(*,*) \\
&= \mathbf{B}_{\mathrm{id}_{(X_1,R_1)}}(h_1, k_{\otimes *(h_2)})* \times \mathbf{B}_{\mathrm{id}_{(X_2,R_2)}}(h_2, k_{*(h_1)\otimes})*
\end{aligned}$$

Now we come to the distributivity law of a bifunctor, namely

$$(\mathcal{H}_1 \circ_N \mathcal{G}_1) \otimes (\mathcal{H}_2 \circ_N \mathcal{G}_2) \sim (\mathcal{H}_1 \otimes \mathcal{H}_2) \circ_N (\mathcal{G}_1 \otimes \mathcal{G}_2)$$

We have

$$\Sigma_{(\mathcal{H}_1 \circ_N \mathcal{G}_1) \otimes (\mathcal{H}_2 \circ_N \mathcal{G}_2)} = (\Sigma_{\mathcal{G}_1} \times \Sigma_{\mathcal{H}_1}) \times (\Sigma_{\mathcal{G}_2} \times \Sigma_{\mathcal{H}_2})$$

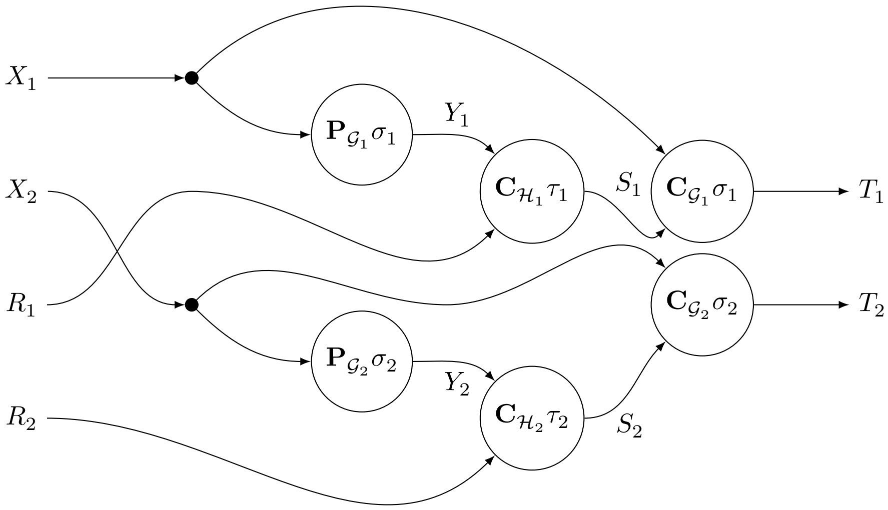
and

$$\Sigma_{(\mathcal{H}_1 \otimes \mathcal{H}_2) \circ_N (\mathcal{G}_1 \otimes \mathcal{G}_2)} = (\Sigma_{\mathcal{G}_1} \times \Sigma_{\mathcal{G}_2}) \times (\Sigma_{\mathcal{H}_1} \times \Sigma_{\mathcal{H}_2})$$

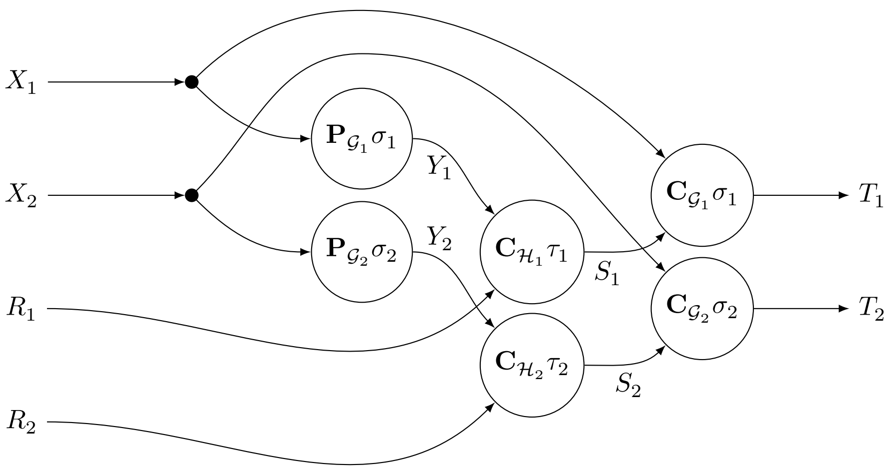
and so we take the isomorphism $i((\sigma_1,\tau_1),(\sigma_2,\tau_2)) = ((\sigma_1,\sigma_2),(\tau_1,\tau_2))$. For the play functions,

$$\begin{aligned}
\mathbf{P}_{(\mathcal{H}_1 \circ_N \mathcal{G}_1) \otimes (\mathcal{H}_2 \circ_N \mathcal{G}_2)}((\sigma_1,\tau_1),(\sigma_2,\tau_2)) &= \mathbf{P}_{\mathcal{H}_1 \circ_N \mathcal{G}_1}(\sigma_1,\tau_1) \otimes \mathbf{P}_{\mathcal{H}_2 \circ_N \mathcal{G}_2}(\sigma_2,\tau_2) \\
&= (\mathbf{P}_{\mathcal{H}_1}\tau_1 \circ \mathbf{P}_{\mathcal{G}_1}\sigma_1) \otimes (\mathbf{P}_{\mathcal{H}_2}\tau_2 \circ \mathbf{P}_{\mathcal{G}_2}\sigma_2) \\
&= (\mathbf{P}_{\mathcal{H}_1}\tau_1 \otimes \mathbf{P}_{\mathcal{H}_2}\tau_2) \circ (\mathbf{P}_{\mathcal{G}_1}\sigma_1 \otimes \mathbf{P}_{\mathcal{G}_2}\sigma_2) \\
&= \mathbf{P}_{\mathcal{H}_1 \otimes \mathcal{H}_2}(\tau_1,\tau_2) \circ \mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1,\sigma_2) \\
&= \mathbf{P}_{(\mathcal{H}_1 \otimes \mathcal{H}_2) \circ_N (\mathcal{G}_1 \otimes \mathcal{G}_2)}((\sigma_1,\sigma_2),(\tau_1,\tau_2))
\end{aligned}$$

The proof for coplay would be extremely tedious to do in a similar style, but we can instead prove it by drawing string diagrams in the symmetric monoidal category $\mathcal{C}$ [Sel11], and observing that one can be deformed into the other. The string diagram corresponding to $\mathbf{C}_{(\mathcal{H}_1 \circ_N \mathcal{G}_1) \otimes (\mathcal{H}_2 \circ_N \mathcal{G}_2)}((\sigma_1, \tau_1), (\sigma_2, \tau_2))$ is



and for $\mathbf{C}_{((\mathcal{H}_1 \otimes \mathcal{H}_2) \circ_N (\mathcal{G}_1 \otimes \mathcal{G}_2))}((\sigma_1, \sigma_2), (\tau_1, \tau_2))$ is



2.2.9 Functoriality of the tensor product, continued

We will first prove the remaining case of the distributivity law for $\circ_N$, namely that

$$((\sigma_1', \tau_1'), (\sigma_2', \tau_2')) \in \mathbf{B}_{(\mathcal{H}_1 \circ_N \mathcal{G}_1) \otimes (\mathcal{H}_2 \circ_N \mathcal{G}_2)}(h, k)((\sigma_1, \tau_1), (\sigma_2, \tau_2))$$

is equivalent to

$$((\sigma_1', \sigma_2'), (\tau_1', \tau_2')) \in \mathbf{B}_{(\mathcal{H}_1 \otimes \mathcal{H}_2) \circ_N (\mathcal{G}_1 \otimes \mathcal{G}_2)}(h, k)((\sigma_1, \sigma_2), (\tau_1, \tau_2))$$

We begin by expanding $\mathbf{B}_{(\mathcal{H}_1 \circ_N \mathcal{G}_1) \otimes (\mathcal{H}_2 \circ_N \mathcal{G}_2)}(h, k)((\sigma_1, \tau_1), (\sigma_2, \tau_2))$ to

$$\mathbf{B}_{\mathcal{H}_1 \circ_N \mathcal{G}_1}(h_1, k_{\otimes(\sigma_2, \tau_2)(h2)})(\sigma_1, \tau_1) \times \mathbf{B}_{\mathcal{H}_2 \circ_N \mathcal{G}_2}(h_2, k_{(\sigma_1, \tau_1)(h_1)\otimes})(\sigma_2, \tau_2)$$

The first term of this is

$$\mathbf{B}_{\mathcal{G}_1}(h_1, (k_{\otimes(\sigma_2, \tau_2)(h_2)})_{\tau_1 \circ})\sigma_1 \times \mathbf{B}_{\mathcal{H}_1}(\mathbf{P}_{\mathcal{G}_1}\sigma_1 \circ h_1, k_{\otimes(\sigma_2, \tau_2)(h_2)})\tau_1$$

and the second is

$$\mathbf{B}_{\mathcal{G}_2}(h_2, (k_{(\sigma_1, \tau_1)(h_1)\otimes})_{\tau_2 \circ})\sigma_2 \times \mathbf{B}_{\mathcal{H}_2}(\mathbf{P}_{\mathcal{G}_2}\sigma_2 \circ h_2, k_{(\sigma_1, \tau_1)(h_1)\otimes})\tau_2$$

The product of these can be written isomorphically as

$$\begin{aligned} &\big(\mathbf{B}_{\mathcal{G}_1}(h_1, (k_{\otimes(\sigma_2, \tau_2)(h_2)})_{\tau_1 \circ})\sigma_1 \times \mathbf{B}_{\mathcal{G}_2}(h_2, (k_{(\sigma_1, \tau_1)(h_1)\otimes})_{\tau_2 \circ})\sigma_2\big) \\ &\times \big(\mathbf{B}_{\mathcal{H}_1}(\mathbf{P}_{\mathcal{G}_1}\sigma_1 \circ h_1, k_{\otimes(\sigma_2, \tau_2)(h_2)})\tau_1 \times \mathbf{B}_{\mathcal{H}_2}(\mathbf{P}_{\mathcal{G}_2}\sigma_2 \circ h_2, k_{(\sigma_1, \tau_1)(h_1)\otimes})\tau_2\big) \end{aligned}$$

On the other hand, $\mathbf{B}_{(\mathcal{H}_1 \otimes \mathcal{H}_2) \circ_N (\mathcal{G}_1 \otimes \mathcal{G}_2)}(h, k)((\sigma_1, \sigma_2), (\tau_1, \tau_2))$ expands to

$$\mathbf{B}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(h, k_{(\tau_1, \tau_2)\circ})(\sigma_1, \sigma_2) \times \mathbf{B}_{\mathcal{H}_1 \otimes \mathcal{H}_2}(\mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1, \sigma_2) \circ h, k)(\tau_1, \tau_2)$$

for which the first term is

$$\mathbf{B}_{\mathcal{G}_1}(h_1, (k_{(\tau_1, \tau_2)\circ})_{\otimes \sigma_2(h_2)})\sigma_1 \times \mathbf{B}_{\mathcal{G}_2}(h_2, (k_{(\tau_1, \tau_2)\circ})_{\sigma_1(h_1)\otimes})\sigma_2$$

and the second is

$$\begin{aligned} &\mathbf{B}_{\mathcal{H}_1}((\mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1, \sigma_2) \circ h)_1, k_{\otimes \tau_2((\mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1, \sigma_2) \circ h)_2)})\tau_1 \\ \times\ &\mathbf{B}_{\mathcal{H}_2}((\mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1, \sigma_2) \circ h)_2, k_{\tau_1((\mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1, \sigma_2) \circ h)_1)\otimes})\tau_2 \end{aligned}$$

Comparing these, it suffices to, firstly, have equalities of histories

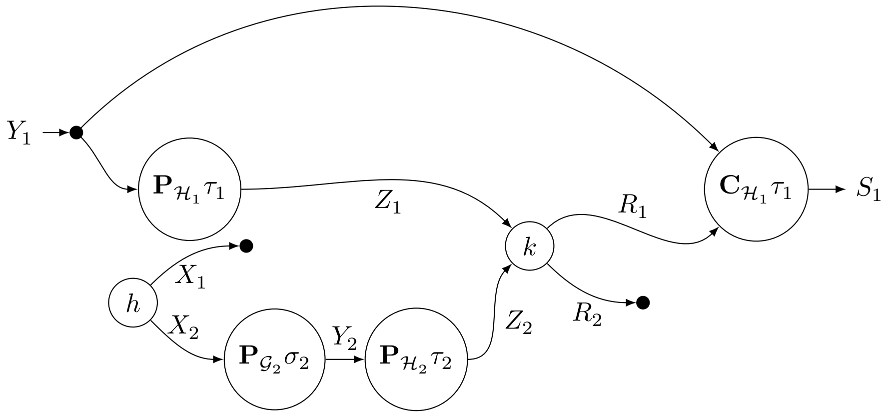
$$\mathbf{P}_{\mathcal{G}_1}\sigma_1 \circ h_1 = (\mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1, \sigma_2) \circ h)_1$$

$$\mathbf{P}_{\mathcal{G}_2}\sigma_2 \circ h_2 = (\mathbf{P}_{\mathcal{G}_1 \otimes \mathcal{G}_2}(\sigma_1, \sigma_2) \circ h)_2$$

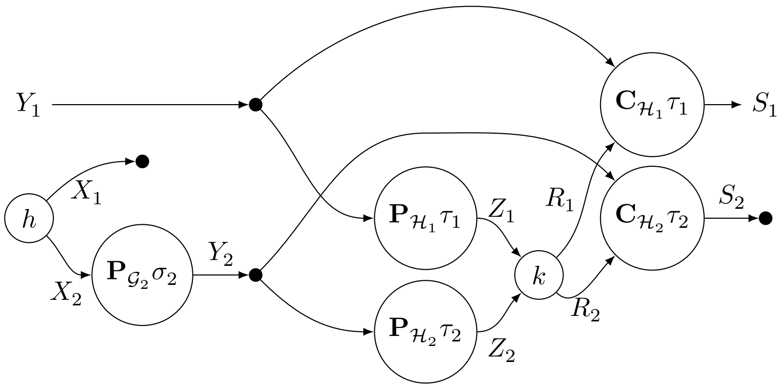
which are both immediate consequences of the comonoid axioms, and secondly to have equalities of continuations

$$\begin{aligned} (k_{\otimes(\sigma_2, \tau_2)(h_2)})_{\tau_1 \circ} &= (k_{(\tau_1, \tau_2)\circ})_{\otimes \sigma_2(h_2)} \\ (k_{(\sigma_1, \tau_1)(h_1)\otimes})_{\tau_2 \circ} &= (k_{(\tau_1, \tau_2)\circ})_{\sigma_1(h_1)\otimes} \\ k_{\otimes(\sigma_2, \tau_2)(h_2)} &= k_{\otimes \tau_2(\mathbf{P}_{\mathcal{G}_2}\sigma_2 \circ h_2)} \\ k_{(\sigma_1, \tau_1)(h_1)\otimes} &= k_{\tau_1(\mathbf{P}_{\mathcal{G}_1}\sigma_1 \circ h_1)\otimes} \end{aligned}$$

To prove these, again we draw string diagrams in $\mathcal{C}$. The first equality is the equivalence between



and



The second is symmetrical to this. Both sides of the third are directly equal to

$$Z_1 \cong Z_1 \otimes I \xrightarrow{Z_2 \otimes h_2} Z_1 \otimes X_2 \xrightarrow{Z_1 \otimes \mathbf{P}_{\mathcal{G}_2}\sigma_2} Z_1 \otimes Y_2 \xrightarrow{Z_1 \otimes \mathbf{P}_{\mathcal{H}_2}\tau_2} Z_1 \otimes Z_2$$
$$\xrightarrow{k} R_1 \otimes R_2 \xrightarrow{\pi_1} R_1$$

and again the fourth is symmetrical.

2.2.10 The monoidal category axioms

The remaining work in proving that $\mathbf{Game}_N(\mathcal{C})$ is a monoidal category is to prove the monoidal category axioms.

In general, proving these axioms takes a significant amount of work. We must define three families of morphisms, the left and right unitors and the associators, prove their naturality, and then prove the commutativity of two diagrams including the Mac Lane pentagon. To prove that a monoidal category is symmetric we must additionally define the braiding morphisms, prove their naturality, and prove commutativity of an additional three diagrams.

Most of this work can be avoided by appealing to Mac Lane's coherence theorem [Mac78] and replacing $\mathcal{C}$ with a monoidally equivalent strict monoidal category.[2] In that case we have equalities of objects

$$(I, I) \otimes (X, R) = (I \otimes X, I \otimes R) = (X, R) = (X \otimes I, R \otimes I) = (X, R) \otimes (I, I)$$

and so we can take all of the unitors to be the identity morphisms (that is, computations formed of pairs of identities, see §2.2.3), which are automatically natural and satisfy the commutative diagrams, simply by the fact that $\mathbf{Game}_N(\mathcal{C})$ is a category. Similarly we have equalities

$$\begin{aligned} ((X_1, R_1) \otimes (X_2, R_2)) \otimes (X_3, R_3) &= (X_1 \otimes X_2 \otimes X_3, R_1 \otimes R_2 \otimes R_3) \\ &= (X_1, R_1) \otimes ((X_2, R_2) \otimes (X_3, R_3)) \end{aligned}$$

and so we can also take the associators to be identities.

For the braiding morphisms we take (the equivalence class of) the computation

$$\begin{aligned} s_{(X_1,R_1),(X_2,R_2)} &= (s_{X_1,X_2}, s_{R_2,R_1}) \\ &: (X_1 \otimes X_2, R_1 \otimes R_2) \to (X_2 \otimes X_1, R_2 \otimes R_1) \end{aligned}$$

For a strict monoidal category the the unit law becomes trivial, so we must prove only the associativity and inverse laws. For this, we will use the result from §2.2.12, that computations respect N-composition and tensor. This is not circular, because we will only use the part of the result that does not already assume that $\mathbf{Game}_N(\mathcal{C})$ is monoidal, and is really shorthand for copying special cases of that proof into this section.

We will begin with the inverse law. For an arbitrary symmetric monoidal category this is

$$s_{B,A} \circ s_{A,B} = \mathrm{id}_{A \otimes B}$$

We take $A = (X_1, R_1)$ and $B = (X_2, R_2)$, and so this is

$$\begin{aligned} &(s_{X_2,X_1}, s_{R_1,R_2}) \circ_N (s_{X_1,X_2}, s_{R_2,R_1}) \\ &: (X_1 \otimes X_2, R_1 \otimes R_2) \to (X_1 \otimes X_2, R_1 \otimes R_2) \end{aligned}$$

Since computation is functorial (§2.2.12), this is

$$(s_{X_2,X_1} \circ s_{X_1,X_2}, s_{R_2,R_1} \circ s_{R_1,R_2})$$

and we can apply the inverse law of $\mathcal{C}$.

The associativity axiom for a strict symmetric monoidal category is

$$\begin{tikzcd} A \otimes B \otimes C \arrow[r, "s_{A,B} \otimes C"] \arrow[d, "s_{A,B\otimes C}"'] & B \otimes A \otimes C \arrow[dl, "B \otimes s_{A,C}"] \\ B \otimes C \otimes A & \end{tikzcd}$$

[2] This is not ideal, but it is only a 'temporary solution' before higher categories are introduced; see the conclusion.

In $\mathbf{Game}_N(\mathcal{C})$, we need to take $A = (X_1, R_1)$, $B = (X_2, R_2)$ and $C = (X_3, R_3)$.

As a lemma, we need the equations

$$(X_2, R_2) \otimes s_{(X_1,R_1),(X_3,R_3)} = (X_2 \otimes s_{X_1,X_3}, R_2 \otimes s_{R_3,R_1})$$

and

$$s_{(X_1,R_1),(X_2,R_2)} \otimes (X_3, R_3) = (s_{X_1,X_2} \otimes X_3, s_{R_2,R_1} \otimes R_3)$$

which are special cases of the fact that computation is a monoidal functor (§2.2.12).

Now, by functoriality, we have that the computations

$$((X_2, R_2) \otimes s_{(X_1,R_1),(X_3,R_3)}) \circ_N (s_{(X_1,R_1),(X_2,R_2)} \otimes (X_3, R_3))$$

and

$$((X_2 \otimes s_{X_1,X_3}) \circ (s_{X_1,X_2} \otimes X_3), (s_{R_2,R_1} \otimes R_3) \circ (R_2 \otimes s_{R_3,R_1}))$$

are equal. Therefore we need only check the equations

$$(X_2 \otimes s_{X_1,X_3}) \circ (s_{X_1,X_2} \otimes X_3) = s_{X_1,X_2 \otimes X_3}$$

and

$$(s_{R_2,R_1} \otimes R_3) \circ (R_2 \otimes s_{R_3,R_1}) = s_{R_2 \otimes R_3,R_1}$$

in $\mathcal{C}$, which both hold because $\mathcal{C}$ is symmetric monoidal.

2.2.11 Strategic triviality

Next, we will formalise some informal remarks that were made in §2.1.9. A game $\mathcal{G} : (X, S) \to (Y, R)$ will be called **strategically trivial** if it satisfies two conditions. Firstly, there must be only one strategy, so $\Sigma_{\mathcal{G}} = 1 \cong \{*\}$. Secondly, the unique strategy must be trivial, in the sense that it can never fail to be an equilibrium. That is, for all contexts $(h, k) : \hom_{\mathcal{C}}(I, X) \times \hom_{\mathcal{C}}(Y, R)$ we must have

$$i* \in \mathbf{B}_{\mathcal{G}}(h, k)(i*)$$

where $i*$ is a name for the unique element of $\Sigma_{\mathcal{G}}$, or equivalently the image of $* : 1$ under the isomorphism $i : 1 \to \Sigma_{\mathcal{G}}$. Notice that strategic triviality is well-defined on equivalence classes: if $\mathcal{G} \sim \mathcal{G}'$ and $\mathcal{G}$ is strategically trivial, then so is $\mathcal{G}'$.

We can now give more explanation for why a strategically trivial game should always be in equilibrium, rather than never: the compositional best response functions in §2.2.4 and §2.2.7 both use cartesian products, and if the set of best responses of a computation was empty, everything would cancel and the entire game would have no equilibria. Put another way, an equilibrium of an aggregate consists of an equilibrium of each component, with suitably modified contexts, and so an equilibrium overall must in particular restrict to an equilibrium on those components that are players.

We directly have that computations and counits (§2.1.9) are strategically trivial. We will now prove that strategically trivial games are closed under

N-composition and tensor. In each case, the set of strategy profiles is $1 \times 1 \cong 1$. For tensor products, the best response function is

$$\begin{aligned}\mathbf{B}_{\mathcal{G}\otimes\mathcal{H}}(h,k)(*,*) &= \mathbf{B}_{\mathcal{G}}(h_1, k_{\otimes *(h_2)})* \times \mathbf{B}_{\mathcal{H}}(h_2, k_{*(h_1)\otimes})* \\ &= \{*\} \times \{*\} = \{(*,*)\}\end{aligned}$$

For N-composition it is

$$\begin{aligned}\mathbf{B}_{\mathcal{H}\circ_N\mathcal{G}}(h,k)(*,*) &= \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})* \times \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}* \circ h, k)* \\ &= \{*\} \times \{*\} = \{(*,*)\}\end{aligned}$$

This is useful because when reasoning about strategically trivial games, we can focus only on the play and coplay functions, since we already know the strategy profiles and best response function. If $\mathcal{G} : (X,S) \to (Y,R)$ is strategically trivial we will moreover often write the play function of $\mathcal{G}$ as though it had type $\mathbf{P}_{\mathcal{G}} : \hom_{\mathcal{C}}(X,Y)$ and the coplay function as $\mathbf{C}_{\mathcal{G}} : \hom_{\mathcal{C}}(X \otimes R, S)$, leaving the strategy $*$ implicit.

N-composition of a game with a strategically trivial game is particularly simple, because the strategically trivial parts act only by transforming the history and continuation. Suppose we have a composition of games

$$(X', S') \xrightarrow{\mathcal{H}_1} (X,S) \xrightarrow{\mathcal{G}} (Y,R) \xrightarrow{\mathcal{H}_2} (Y', R')$$

in $\mathbf{Game}_N(\mathcal{C})$ where $\mathcal{H}_1$ and $\mathcal{H}_2$ are strategically trivial. Then

$$\Sigma_{\mathcal{H}_2\circ_N\mathcal{G}\circ_N\mathcal{H}_1} \cong \Sigma_{\mathcal{G}}$$

and

$$\mathbf{B}_{\mathcal{H}_2\circ_N\mathcal{G}\circ_N\mathcal{H}_1}(h,k) = \mathbf{B}_{\mathcal{G}}(\mathbf{P}_{\mathcal{H}_1} \circ h, k')$$

where

$$k' : Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y\otimes\mathbf{P}_{\mathcal{H}_2}} Y \otimes Y' \xrightarrow{Y\otimes k} Y \otimes R' \xrightarrow{\mathbf{C}_{\mathcal{H}_2}} R$$

which, when $\mathcal{C} = \mathbf{Set}$, is

$$k'y = \mathbf{C}_{\mathcal{H}_2}(y, k(\mathbf{P}_{\mathcal{H}_2}y))$$

As can be seen, $\mathcal{H}_1$ affects only the history, and $\mathcal{H}_2$ affects only the continuation.

Similarly for tensor products, if we have games $\mathcal{G} : (X_1, S_1) \to (Y_1, R_1)$ and $\mathcal{H} : (X_2, S_2) \to (Y_2, R_2)$ where $\mathcal{H}$ is strategically trivial[3], then

$$\mathbf{B}_{\mathcal{G}\otimes\mathcal{H}}(h,k)\sigma = \mathbf{B}_{\mathcal{G}}(h_1, k')\sigma$$

where

$$k' : Y_1 \cong Y_1 \otimes I \xrightarrow{Y_1\otimes h_2} Y_1 \otimes X_2 \xrightarrow{Y_1\otimes\mathbf{P}_{\mathcal{H}}} Y_1 \otimes Y_2 \xrightarrow{k} R_1 \otimes R_2 \xrightarrow{\pi_1} R_1$$

When $\mathcal{C} = \mathbf{Set}$ this is

$$\mathbf{B}_{\mathcal{G}\otimes\mathcal{H}}((h_1,h_2),k)\sigma = \mathbf{B}_{\mathcal{G}}(h_1, \lambda(y_1 : Y_1).k_1(y_1, \mathbf{P}_{\mathcal{H}}h_2))\sigma$$

[3] The case where $\mathcal{G}$ is strategically trivial is symmetric.

2.2.12 Computations as a monoidal functor

We will now prove that computation, defined in §2.1.9, gives us a faithful monoidal functor

$$(-,-) : \mathcal{C} \times \mathcal{C}^{\mathrm{op}} \hookrightarrow \mathbf{Game}_N(\mathcal{C})$$

This will be used implicitly many times in chapter 3 to simplify calculations by working in $\mathcal{C}$ rather than $\mathbf{Game}_N(\mathcal{C})$, and it is also needed to justify the syntax for objects introduced in §2.3.5, and the string diagram notation for computations in §2.3.6.

We will first prove that we have a bifunctor $\mathcal{C} \times \mathcal{C}^{\mathrm{op}} \to \mathbf{Game}_N(\mathcal{C})$.

In the product category $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ the objects are pairs of objects of $\mathcal{C}$, and the morphisms are pairs of morphisms with the second reversed. The identity morphism on the object (X, R) of $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ is $(\mathrm{id}_X, \mathrm{id}_R)$ which, lifted to a computation, is also the the identity game on (X, R) (see §2.2.3).

The composition

$$(X, T) \xrightarrow{(f_1, f_2)} (Y, S) \xrightarrow{(g_1, g_2)} (Z, R)$$

in $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ is, by definition,

$$(g_1, g_2) \circ (f_1, f_2) = (g_1 \circ f_1, f_2 \circ g_2)$$

and we must prove that the games denoted by these two expressions are equal. Since both are strategically trivial (§2.2.11), we need only check the play and coplay functions. The play functions are

$$\mathbf{P}_{(g_1 \circ f_1, f_2 \circ g_2)} : X \xrightarrow{g_1 \circ f_1} Z$$

and

$$\mathbf{P}_{(g_1, g_2) \circ_N (f_1, f_2)} : X \xrightarrow{\mathbf{P}_{(f_1, f_2)}} Y \xrightarrow{\mathbf{P}_{(g_1, g_2)}} Z$$

which are equal. The coplay functions are

$$\mathbf{P}_{(g_1 \circ f_1, f_2 \circ g_2)} : X \otimes R \xrightarrow{\pi_2} R \xrightarrow{f_2 \circ g_2} T$$

and

$$\mathbf{C}_{(g_1, g_2) \circ_N (f_1, f_2)} : X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{(f_1, f_2)} \otimes R} X \otimes Y \otimes R$$

$$\xrightarrow{X \otimes \mathbf{C}_{(g_1, g_2)}} X \otimes S \xrightarrow{\mathbf{C}_{(f_1, f_2)}} T$$

The latter is

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes f_1 \otimes R} X \otimes Y \otimes R$$

$$\xrightarrow{X \otimes \pi_2} X \otimes R \xrightarrow{X \otimes g_2} X \otimes S \xrightarrow{\pi_2} S \xrightarrow{f_2} T$$

and these are equal

Next, we must prove that the embedding also respects the monoidal structure. The monoidal unit of the product monoidal category $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ is (I, I), which is also the monoidal unit of $\mathbf{Game}_N(\mathcal{C})$ (see §2.2.10).

Suppose $(f_1, f_2) : (X_1, S_1) \to (Y_1, R_1)$ and $(g_1, g_2) : (X_2, S_2) \to (Y_2, R_2)$ are morphisms of $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$. Their monoidal product is

$$(f_1 \otimes g_1, f_2 \otimes g_2) : (X_1 \otimes X_2, S_1 \otimes S_2) \to (Y_1 \otimes Y_2, R_1 \otimes R_2)$$

We must therefore prove the equality of games

$$(f_1, f_2) \otimes (g_1, g_2) = (f_1 \otimes g_1, f_2 \otimes g_2)$$

Again both are strategically trivial, so we need only work with the play and coplay functions. For the play function we have

$$\mathbf{P}_{(f_1,f_2)\otimes(g_1,g_2)} = \mathbf{P}_{(f_1,f_2)} \otimes \mathbf{P}_{(g_1,g_2)} = f_1 \otimes g_1 = \mathbf{P}_{(f_1\otimes g_1, f_2\otimes g_2)}$$

For coplay, the former is

$$\mathbf{C}_{(f_1,f_2)\otimes(g_1,g_2)} : X_1 \otimes X_2 \otimes R_1 \otimes R_2 \xrightarrow{\cong} X_1 \otimes R_1 \otimes X_2 \otimes R_2$$
$$\xrightarrow{\mathbf{C}_{(f_1,f_2)}\otimes \mathbf{C}_{(g_1,g_2)}} S_1 \otimes S_2$$

which is

$$X_1 \otimes X_2 \otimes R_1 \otimes R_2 \xrightarrow{\cong} X_1 \otimes R_1 \otimes X_2 \otimes R_2$$
$$\xrightarrow{\pi_2 \otimes \pi_2} R_1 \otimes R_2 \xrightarrow{f_2 \otimes g_2} S_1 \otimes S_2$$

and the latter is

$$\mathbf{C}_{(f_1\otimes g_1, f_2\otimes g_2)} : X_1 \otimes X_2 \otimes R_1 \otimes R_2 \xrightarrow{\pi_2} R_1 \otimes R_2 \xrightarrow{f_2 \otimes g_2} S_1 \otimes S_2$$

which is equal.

2.2.13 The counit law

We will now prove a result, which we will return to in §2.3.6, that connects computations with the counit game, both introduced in §2.1.9. Let $f : \hom_{\mathcal{C}}(X, Y)$. Then the following diagram commutes in $\mathbf{Game}_N(\mathcal{C})$:

$$\begin{tikzcd}
(X, Y) \arrow[r, "(\mathrm{id}_X, f)"] \arrow[d, "(f, \mathrm{id}_Y)"'] & (X, X) \arrow[d, "\tau_X"] \\
(Y, Y) \arrow[r, "\tau_Y"] & (I, I)
\end{tikzcd}$$

This will be called the **counit law**. Since both games are strategically trivial, we need only check the behaviour of the play and coplay functions.

The definitions in §2.1.9 give us

$$\begin{aligned} \mathbf{P}_{(\mathrm{id}_X, f)} &: X \xrightarrow{\mathrm{id}_X} X & \mathbf{C}_{(\mathrm{id}_X, f)} &: X \otimes X \xrightarrow{\pi_2} X \xrightarrow{f} Y \\ \mathbf{P}_{(f, \mathrm{id}_Y)} &: X \xrightarrow{f} Y & \mathbf{C}_{(f, \mathrm{id}_Y)} &: X \otimes Y \xrightarrow{\pi_2} Y \xrightarrow{\mathrm{id}_Y} Y \\ \mathbf{P}_{\varepsilon_X} &: X \xrightarrow{!_X} I & \mathbf{C}_{\varepsilon_X} &: X \otimes I \xrightarrow{\pi_1} X \\ \mathbf{P}_{\varepsilon_Y} &: Y \xrightarrow{!_Y} I & \mathbf{C}_{\varepsilon_Y} &: Y \otimes I \xrightarrow{\pi_1} Y \end{aligned}$$

Composing these sequentially gives us the play functions

$$\begin{aligned} \mathbf{P}_{\tau_X \circ_N (\mathrm{id}_X, f)} &: X \xrightarrow{\mathrm{id}_X} X \xrightarrow{!_X} I \\ \mathbf{P}_{\tau_Y \circ_N (f, \mathrm{id}_Y)} &: X \xrightarrow{f} Y \xrightarrow{!_Y} I \end{aligned}$$

which are equal, and the coplay functions

$$\mathbf{C}_{\varepsilon_X \circ_N (\mathrm{id}_X, f)} : X \otimes I \xrightarrow{\Delta_X \otimes I} X \otimes X \otimes I \xrightarrow{X \otimes \pi_1} X \otimes X \xrightarrow{\pi_2} X \xrightarrow{f} Y$$

and

$$\begin{aligned} \mathbf{C}_{\varepsilon_Y \circ_N (f, \mathrm{id}_Y)} : X \otimes I \xrightarrow{\Delta_X \otimes I} X \otimes X \otimes I \xrightarrow{X \otimes f \otimes I} X \otimes Y \otimes I \\ \xrightarrow{X \otimes \pi_1} X \otimes Y \xrightarrow{\pi_2} Y \end{aligned}$$

which are both equal to

$$X \otimes I \xrightarrow{\pi_1} X \xrightarrow{f} Y$$

The game denoted by these two equal expressions is important, because when post-composed with another game it will behave like a continuation, and we will generally use it when f is an outcome function. Let $\mathcal{G} : (X, S) \to (Y, R)$ be a game, and let $k : \mathrm{hom}_{\mathcal{C}}(Y, R)$ be a continuation for $\mathcal{G}$. Consider the game

$$(X, S) \xrightarrow{\mathcal{G}} (Y, R) \xrightarrow{(k, \mathrm{id}_R)} (R, R) \xrightarrow{\varepsilon_R} (I, I)$$

which by the counit law, can be equivalently written

$$(X, S) \xrightarrow{\mathcal{G}} (Y, R) \xrightarrow{(\mathrm{id}_Y, k)} (Y, Y) \xrightarrow{\varepsilon_Y} (I, I)$$

Then

$$\Sigma_{\varepsilon_R \circ_N (k, \mathrm{id}_R) \circ_N \mathcal{G}} \cong \Sigma_{\mathcal{G}}$$

and for any $h : \mathrm{hom}_{\mathcal{C}}(I, X)$ and $\sigma : \Sigma_{\mathcal{G}}$ we have

$$\mathbf{B}_{\mathcal{G}}(h, k)\sigma \cong \mathbf{B}_{\varepsilon_R \circ_N (k, \mathrm{id}_R) \circ_N \mathcal{G}}(h, \mathrm{id}_I)\sigma$$

under the same isomorphism.

To see this, the right hand side by the definition in §2.2.4 is

$$\begin{aligned} &\mathbf{B}_{\varepsilon_R \circ_N (k, \mathrm{id}_R) \circ_N \mathcal{G}}(h, \mathrm{id}_I)\sigma \\ &= \mathbf{B}_{\mathcal{G}}(h, (\mathrm{id}_I)_{*\circ})\sigma \times \bigcap_{\sigma' : \Sigma_{\mathcal{G}}} \mathbf{B}_{\varepsilon_R \circ (k, \mathrm{id}_R)}(\mathbf{P}_{\mathcal{G}}\sigma' \circ h, \mathrm{id}_I)* \end{aligned}$$

and, since $\varepsilon_R \circ_N (k, \mathrm{id}_R)$ is strategically trivial, this is

$$\mathbf{B}_\mathcal{G}(h, (\mathrm{id}_I)_{*\circ})\sigma \times \{*\}$$

The final step is to see that $(\mathrm{id}_I)_{*\circ} = k$. By definition it is

$$(\mathrm{id}_I)_{*\circ} : Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes \mathbf{P}_{\varepsilon_R \circ (k, \mathrm{id}_R)}} Y \otimes I \xrightarrow{Y \otimes \mathrm{id}_I} Y \otimes I \xrightarrow{C_{\varepsilon_R \circ (k, \mathrm{id}_R)}} R$$

which reduces to

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes !_Y} Y \otimes I \xrightarrow{\pi_1} Y \xrightarrow{k} R$$

and hence to k.

The equations

$$\mathbf{B}_{\varepsilon_R \circ_N (k, \mathrm{id}_R) \circ_N \mathcal{G}}(h, \mathrm{id}_I)\sigma \cong \mathbf{B}_\mathcal{G}(h, k)\sigma \cong \mathbf{B}_{\varepsilon_Y \circ_N (\mathrm{id}_Y, k) \circ_N \mathcal{G}}(h, \mathrm{id}_I)\sigma$$

are very important because they allow us to move between 'internal' and 'external' views of the continuation k. We will see this used several times in §2.3, §3.1 and §3.2, when k is an outcome or utility function.

2.2.14 Information flow in games

In §2.1 and §2.2 we have given several definitions whose intuitive justifications have been incomplete at best. This will remain the case until chapter 3, in which we will demonstrate that game theory can be done inside the category $\mathbf{Game}_N(\mathcal{C})$. However we would also like to have a separate justification of the 'low level' operations, especially for readers whose motivations come from category theory.

There is, however, one known and very surprising connection to existing mathematics. The data $(\Sigma_\mathcal{G}, \mathbf{P}_\mathcal{G}, \mathbf{C}_\mathcal{G}, \mathbf{B}_\mathcal{G})$ specifying a game $\mathcal{G}$ could be divided into two kinds of data: the strategy profiles $\Sigma_\mathcal{G}$ and best response functions $\mathbf{B}_\mathcal{G}$ are motivated in terms of game theory, whereas the play and coplay functions $\mathbf{P}_\mathcal{G}\sigma$, $\mathbf{C}_\mathcal{G}\sigma$ for a fixed strategy σ are used for *information flow*. These latter are strongly reminiscent of *dialectica categories*.

The dialectica categories are categorical models of intuitionistic logic introduced in [dP91], which are based on Gödel's dialectica interpretation [AF98]. The dialectica category $\mathbf{D}(\mathbf{Set})$ has as objects relations $\varphi \subseteq X \times R$, which in particular specifies a pair[4] of sets (X, R). A morphism from $\varphi \subseteq X \times S$ to $\psi \subseteq Y \times R$ is a pair (f, g) where

$$f : X \to Y$$

and

$$g : X \times R \to S$$

[4]There are several constructions in category theory besides dialectica categories that take objects to be pairs, generally to obtain some form of duality. Chu spaces are already known to be related to dialectica categories [dP07]. The Int-construction [JSV96] is better known, but has a quite different structure. That paper explicitly says that the pair (X, R) should be considered as a formalisation of $X \otimes R^*$, as we will begin to do in §2.3.5. For remarks on duality in $\mathbf{Game}_N(\mathcal{C})$, see §2.3.6 and §2.3.7.

such that for all $x : X$ and $r : R$ we have

$$\varphi(x, g(x, r)) \implies \psi(fx, r)$$

This has a notable similarity with $f = \mathbf{P}_\mathcal{G}\sigma$ and $g = \mathbf{C}_\mathcal{G}\sigma$, if we take the relations to be trivial (either both empty or both full). More importantly, the definition of categorical composition is essentially the same in $\mathbf{Game}_N(\mathcal{C})$ as in $\mathbf{D}(\mathcal{C})$, so the definition that was claimed in §2.2.3 to not be intuitively justifiable is in fact largely an instance of something already known, and the proofs in §2.2.5 and §2.2.6 partly resemble the corresponding proofs for dialectica categories.

It should be mentioned that dialectica categories (and functional interpretations in proof theory) can be seen in game *semantic* terms, see for example [Bla97] and [Hed15a]. The relations φ and ψ are seen as games, in the game-semantic sense of two-player win/lose games, with (f, g) being a strategy for the first player (proponent) and (x, r) being a strategy for the second player (opponent), in a relative game $\varphi \to \psi$. Then the existence of a morphism $\varphi \to \psi$, which logically relates to provability of the implication, amounts to the existence of a winning strategy for the first player for ψ relative to φ. However, this apparent connection between game theory and game semantics seems to be only coincidental (see §0.3).

Finally, morphisms between indexed containers [AGH+06] have a similar form again, and a corresponding game-semantic view appears in [Hyv14].

2.3 String diagrams

2.3.1 Discussion

String diagrams are a graphical calculus that can be used to visualise information flow in monoidal categories. Their earliest appearance may be Penrose's graphical tensor notation in [Pen71], with another precursor being Girard's proof nets for linear logic [Gir87], and the mathematical foundations were formalised in [JS91]. They became well known through the work of Samson Abramsky, Bob Coecke and others on quantum information theory [Coe11], and later through the work of Bob Coecke, Mehrnoosh Sadrzadeh and others on distributional semantics in linguistics [HSG13]. String diagrams are also being applied in other areas such as bialgebra [McC12] and computability [Pav13].

As is the case in quantum physics and linguistics, string diagrams can be used in game theory to *visualise information flow*. This visualisation is a separate issue to compositionality (although being able to compose string diagrams is a crucial requirement), and is a separate contribution of this thesis. The concept of information flow in games will only be introduced informally and by example, starting in §2.3.9. The effect of the counit game on information flow, described in §2.3.6, is particularly interesting, but again will only be discussed informally.

For readers unfamiliar with string diagrams, §2.3.2 and §2.3.3 introduce them in an informal way that will be sufficient for our purposes. Alternatively, [BS10] is a good introduction. A survey of the many types of string diagrams is given in [Sel11], although the exact variant we introduce in §2.3.5 does not match any of the usual definitions.

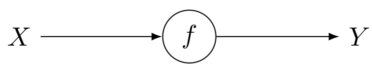
The purpose of string diagrams varies by discipline. In quantum information theory, the emphasis is generally on the ability of string diagrams to reduce complex calculations in tensor calculus to trivial topological deformation, a point made forcibly in [Coe05]. In linguistics, there is more emphasis on the use of string diagrams as a device for visualising the logical structure of sentences, whereas the underlying categorical structure is used in a more formal way by considering functorial semantics.

In game theory, we will similarly emphasise string diagrams as a visualisation tool. However, more so than in linguistics, the algebraic expressions denoting even simple games can be quite complicated, and we will make use of string diagrams as a *tool for making definitions*. A good example is in §3.2.2, where we will define a particular game by its (simple) string diagram, and the subsequent work to compute the denotation of the string diagram is quite involved. However, this work could quite easily be automated, as described in the conclusion.

It should be noted that this thesis does not contain a theorem that characterises exactly the topological moves which are allowed on string diagrams; many are proved to be valid, and some are known to be invalid, but it has not been proved that this exhausts all possible moves. In the absence of such a theorem, we cannot strictly speak about 'the game denoted by a string diagram', because it cannot be ruled out that a topologically equivalent string diagram denotes a different game. Therefore every string diagram should, for the time being, be accompanied by an algebraic term (with operators $\circ_N$ and $\otimes$) showing the intended reading.

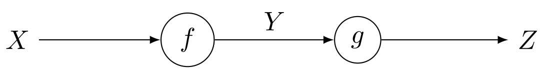
2.3.2 String diagrams for monoidal categories

The basic components of string diagrams are **strings** and **beads**. A simple string diagram has the form



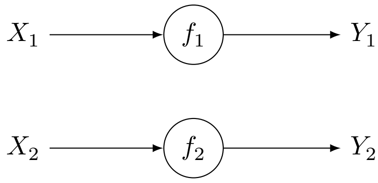
This string diagram *denotes* a morphism $f : X \to Y$ in some monoidal category. The strings in the diagram are labelled by objects of the category, and the beads by morphisms.

The composition of the category is denoted by end-to-end composition of string diagrams. If we have morphisms $f : X \to Y$ and $g : Y \to Z$, the composition $g \circ f : X \to Z$ is denoted by

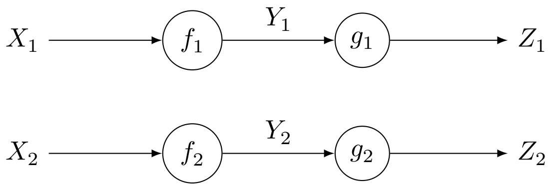


Notice that the associativity of composition is trivialised, because there is only one way to graphically compose three morphisms. This is the simplest example of a powerful fact about string diagrams, that coherence conditions in a category are reduced to graphical identity or topological deformation.

The other way of composing morphisms, namely the tensor product, is denoted by side-by-side composition. If we have morphisms $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$, the tensor product $f_1 \otimes f_2 : X_1 \otimes X_2 \to Y_1 \otimes Y_2$ is denoted by

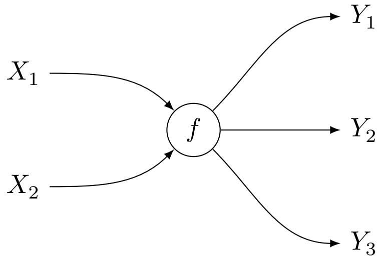


Again, notice that the associator of the tensor product is reduced to graphical identity. According to these rules, the string diagram

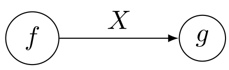


could denote either $(g_1 \circ f_1) \otimes (g_2 \circ f_2)$ or $(g_1 \otimes g_2) \circ (f_1 \otimes f_2) : X_1 \otimes X_2 \to Z_1 \otimes Z_2$, but these are equal.

A morphism $f : X_1 \otimes X_2 \to Y_1 \otimes Y_2 \otimes Y_3$ is denoted by a bead with two strings entering on the left and three exiting on the right:



The unit object of the tensor product is denoted by empty space. For example, if we have morphisms $f : I \to X$ and $g : X \to I$, the composition $g \circ f : I \to I$ is denoted by

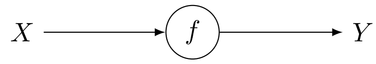


The identity morphism on an object X is denoted by simply a string labelled with X. The coherence laws for the identity morphisms and unit object are again reduced to topological deformations.

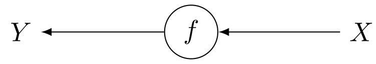
2.3.3 Compact closed categories

An important structure that a monoidal category $\mathcal{C}$ can carry, which has a particularly elegant interpretation in string diagrams, is that of a compact closed category. In a compact closed category we have a **duality**, which is a monoidal functor $-^* : \mathcal{C} \to \mathcal{C}^{\mathrm{op}}$. The prototypical example of a compact closed category is the category of finite-dimensional vector spaces and linear maps over a fixed field, with duality given by duality of vector spaces; another well known example is the category of sets and relations, with duality being the identity on sets, and giving inverse relations.

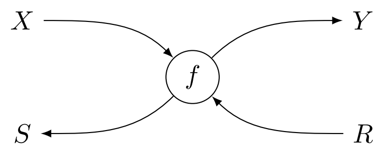
Whereas an object X is denoted by an X-labelled string running from left to right, the dual object X^* is denoted by an X-labelled string running from right to left. Then, given a morphism $f : X \to Y$ denoted by



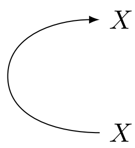
we will denote the dual morphism $f^* : Y^* \to X^*$ by its rotation:



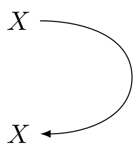
The tensor product is still denoted by side-by-side composition, and so for example a morphism $f : X \otimes S^* \to Y \otimes R^*$ in a compact closed category would be denoted by



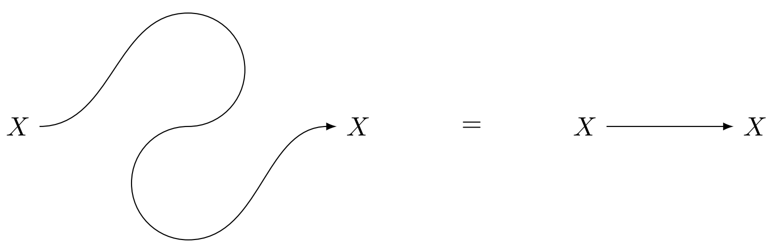
We additionally have unit morphisms $\eta_X : I \to X \otimes X^*$ denoted by a **cap**



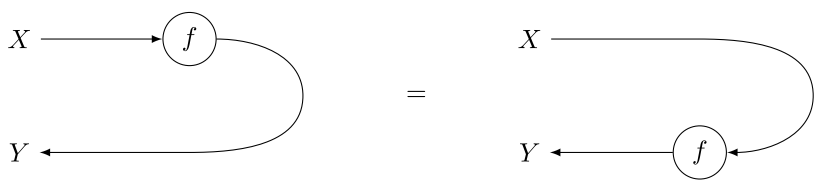
and counits $\varepsilon_X : X \otimes X^* \to I$ denoted by a **cup**



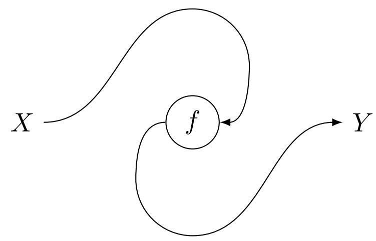
The axioms of a compact closed category specify precisely that the units and counits behave in graphically intuitive ways. For example, a bent string can be straightened (this is called the 'yanking equation'),



and a bead can be slid around a cap or cup:



Combining these, we can topologically deform a string diagram in arbitrary ways, for example we can equally denote a morphism $f : X \to Y$ by the string diagram

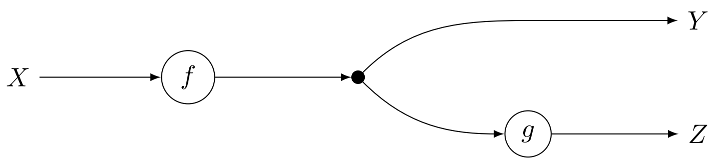


2.3.4 Boxing and compositionality

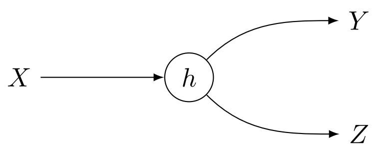
A simple but very important observation is that string diagrams are inherently fully compositional. Given an arbitrary string diagram, we could make it definitionally equal to a single bead, preserving only the strings entering and leaving the diagram. Such strings will be called **open ports**. To give an example, suppose we have morphisms $f : X \to Y$ and $g : Y \to Z$, and we define a new morphism

$$(Y \otimes g) \circ \Delta_Y \circ f : X \to Y \otimes Z$$

which is denoted by the string diagram



This string diagram has an X-labelled open port entering to the left, and Y- and Z-labelled open ports leaving to the right. If we define $h : X \to Y \otimes Z$ to be this morphism, then we can replace this string diagram with



This amounts to *forgetting* the definition of h and remembering only its type, that is to say, the types, order and variance of its open ports.

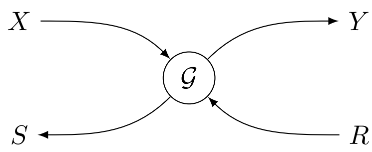
We can imagine that if we *zoom in* to the h-labelled bead in this string diagram, we will come to the previous string diagram, with the open ports being *physically the same as* the strings attached to the h-labelled bead (see the conclusion section).

Conversely, the act of obtaining the lower string diagram from the upper amounts to drawing a box around the entire diagram, so that the points at which the open strings intersect the box become exactly the points at which the strings enter and leave the h-labelled bead. We will refer to this as **boxing** the string diagram, and it is the graphical analogue of making a definitional equality.

Boxing is the aspect of compositionality that can be used to work in a scalable way. In a more serious example, the morphism h could be a very complicated process, which we can then use and re-use without worrying (or even knowing) how it is defined. In software engineering, the boxed morphism would variously be known as a 'type', a 'signature' or an 'interface'.

2.3.5 The geometry of games

The idea behind string diagrams in game theory is that although $\mathbf{Game}_N(\mathcal{C})$ does not have as much structure as the categories usually considered with string diagrams, we will abuse notation *as though* we had this structure. A game $\mathcal{G} : (X, S) \to (Y, R)$ will be denoted by the string diagram



That is, we are *pretending* that $\mathbf{Game}_N(\mathcal{C})$ has a duality $-^*$, and that $\mathcal{G}$ is a morphism $\mathcal{G} : X \otimes S^* \to Y \otimes R^*$.

This is a reasonable thing to do. If we define an operation on objects by $(X, R)^* = (R, X)$, we find that this operation can be extended in a compatible way to some morphisms, but not all. Thus we can consider $-^*$ to be a partial functor, which is total on objects (that is, a 'functor' whose underlying function on morphisms is weakened to only a partial function; alternatively it is a functor defined on a suitable wide subcategory). Since we have a symmetric monoidal embedding $\mathcal{C} \hookrightarrow \mathbf{Game}_N(\mathcal{C})$ (§2.2.12), we can write X and R for (X, I) and (R, I). Thus $R^* = (I, R)$, and so $X \otimes R^* = (X, I) \otimes (I, R) = (X \otimes I, I \otimes R)$ which, treating natural isomorphism as though it is identity, gives $X \otimes R^* = (X, R)$.

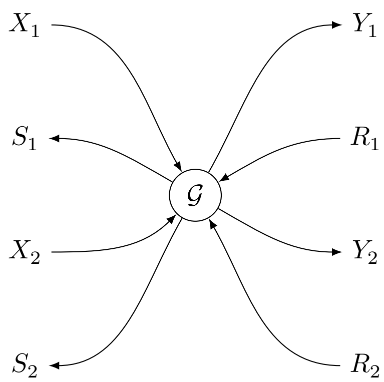
We will extend this notation in several intuitive ways. For example, if we have a game

$$\mathcal{G} : (X_1 \otimes X_2, S_1 \otimes S_2) \to (Y_1 \otimes Y_2, R_1 \otimes R_2)$$

we may write it as

$$\mathcal{G} : X_1 \otimes S_1^* \otimes X_2 \otimes S_2^* \to Y_1 \otimes R_1^* \otimes Y_2 \otimes R_2^*$$

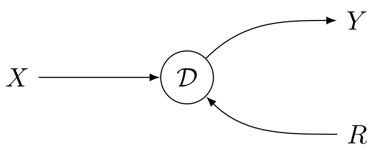
(among many other possibilities, one for each permutation), and the string diagram as



In general, we will be unconcerned but consistent with the order of the types in such situations.

Of course this comes at a cost: not all topological operations on our string diagrams preserve meaning, or even well-formedness. This is something we will need to take extra care about. For example, in a compact closed category there is no real distinction between 'past' and 'future', but in $\mathbf{Game}_N(\mathcal{C})$ they are distinct: a game $\mathcal{G} : X \to Y$ cannot be turned into $\mathcal{G} : I \to X^* \otimes Y$. An explanation for this is given in §2.3.12.

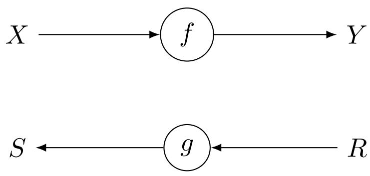
We will omit some strings when some of the types involved are the monoidal unit. For example, if we have a decision $\mathcal{D} : (X, I) \to (Y, R)$ then we will write $\mathcal{D} : X \to Y \otimes R^*$, and the string diagram denoting it is



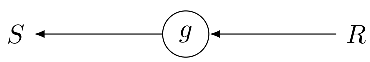
A closed game could be denoted simply by a bead with no strings attached.

2.3.6 Partial duality

A computation $(f, g) : (X, S) \to (Y, R)$ will be denoted by the string diagram

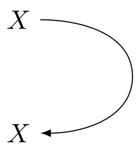


That is, we are *pretending* that $(f, g) = f \otimes g^*$, where $g^* : S^* \to R^*$ is the dual of g. This is another case of treating $-^*$ as a partial functor, this time defined on morphisms of $\mathcal{C}$. From this point onwards we will no longer treat computations formally as pairs (f, g), but as individual games f and g^* which we tensor together when necessary. This is an extension of the fact that computation is a monoidal embedding (§2.2.12). For example, if $g : R \to S$ then we can write $g^* = (\mathrm{id}_I, g) : S^* \to R^*$ and use the string diagram



since id_I is denoted by empty space.

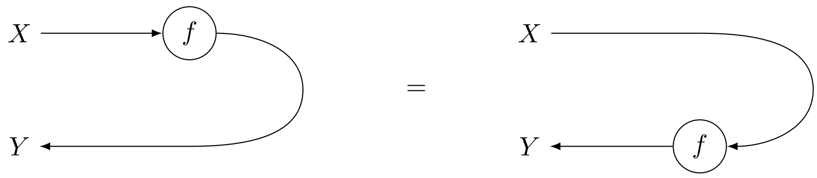
The counit game $\varepsilon_X : (X, X) \to (I, I)$ can now be written $\varepsilon_X : X \otimes X^* \to I$, and will be denoted by a cup



The counit law

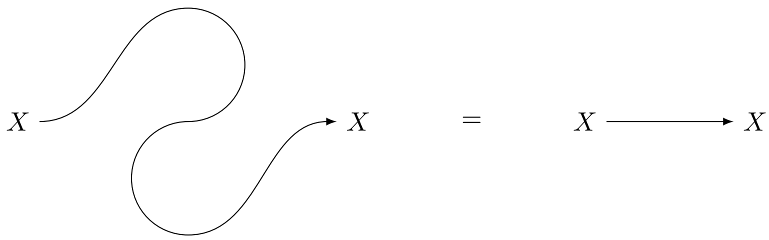
$$\varepsilon_Y \circ (f, \mathrm{id}_Y) = \varepsilon_X \circ (\mathrm{id}_X, f)$$

which we proved in §2.2.13 now appears as a coherence law



telling us that we can slide a computation around the counit like a bead. This takes the same form as the coherence law for the counit of a compact closed category (see §2.3.3), but applies only to computations.

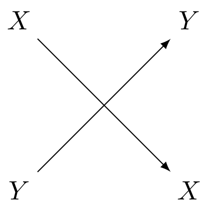
The partiality of the duality means that, unlike in a compact closed category, we cannot rotate beads. For example, in a compact closed category we have a valid equation



but we cannot do so here because the cap does not denote a morphism of $\mathbf{Game}_N(\mathcal{C})$, so the left hand side is not a well formed string diagram.

2.3.7 Covariance, contravariance and symmetries

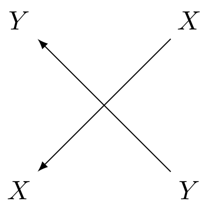
This kind of duality has one strange consequence, concerning how covariant and contravariant (or forward and backward) strings interact. If we have two strings pointing in the same direction which cross, such as



then the denoted morphism is a symmetry $\sigma_{X,Y} : X \otimes Y \to Y \otimes X$, or more formally, the computation

$$(\sigma_{X,Y}, \mathrm{id}_I) : (X \otimes Y, I) \to (Y \otimes X, I)$$

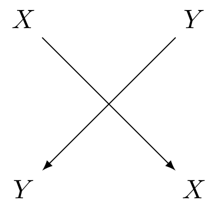
Similarly, in the contravariant direction, the string diagram



denotes $\sigma^*_{X,Y} : Y^* \otimes X^* \to X^* \otimes Y^*$, or again more formally,

$$(\mathrm{id}_I, \sigma_{X,Y}) : (I, Y \otimes X) \to (I, X \otimes Y)$$

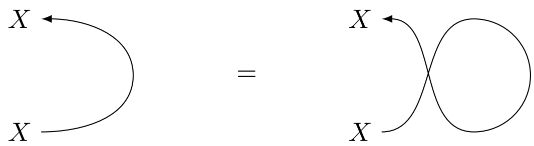
However, now consider the string diagram



The object on the left is $X \otimes Y^*$, which formally denotes (X, Y). The object on the right is $Y^* \otimes X$, which again formally denotes (X, Y). Thus, the denotation σ_{X,Y^*} of this string diagram is an identity, not a symmetry.

More generally, if we have a string diagram which consists only of crossings of covariant and contravariant strings, the denoted game is an identity. We will see this in practice in the coming sections, for example in §2.3.9.

Another example of the same idea is drawing a counit with the opposite orientation than in §2.3.6. This is justified by the topological deformation



where the right hand side is the composition $\varepsilon_X \circ \sigma_{X^*,X}$, because the symmetry $\sigma_{X^*,X}$ is trivial.

2.3.8 Copying and deleting information

The underlying categories $\mathcal{C}$ introduced in §2.1.2 are not compact closed in general, but they do have one piece of additional structure assumed, namely that every object is a cocommutative coalgebra in a canonical way. This means that we have canonical morphisms

$$!_X : X \to I$$

and

$$\Delta_X : X \to X \otimes X$$

for each object X. These morphisms can be lifted into $\mathbf{Game}_N(\mathcal{C})$ either covariantly, or contravariantly as

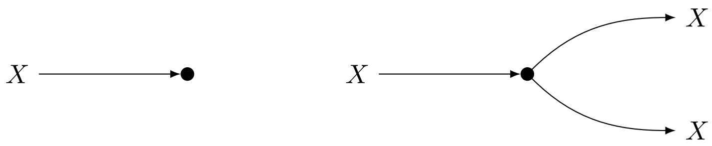
$$!^*_X : I \to X^*$$

and

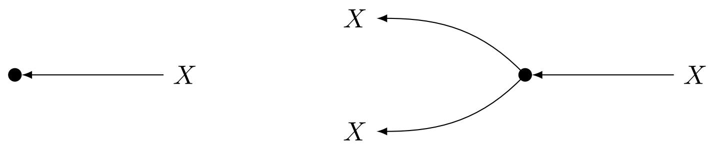
$$\Delta^*_X : X^* \otimes X^* \to X^*$$

Thus, the covariant objects[5] of $\mathbf{Game}_N(\mathcal{C})$ are cocommutative coalgebras, and contravariant objects are commutative algebras.

As is usual with string diagrams, we will denote the algebraic and coalgebraic operators by small filled circles. That is, the coalgebraic operators on a covariant object are denoted by



and the algebraic operators on a contravariant object by their rotations



These operators are not as ubiquitous in game theory as they are, for example, in quantum information theory, but nearly every game will involve copying information at some point in its definition, for example if both a player and a utility function need to be aware of the same value.

2.3.9 A bimatrix game

We have now, finally, built up enough theory to give some recognisable, textbook examples of games. We will begin with a bimatrix game, with mixed strategy Nash equilibrium as the solution concept.

Bimatrix games are two player classical games with mixed strategies, introduced in §1.2.6. Thus we have two players, who simultaneously make choices from sets X, Y. For simplicity, we will assume that these are finite. We additionally have utility functions $q_1, q_2 : X \times Y \to \mathbb{R}$ that give the utility for each player on each play.

The best response function for such a game is

$$\mathbf{B}(\sigma_1, \sigma_2) = \left\{ (\sigma_1', \sigma_2') : \mathscr{D}X \times \mathscr{D}Y \;\middle|\; \begin{array}{l} \mathbb{E}[q_1(\sigma_1', \sigma_2)] \geq \mathbb{E}[q_1(x', \sigma_2)], \\ \mathbb{E}[q_2(\sigma_1, \sigma_2')] \geq \mathbb{E}[q_2(\sigma_1, y')] \\ \text{for all } x' : X \text{ and } y' : Y \end{array} \right\}$$

and a mixed strategy Nash equilibrium is a fixpoint of this $\mathbf{B}$.

A prototypical example of a bimatrix game is matching pennies. In this example, we have $X = Y = \{H, T\}$, with

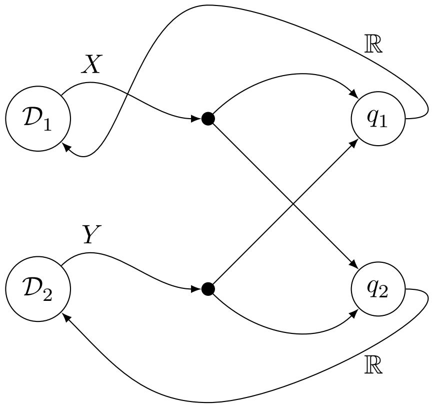
$$q_1(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

and $q_2(x, y) = 1 - q_1(x, y)$. The unique mixed strategy Nash equilibrium of this game is the strategy profile in which both players choose either H or T

[5] A 'covariant object' is an object of the form (X, I), and a 'contravariant object' is of the form (I, X). A general object is the tensor product of a covariant part and a contravariant part, and carries neither an algebra or a coalgebra structure.

with probability $\frac{1}{2}$, with the expected utility for both players being $\frac{1}{2}$. Other well known examples of bimatrix games include the prisoner's dilemma and the chicken game, and can be found in myriad books and lecture notes on introductory game theory.

In order to model a bimatrix game, let $\mathcal{G}$ be the game in $\mathbf{Game}_N(\mathbf{SRel})$ denoted by the string diagram



This diagram is built from pieces that have been introduced, composed together using $\circ_N$ and $\otimes$:

- The decisions $\mathcal{D}_1 : I \to X \otimes \mathbb{R}^*$ and $\mathcal{D}_2 : I \to Y \otimes \mathbb{R}^*$ are expected utility maximising decisions (§2.1.8), both with histories of type I.
- The two black nodes are comultiplications (§2.3.8).
- The two crossing points are symmetries, one of which is trivial and one of which is nontrivial (§2.3.7).
- The beads labelled q_1 and q_2 are the utility functions considered as covariant computations (§2.3.6).
- The two counits are drawn with opposite orientations (§2.3.7).

If we calculate the type of strategies of $\mathcal{G}$, we indeed get $\Sigma_\mathcal{G} = \mathscr{D}X \times \mathscr{D}Y$. Because $\mathcal{G} : I \to I$ is closed game (see §2.1.6), the best response function has type

$$\mathbf{B}_\mathcal{G} : \Sigma_\mathcal{G} \to \mathscr{P}\Sigma_\mathcal{G}$$

Moreover, $\mathbf{B}_\mathcal{G}$ is equal to the best response function $\mathbf{B}$ of the bimatrix game we began with. In particular, the equilibria of $\mathcal{G}$ are exactly the mixed strategy Nash equilibria of the bimatrix game. The proof of this is quite involved, and is mostly given in §3.1.7.

2.3.10 A sequential game

As a second example, we will model a two-player game of perfect information, with players modelled by arbitrary multi-valued selection functions, giving an instance of the sequential games defined in §1.3.4. Let X, Y and R be arbitrary sets, and consider the game defined by the outcome function $q : X \times Y \to R$ and the multi-valued selection functions

$$E_1 : (X \to R) \to \mathscr{P}X$$

$$E_2 : (Y \to R) \to \mathscr{P}Y$$

Recall that a subgame perfect equilibrium of this game is a strategy profile

$$(\sigma, \tau) : X \times (X \to Y)$$

such that

$$\sigma \in E_1(\mathscr{U}^q_{\langle\rangle}(\sigma, \tau))$$

and for all $x : X$,

$$\tau x \in E_2(\mathscr{U}^q_{\langle x\rangle}(\sigma, \tau))$$

where

$$\mathscr{U}^q_{\langle\rangle}(\sigma, \tau)x = q(x, \tau x)$$

and

$$\mathscr{U}^q_{\langle x\rangle}(\sigma, \tau)y = q(x, y)$$

A Nash equilibrium, on the other hand, weakens the second condition to only be required for $x = \sigma$.

Let $\mathcal{D}_1 : I \to X \otimes R^*$ be the decision in $\mathbf{Game}(\mathbf{Set})$ defined by the response function

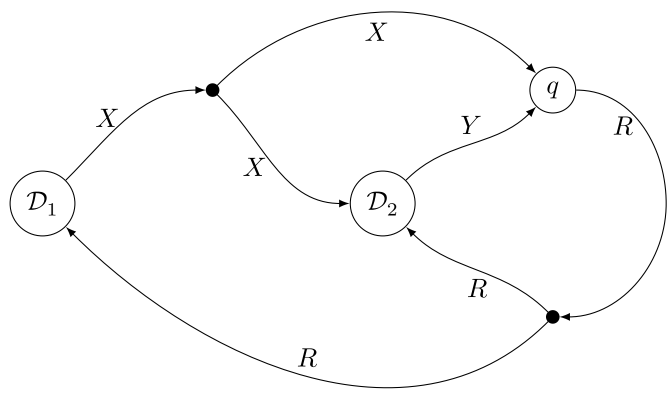
$$\mathbf{R}_{\mathcal{D}_1}(*, k) = E_1 k$$

and let $\mathcal{D}_2 : X \to Y \otimes R^*$ be the decision defined by

$$\mathbf{R}_{\mathcal{D}_2}(x, k) = \{\tau' : X \to Y \mid \tau' x \in E_2 k\}$$

These are instances of the construction in §2.1.7 of a decision from a multi-valued selection function.

Let $\mathcal{G} : I \to I$ be the closed game in $\mathbf{Game}_N(\mathbf{Set})$ denoted by the string diagram



or the algebraic expression

$$\mathcal{G} = \tau_R \circ_N (q \otimes \Delta_R^*) \circ_N (X \otimes \mathcal{D}_2 \otimes R^*) \circ_N (\Delta_X \otimes R^*) \circ_N \mathcal{D}_1$$

The set of strategy profiles for this game is $\Sigma_{\mathcal{G}} = X \times (X \to Y)$, and the fixpoints of $\mathbf{B}_{\mathcal{G}} : \Sigma_{\mathcal{G}} \to \mathscr{P}\Sigma_{\mathcal{G}}$ are precisely the Nash equilibria.

On the other hand, if we use SP-composition rather than N-composition then $\mathbf{B}_{\mathcal{G}}$ is precisely the best response function in §1.3.5, whose fixpoints are the subgame perfect equilibria. However, because $\mathbf{Game}_{SP}(\mathbf{Set})$ is only premonoidal the string diagram language is not well-defined, and we can only use the algebraic expression.

This example can also be extended in a graphically intuitive way to games of imperfect information, in which the second player can only observe some function of the first player's choice. The construction will be sketched in §3.2.7, although full proofs will not be given.

2.3.11 Coordination and differentiation games

As our final introductory example, we will return to the coordinating and differentiating behaviour of §1.2.9. As in the voting game introduced in §1.2.7, we will let X be finite and consider the outcomes to also be $R = X$. We will consider a pair of decisions

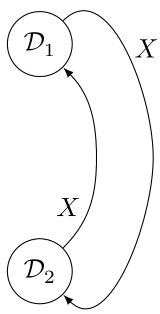
$$\mathcal{D}_{\text{fix}}, \mathcal{D}_{\text{nonfix}} : I \to X \otimes X^*$$

in $\mathbf{Game}_N(\mathbf{Set})$, defined by the response functions given exactly by the multi-valued selection functions E_{fix} and E_{nonfix}, so

$$\mathbf{B}_{\mathcal{D}_{\text{fix}}}(*, k)\sigma = E_{\text{fix}} k$$

$$\mathbf{B}_{\mathcal{D}_{\text{nonfix}}}(*, k)\sigma = E_{\text{nonfix}} k$$

Consider the string diagram



where $\mathcal{D}_1$ and $\mathcal{D}_2$ are chosen from $\mathcal{D}_{\text{fix}}$ and $\mathcal{D}_{\text{nonfix}}$. One way to write this algebraically is

$$\mathcal{G} = (\varepsilon_X \otimes \varepsilon_X) \circ (s_{X,X} \otimes X^* \otimes X^*) \circ (\mathcal{D}_1 \otimes \mathcal{D}_2) : I \to I$$

The type of strategy profiles is $\Sigma_\mathcal{G} = X \times X$. The best response function $\mathbf{B}_\mathcal{G} : X \times X \to \mathscr{P}(X \times X)$ is given by

$$\mathbf{B}_\mathcal{G}(\sigma, \tau) = \mathbf{B}_{\mathcal{D}_1}(*, k_1)\sigma \times \mathbf{B}_{\mathcal{D}_2}(*, k_2)\tau$$

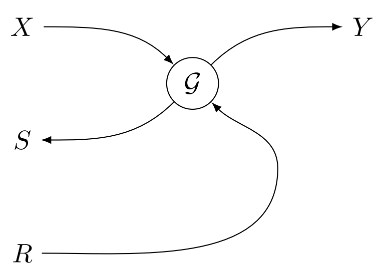
where $k_1 x = \tau$ and $k_2 x = \sigma$ are constant functions. Since $\sigma' \in \mathbf{B}_{\mathcal{D}_\text{fix}}(*, k_1)\sigma$ iff $\sigma' = \tau$, and $\tau' \in \mathbf{B}_{\mathcal{D}_\text{fix}}(*, k_2)\tau$ iff $\tau' = \sigma$, if we take $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_\text{fix}$ then (σ, τ) is an equilibrium of $\mathcal{G}$ iff $\sigma = \tau$. Similarly, if we take $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_\text{nonfix}$ then (σ, τ) is an equilibrium of $\mathcal{G}$ iff $\sigma \neq \tau$. Thus we again obtain coordinating and differentiating behaviour.

If we take $\mathcal{D}_1 = \mathcal{D}_\text{fix}$ and $\mathcal{D}_2 = \mathcal{D}_\text{nonfix}$ then $\mathcal{G}$ has no equilibria, because for (σ, τ) to be an equilibrium we must have both $\sigma = \tau$ and $\sigma \neq \tau$. It would be interesting to consider variants of $\mathcal{D}_\text{fix}$ and $\mathcal{D}_\text{nonfix}$ in $\mathbf{Game}_N(\mathbf{SRel})$, because this game should intuitively have a mixed strategy Nash equilibrium, but this is not free because we cannot take the mixed extension of a selection function (see §1.2.6).

This example demonstrates that open games can model non-classical preferences described by selection functions. It is interesting because the game has no outcome function in a very literal sense, but instead the string diagram visualises the information flow where each player directly reasons about the move of the other. However this issue is orthogonal to compositionality, and we will not discuss it further.

2.3.12 Designing for compositionality

We now return to theoretical considerations, having gained some intuition about the possible examples. The fact that the counit ε_X is not dualisable has some interesting consequences for information flow in $\mathbf{Game}_N(\mathcal{C})$. A game of type $\mathcal{G} : X \otimes S^* \to Y \otimes R^*$ can be thought of as accepting a value of type X from the past, and a value of type R from the future (see §2.1.4). If we have a value of type R in the past (for example, because an agent's utility is determined entirely by events in the past) then we can use the counit to bend the string around, as in



To put it another way: *a value known in the past will still be known in the future.*

On the other hand, if we only have a value of type X in the future, we cannot use it (unlike in a compact closed category) because an agent cannot directly observe the future, but only reason about it (which is the purpose of R). This 'causality' point of view suggests a heuristic explanation of why $\mathbf{Game}_N(\mathcal{C})$ has the structure that it does, namely that $-^*$ behaves in some ways like a duality, but is not a functor. Indeed, if $\mathbf{Game}_N(\mathcal{C})$ were compact closed it would be straightforward to construct a paradoxical situation analogous to the grandfather paradox, in which a strategy negates the value chosen by itself. Categories with more structure typically get around this problem using a 'failure' state such as the zero vector, the empty relation or the nonterminating computation.

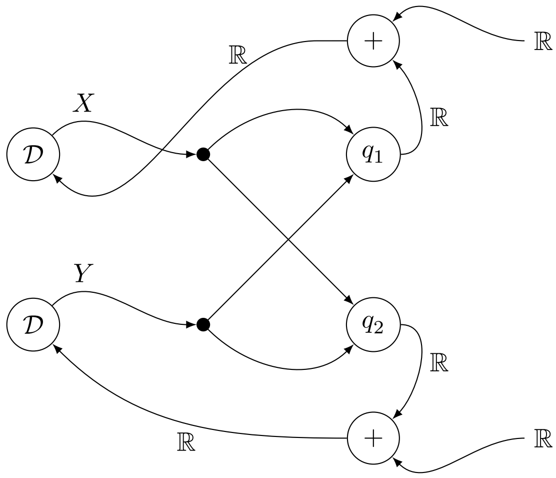
This will have a major impact on *design for compositionality*, which may become an important research topic in applied compositional game theory. In §3.1 and §3.2 we will show that certain known, simple classes of classical games can be faithfully represented as closed games. The purpose of this, as described in §0.5, is to show formally that compositional game theory is indeed game theory. However, a practical use of compositional game theory will likely look quite different.

In §2.3.9 we represented a bimatrix game as a closed game $\mathcal{G} : I \to I$. Since the game is closed, there is no nontrivial way in which it can interact with its environment, and consequently there is no nontrivial way to compose it with other games. This is unsurprising, because the point was to produce a faithful model of a classical bimatrix game, and classical bimatrix games do not interact with their environment, that is to say, they cannot be composed.

The question of how a bimatrix game should be represented in order to allow compositionality is a nontrivial modelling problem: we want to model the bimatrix game as an open game so that it could be reused in later, more complicated problems. The simpler half of this problem is what to take for Y and S, the types of values flowing from the game to its environment: a reasonable choice is $Y = X' \otimes Y'$, where X' and Y' are the types of choices made by the two players, and $S = I$, because there is no useful sense in which the bimatrix game can generate coutility.

The type X, given suitable features of the underlying category $\mathcal{C}$, could vary over all nonempty finite sets, with the game being parametrically polymorphic in X. The decisions, as defined in §2.1.8, do indeed seem to be parametrically polymorphic in X. The non-compositional example in §2.3.9 is recovered by setting $X = I$.

The difficult case is R. The problem is to determine how the utility of two concrete players should be affected by unknown events in the future. The simplest way to deal with this is to take $R = \mathbb{R} \otimes \mathbb{R}$ and use the well-understood ubiquity of real-valued utility as a 'universal representation', receiving an additional utility from the future that is simply added to the utility for each player from the game itself. The string diagram corresponding to this description is



One notable drawback of this is that the preferences of the players, which determine the appropriate utilities, become distributed throughout the string diagram rather than being localised. A possible alternative approach would be to have a game that is parametric in both a type R and in a rational preference relation on R. In any particular instantiation, the type and preference relation could be chosen depending on the modelling problem at hand.

Chapter 3

Game theory via open games

3.1 Normal form games

3.1.1 Discussion

In chapter 2 of this thesis we have built up an abstract theory, and this will be continued in §3.3 when we study solvability of games. The central argument being laid out (see the conclusion for a self-contained version) is that this approach to game theory is *better* than classical approaches, at least insofar as it is more scalable. However a vital piece of the argument is missing: other than the language used in the informal explanations, and the examples without proof at the end of §2.3, we have not demonstrated that open games have any connection *at all* with game theory.

The purpose of this chapter is to show that our theory does agree with classical game theory in certain situations, specifically normal form games with pure and mixed Nash equilibria, and certain extensive form games with subgame perfect equilibria. Given a classical game theoretic situation, we should be able to draw a string diagram that *looks like* the information flow in the target situation, and which moreover denotes *the same* game. In order to discuss sameness of games, without even having a common framework, we will use the approach discussed in §1.2.4: two games will be considered the same if they have the same sets of strategies Σ, and the same best response functions $\mathbf{B} : \Sigma \to \mathscr{P}\Sigma$. As we saw in §2.1.6, an abstract scalar in $\mathbf{Game}_N(\mathbf{Set})$ or $\mathbf{Game}_N(\mathbf{SRel})$ is defined precisely by Σ and $\mathbf{B}$ of this type. Therefore our aim is, given some existing notion of game, to construct an abstract scalar whose strategies and best responses are the same.

Game theory is a large subject, with entire areas devoted to various extensions and special cases of the basic definitions. For this reason there cannot be (and nor should there be) a single theorem that subsumes classical game theory into our new framework. For now, we are going to focus on some simple and common special cases. In this section, we begin with normal form finite games and pure strategies, as introduced in §1.2, and we will prove that every such game can be translated into a string diagram, whose strategies and Nash equilibria are the same. For additional simplicity we will work only with classical games (see §1.2.5), in order that this section can be understood only in terms of classical game theory, such as is introduced in [LBS08], without requiring knowledge of selection functions. The proofs in this section and §3.2 generalise immediately to arbitrary selection functions, using the translation from multi-valued selection functions to decisions in §2.1.7.

Beginning with §3.1.6, we will also show the same result for normal form games and mixed strategies, which is the setting in which Nash proved his famous existence theorem [Nas50].

3.1.2 Tensor products of decisions

Consider a finite normal form classical game, in the sense of §1.2.5. We have $N \geq 1$ players, where the ith player makes a choice from the finite set X_i, and receives a real-valued utility, which she aims to maximise. Since the choices in normal form games take place simultaneously, no player observes any history. Since we are intending to model pure strategies, which involve no side effects, we will work in the category $\mathbf{Game}_N(\mathbf{Set})$. The choice of the ith player will therefore be modelled by the utility-maximising decision

$$\mathcal{D}_i : I \to X_i \otimes \mathbb{R}^*$$

which is given as the first example in §2.1.8. For completeness, this is defined by the data

- $\Sigma_{\mathcal{D}_i} = X_i$
- $\mathbf{P}_{\mathcal{D}_i} \sigma * = \sigma$
- $\mathbf{C}_{\mathcal{D}_i} \sigma(x, u) = *$
- $\mathbf{B}_{\mathcal{D}_i} k\sigma = \{\sigma' : X_i \mid k\sigma' \geq kx' \text{ for all } x' : X_i\}$

The purpose of this section is to study the tensor product of finitely many such decisions.

We will work by induction on N, the number of players. Define a game

$$\mathcal{G}_N : I \to \bigotimes_{i=1}^N X_i \otimes (\mathbb{R}^*)^{\otimes N}$$

where $(\mathbb{R}^*)^{\otimes N}$ denotes the N-fold tensor $\mathbb{R}^* \otimes \cdots \otimes \mathbb{R}^*$, by

$$\mathcal{G}_N = \bigotimes_{i=1}^N \mathcal{D}_i$$

More formally, we recursively define $\mathcal{G}_1 = \mathcal{D}_1$, and $\mathcal{G}_{N+1} = \mathcal{G}_N \otimes \mathcal{D}_{N+1}$.

Our task is to find a closed form description of $\mathcal{G}_N$. It is simple to see that the set of strategy profiles is a cartesian product

$$\Sigma_{\mathcal{G}_N} = \prod_{i=1}^N X_i$$

The play function, whose type is isomorphic to

$$\mathbf{P}_{\mathcal{G}_N} : \prod_{i=1}^N X_i \to \prod_{i=1}^N X_i$$

is given by the identity function, because in the inductive case we have

$$\mathrm{id}_{\prod_{i=1}^N X_i} \times \mathrm{id}_{X_{N+1}} = \mathrm{id}_{\prod_{i=1}^{N+1} X_i}$$

Notice that this is equal to the play function $\mathbf{P}$ for normal form games defined in §1.2.2. The coplay function has type isomorphic to

$$\mathbf{C}_{\mathcal{G}_N} : \prod_{i=1}^N X_i \times \mathbb{R}^n \to 1$$

and so is uniquely defined.

3.1.3 Best response for a tensor of decisions

The best response function has type isomorphic to

$$\mathbf{B}_{\mathcal{G}_N} : \left(\prod_{i=1}^N X_i \to \mathbb{R}^N \right) \to \prod_{i=1}^N X_i \to \mathscr{P} \prod_{i=1}^N X_i$$

and is given by

$$\mathbf{B}_{\mathcal{G}_N} k\sigma = \left\{ \sigma' : \prod_{i=1}^N X_i \;\middle|\; k_i(\sigma'_i, \sigma_{-i}) \geq k_i(x', \sigma_{-i}) \right.$$
$$\left. \text{for all } 1 \leq i \leq N, x' : X_i \right\}$$

where $k_i = \pi_i \circ k$, and the notation (σ_i, σ'_{-i}) was introduced in §1.2.2. We will prove this by induction on N, the number of players.

This is the key step to connect our games with classical normal form games, because it has exactly the same form as the best response function in §1.2.5, and can be stated equivalently as saying that $B_{\mathcal{G}_N} k\sigma$ is the set of best responses to σ in the N-player normal form game whose outcome function is k.

We begin with the case $N = 1$. Since $\mathcal{G}_1 = \mathcal{D}_1$ is a decision, by definition we directly have

$$\mathbf{B}_{\mathcal{G}_1} k\sigma = \{\sigma' : X_1 \mid k\sigma' \geq kx'_1 \text{ for all } x'_1 : X_1\}$$

To see that this is of the required form, simply note that for $N = 1$ we have $(x_1, \sigma_{-1}) = x_1$, and that the projection $(-)_1$ is the identity.

For the inductive step, by definition of the tensor product,

$$\begin{aligned} \mathbf{B}_{\mathcal{G}_{N+1}} k\sigma &= \mathbf{B}_{\mathcal{G}_N \otimes \mathcal{D}_{N+1}} k(\sigma_{-(N+1)}, \sigma_{N+1}) \\ &= \mathbf{B}_{\mathcal{G}_N} k_{\otimes \sigma_{N+1}} \sigma_{-(N+1)} \times \mathbf{B}_{\mathcal{D}_{N+1}} k_{\sigma_{-(N+1)} \otimes} \sigma_{N+1} \end{aligned}$$

We introduce here the less cluttered notation $k_{\otimes\sigma_{N+1}}$ and $k_{\sigma_{-(N+1)}\otimes}$ as shorthand for $k_{\otimes\sigma_{N+1}(*)}$ and $k_{\sigma_{-(N+1)}(*)\otimes}$ (see §2.2.7), because all histories are trivial for decisions in normal form games.

Before we proceed further, we must explicitly calculate $k_{\otimes\sigma_N}$ and $k_{\sigma_{-(N+1)}\otimes}$. The original continuation is

$$k : \prod_{i=1}^{N+1} X_i \to \mathbb{R}^{N+1}$$

which we will write in the isomorphic form

$$k : \prod_{i=1}^{N} X_i \times X_{N+1} \to \mathbb{R}^N \times \mathbb{R}$$

The left continuation $k_{\otimes\sigma_{N+1}} : \prod_{i=1}^N X_i \to \mathbb{R}^N$ is, by the definition in §2.2.7, given by

$$\prod_{i=1}^N X_i \xrightarrow{\cong} \prod_{i=1}^N X_i \times 1 \xrightarrow{\prod_{i=1}^N X_i \times \mathbf{P}_{\mathcal{D}_{N+1}} \sigma_{N+1}} \prod_{i=1}^N X_i \times X_{N+1}$$

$$\xrightarrow{k} \mathbb{R}^N \times \mathbb{R} \xrightarrow{\pi_{-(N+1)}} \mathbb{R}^N$$

Since $\mathcal{D}_{N+1}$ is a decision in $\mathbf{Game}_N(\mathbf{Set})$ its play function is

$$\mathbf{P}_{\mathcal{D}_{N+1}} \sigma_{N+1} * = \sigma_{N+1}$$

so this is simply

$$k_{\otimes\sigma_{N+1}} \sigma' = k_{-(N+1)}(\sigma', \sigma_{N+1})$$

Similarly the right continuation $k_{\sigma_{-(N+1)}\otimes} : X_{N+1} \to \mathbb{R}$ by definition is

$$X_{N+1} \xrightarrow{\cong} 1 \times X_{N+1} \xrightarrow{\mathbf{P}_{\mathcal{G}_N} \sigma_{-(N+1)} * \times X_{N+1}} \prod_{i=1}^N X_i \times X_{N+1}$$

$$\xrightarrow{k} \mathbb{R}^N \times \mathbb{R} \xrightarrow{\pi_{N+1}} \mathbb{R}$$

Since we have already checked in the previous section that

$$\mathbf{P}_{\mathcal{G}_N} \sigma_{-(N+1)} * = \sigma_{-(N+1)}$$

this is explicitly

$$k_{\sigma_{-(N+1)}\otimes} \sigma' = k_{N+1}(\sigma_{-(N+1)}, \sigma')$$

3.1.4 Best response for a tensor of decisions, continued

We can now expand

$$\mathbf{B}_{\mathcal{G}_{N+1}} k \sigma = \mathbf{B}_{\mathcal{G}_N} k_{\otimes\sigma_{N+1}} \sigma_{-(N+1)} \times \mathbf{B}_{\mathcal{D}_{N+1}} k_{\sigma_{-(N+1)}\otimes} \sigma_{N+1}$$

The first term $\mathbf{B}_{\mathcal{G}_N} k_{\otimes\sigma_{N+1}} \sigma_{-(N+1)}$, by the inductive hypothesis, is the set of $\sigma' : \prod_{i=1}^N X_i$ such that for all $1 \leq i \leq N$ and $x_i' : X_i$,

$$(k_{\otimes\sigma_{N+1}})_i(\sigma_i', (\sigma_{-(N+1)})_{-i}) \geq (k_{\otimes\sigma_{N+1}})_i(x_i', (\sigma_{-(N+1)})_{-i})$$

Expanding the definition of $k_{\otimes\sigma_{N+1}}$, this is

$$(k_{-(N+1)})_i(\sigma'_i, (\sigma_{-(N+1)})_i, \sigma_{N+1}) \geq (k_{-(N+1)})_i(x'_i, (\sigma_{-(N+1)})_i, \sigma_{N+1})$$

Noticing that the composition of the projections

$$\prod_{i=1}^{N+1} X_i \xrightarrow{\pi_{-(N+1)}} \prod_{i=1}^{N} X_i \xrightarrow{\pi_i} X_i$$

is itself just a projection, this can be simplified to

$$k_i(\sigma'_i, \sigma_{-i}, \sigma_{N+1}) \geq k_i(x'_i, \sigma_{-i}, \sigma_{N+1})$$

This notation is the obvious extension of that introduced in §1.2.2, namely

$$(x'_i, \sigma_{-i}, \sigma_{N+1})_j = \begin{cases} x'_i & \text{if } j = i \\ \sigma_i & \text{if } 1 \leq j \leq N \text{ and } j \neq i \\ \sigma_{N+1} & \text{if } j = N+1 \end{cases}$$

Similarly, the second term $\mathbf{B}_{\mathcal{D}_{N+1}} k_{\sigma_{-(N+1)} \otimes} \sigma_{N+1}$ is by definition the set of $\sigma' : X_{N+1}$ such that for all $x'_{N+1} : X_{N+1}$,

$$k_{\sigma_{-(N+1)}\otimes} \sigma' \geq k_{\sigma_{-(N+1)}\otimes} x'_{N+1}$$

and this expands to

$$k_{N+1}(\sigma_{-(N+1)}, \sigma') \geq k_{N+1}(\sigma_{-(N+1)}, x'_{N+1})$$

Putting these together, the cartesian product

$$\mathbf{B}_{\mathcal{G}_{N+1}} k\sigma = \mathbf{B}_{\mathcal{G}_N} k_{\otimes\sigma_{N+1}} \sigma_{-(N+1)} \times \mathbf{B}_{\mathcal{D}_{N+1}} k_{\sigma_{-(N+1)}\otimes} \sigma_{N+1}$$

is equal to the set of $\sigma' : \prod_{i=1}^{N+1} X_i$, such that both of these conditions hold. The first holds for all $i \leq N$, and we notice that the second is equal to the first for $i = N+1$. This means that the set $\mathbf{B}_{\mathcal{G}_{N+1}} k\sigma$ has exactly the form we required, and we are done.

3.1.5 The payoff functions

It may be that the game $\mathcal{G}_N$ we have constructed is the properly idiomatic representation of a normal form game as an open game, leaving the utility function as a continuation parameter in the best response function. However, if we have in mind some concrete utility function

$$q : \prod_{i=1}^{N} X_i \to \mathbb{R}^N$$

then we can use the result proved in §2.2.13, that the closed game[1]

$$\mathcal{H} = \varepsilon_{\mathbb{R}^{\otimes N}} \circ_N (q \otimes (\mathbb{R}^*)^{\otimes N}) \circ_N \mathcal{G}_N : I \to I$$

[1] Note that the subscript in $\circ_N$ is not related to the number of players N.

has the property that

$$\mathbf{B}_{\mathcal{G}_N} q\sigma = \mathbf{B}_{\mathcal{H}} * \sigma$$

Therefore $\Sigma_{\mathcal{H}} = \Sigma_{\mathcal{G}_N}$ and $\mathbf{B}_{\mathcal{H}} : \Sigma_{\mathcal{G}_N} \to \mathscr{P}\Sigma_{\mathcal{G}_N}$ specify the same strategies and best responses for $\mathcal{H}$ as for the classical normal form game we began with. As discussed in §3.1.1, this is precisely the aim we began with.

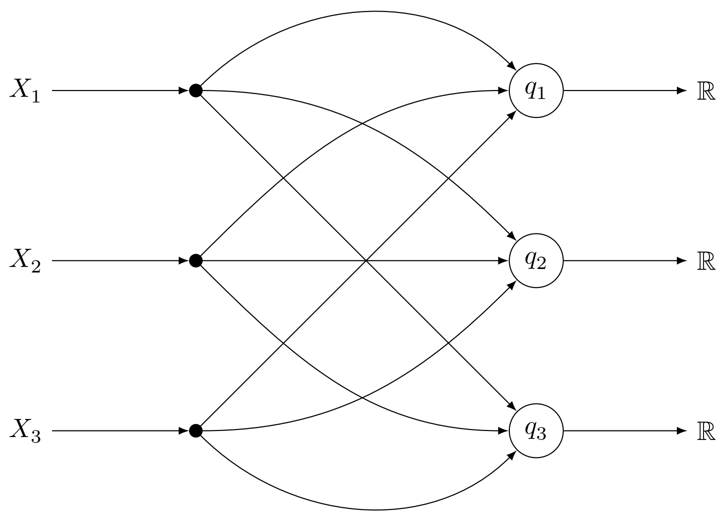
Because computations are a monoidal embedding, we can go further and decompose q, so that the information flow inside q is visible in the string diagram. Typically, q is constructed by beginning with a utility function

$$q_i : \prod_{i=1}^{N} X_i \to \mathbb{R}$$

for each player i. The tensor product of these, considered as covariant computations, is

$$\bigotimes_{i=1}^{N} q_i : \bigotimes_{i=1}^{N} \bigotimes_{j=1}^{N} X_j \to \mathbb{R}^{\otimes N}$$

Since each of the N utility functions requires access to all N choices, we need to copy each choice N times and then braid them into the right order. This is implemented by a copy followed by symmetries. For example, when $N = 3$ we use



In general, the copying is implemented by a covariant computation

$$\bigotimes_{i=1}^{N} X_i \to \bigotimes_{i=1}^{N} \bigotimes_{j=1}^{N} X_j$$

whose play function is

$$\mathbf{P}(x_1, \ldots, x_N) = (x_1, \ldots, x_N, \ldots, x_1, \ldots, x_N)$$

Of course, particular features such as common subexpressions can be used to simplify such a string diagram in many examples.

The example game in §2.3.9 has this form, but uses mixed strategies, which we will now develop. In the remainder of this section, and in §3.2, we will work with the open form of a game, leaving the outcome function implicit as a continuation parameter in the best response function.

3.1.6 Stochastic decisions

The result that normal form games can be faithfully represented as open games also holds if we change our solution concept from pure to mixed strategy Nash equilibrium. Mixed strategy Nash equilibria of classical games were introduced in §1.2.6. Following the example in §2.3.9 we now describe the general case, making use of the probability distribution monad $\mathscr{D}$ introduced in §2.1.3.

We will now work in the category $\mathbf{Game}_N(\mathbf{SRel})$, where $\mathbf{SRel}$ is the Kleisli category of $\mathscr{D}$. Following §3.1.2, we have a sequence of decisions

$$\mathcal{D}_i : I \to X_i \otimes \mathbb{R}^*$$

which are defined by the rationality function

$$\mathbf{R} : (X_i \to \mathscr{D}\mathbb{R}) \to \mathscr{P}(\mathscr{D}X_i)$$

given by

$$\mathbf{R}k = \{\sigma : \mathscr{D}X_i \mid \mathbb{E}[k^*\sigma] \geq \mathbb{E}[kx'] \text{ for all } x' : X_i\}$$

These are the expected utility maximising decisions introduced in §2.1.8. To recall, $\mathcal{D}_i$ has the set of strategies

$$\Sigma_{\mathcal{D}_i} = \hom_{\mathbf{SRel}}(I, X_i) = \mathscr{D}X_i$$

Its play function has type

$$\mathbf{P}_{\mathcal{D}_i} : \Sigma_{\mathcal{D}_i} \to \hom_{\mathbf{SRel}}(I, X_i) = \mathscr{D}X_i \to \mathscr{D}X_i$$

and is equal to the identity function, and its coplay function

$$\mathbf{C}_{\mathcal{D}_i} : \Sigma_{\mathcal{D}_i} \to \hom_{\mathbf{SRel}}(I \otimes \mathbb{R}, I) = \mathscr{D}X_i \to \mathbb{R} \to \mathscr{D}1$$

and is uniquely defined as the function onto $\mathscr{D}1 = 1$, the only probability distribution on a one-element set. Finally, its best response function is

$$\begin{aligned}\mathbf{B}_{\mathcal{D}_i} &: \hom_{\mathbf{SRel}}(I, I) \times \hom_{\mathbf{SRel}}(X_i, \mathbb{R}) \to \Sigma_{\mathcal{D}_i} \to \mathscr{P}\Sigma_{\mathcal{D}_i} \\ &= (X_i \to \mathscr{D}\mathbb{R}) \to \mathscr{D}X_i \to \mathscr{P}(\mathscr{D}X_i)\end{aligned}$$

given by

$$\mathbf{B}_{\mathcal{D}_i} k\sigma = \{\sigma' : \mathscr{D}X_i \mid \mathbb{E}[k^*\sigma] \geq \mathbb{E}[kx'] \text{ for all } x' : X_i\}$$

As in §3.1.2, we will define the game

$$\mathcal{G}_N : I \to \bigotimes_{i=1}^N X_i \otimes (\mathbb{R}^*)^{\otimes N}$$

by

$$\mathcal{G}_N = \bigotimes_{i=1}^N \mathcal{D}_i$$

By induction we get a closed form for the strategy set

$$\Sigma_{\mathcal{G}_N} = \prod_{i=1}^N \mathscr{D} X_i$$

and also a closed form for the play function

$$\mathbf{P}_{\mathcal{G}_N} : \prod_{i=1}^N \mathscr{D} X_i \to \mathscr{D} \prod_{i=1}^N X_i$$

given by the monoidal product of the monad $\mathscr{D}$, in other words,

$$\mathbb{P}\left[\mathbf{P}_{\mathcal{G}_N} \sigma = x\right] = \prod_{i=1}^N \mathbb{P}\left[\sigma_i = x_i\right]$$

The coplay function is uniquely defined by its type

$$\mathbf{C}_{\mathcal{G}_N} : \prod_{i=1}^N \mathscr{D} X_i \to \mathbb{R}^N \to \mathscr{D} 1$$

again because $\mathscr{D} 1 = 1$.

3.1.7 Best response for a tensor of stochastic decisions

We will prove by induction that the best response function

$$\mathbf{B}_{\mathcal{G}_N} : \left(\prod_{i=1}^N X_i \to \mathscr{D} \mathbb{R}^n \right) \to \prod_{i=1}^N \mathscr{D} X_i \to \mathscr{P} \prod_{i=1}^N \mathscr{D} X_i$$

has the closed form

$$\mathbf{B}_{\mathcal{G}_N} k \sigma = \left\{ \sigma' : \prod_{i=1}^N \mathscr{D} X_i \;\middle|\; \mathbb{E}[k_i(\sigma'_i, \sigma_{-i})] \geq \mathbb{E}[k_i(x'_i, \sigma_{-i})] \right.$$
$$\left. \text{for all } x'_i : X_i, 1 \leq i \leq N \right\}$$

This is exactly the best response function for games with mixed strategies given in §1.2.6. In the case $N = 1$, it is

$$\mathbf{B}_{\mathcal{G}_N} k \sigma = \{ \sigma' : \mathscr{D} X_1 \mid \mathbb{E}[k \sigma'] \geq \mathbb{E}[k x'_1] \text{ for all } x'_1 : X_1 \}$$

which holds by definition.

Note that from this point, we begin to abuse notation by leaving the Kleisli extension of $\mathscr{D}$ implicit. This is standard practice in mathematics, where for example an ordinary non-stochastic function f can be applied to a random variable α, resulting in a random variable $f \alpha$.

By the same reasoning as in §3.1.3, the condition

$$(\sigma'_{-(N+1)}, \sigma'_{N+1}) \in \mathbf{B}_{\mathcal{G}_{N+1}} k (\sigma_{-(N+1)}, \sigma_{N+1})$$

is equivalent to the pair of conditions

$$\sigma'_{-(N+1)} \in \mathbf{B}_{\mathcal{G}_N} k_{\otimes \sigma_{N+1}} \sigma_{-(N+1)}$$

and

$$\sigma'_{N+1} \in \mathbf{B}_{\mathcal{D}_{N+1}} k_{\sigma_{-(N+1)} \otimes} \sigma_{N+1}$$

The former, by the inductive hypothesis, is equivalent to the claim that for all $1 \leq i \leq N$ and $x'_i : X_i$,

$$\mathbb{E}[(k_{\otimes \sigma_{N+1}})_i((\sigma'_{-(N+1)})_i, (\sigma_{-(N+1)})_{-i})] \geq \mathbb{E}[(k_{\otimes \sigma_{N+1}})_i(x'_i, (\sigma_{-(N+1)})_{-i})]$$

The latter, by definition of $\mathcal{D}_{N+1}$, is equivalent to the claim that for all deviations $x'_{N+1} : X_{N+1}$,

$$\mathbb{E}[k_{\sigma_{-(N+1)} \otimes} \sigma'_{N+1}] \geq \mathbb{E}[k_{\sigma_{-(N+1)} \otimes} x'_{N+1}]$$

We must prove that for all $1 \leq i \leq N+1$ and $x'_i : X_i$,

$$\mathbb{E}[k_i(\sigma'_i, \sigma_{-i})] \geq \mathbb{E}[k_i(x'_i, \sigma_{-i})]$$

In order to obtain this from the inductive hypothesis, it remains to prove that

$$(k_{\otimes \sigma_{N+1}})_i(x'_i, (\sigma_{-(N+1)})_{-i}) = k_i(x'_i, \sigma_{-i})$$

for $1 \leq i \leq N$ and $x'_i : X_i$, and

$$k_{\sigma_{-(N+1)} \otimes} x'_{N+1} = k_{N+1}(\sigma_{-(N+1)}, x'_{N+1})$$

for $x'_{N+1} : X_{N+1}$. Continuing to abuse notation, for the former we can say that

$$\begin{aligned}(k_{\otimes \sigma_{N+1}})_i(x'_i, (\sigma_{-(N+1)})_{-i}) &= (k_{-(N+1)})_i(x'_i, (\sigma_{-(N+1)})_{-i}, \sigma_{N+1}) \\ &= k_i(x'_i, \sigma_{-i})\end{aligned}$$

and similarly,

$$k_{\sigma_{-(N+1)} \otimes} x'_{N+1} = k_{N+1}(\sigma_{-(N+1)}, x'_{N+1})$$

holds by definition. More carefully, both sides of the former are given by

$$\mathbb{P}\left[k_i(x'_i, \sigma_{-i}) = u_i\right] = \sum_{\substack{u_{-i} : \mathbb{R}^N \\ x_{-i} : \prod_{j \neq i} X_j}} \left(\prod_{j \neq i} \mathbb{P}\left[\sigma_j = x_j\right] \right) \cdot \mathbb{P}\left[k(x'_i, x_{-i}) = u\right]$$

and the latter by

$$\begin{aligned}&\mathbb{P}\left[k_{N+1}(\sigma_{-(N+1)}, x'_{N+1}) = u_{N+1}\right] \\ = &\sum_{\substack{u_{-(N+1)} : \mathbb{R}^N \\ x_{-(N+1)} : \prod_{j=1}^N X_j}} \left(\prod_{j=1}^N \mathbb{P}\left[\sigma_j = x_j\right] \right) \cdot \mathbb{P}\left[k(x_{-(N+1)}, x'_{N+1}) = u\right]\end{aligned}$$

3.2 Extensive form

3.2.1 Discussion

In §3.1 we showed that normal form games can be translated into abstract scalars in $\mathbf{Game}_N(\mathbf{Set})$ or $\mathbf{Game}_N(\mathbf{SRel})$ in a way that preserves the pure or mixed strategies and Nash equilibria, respectively. In this chapter, we will focus on translating extensive form games into abstract scalars in $\mathbf{Game}_{SP}(\mathbf{Set})$, while preserving pure strategies and subgame perfect equilibria. $\mathbf{Game}_{SP}(\mathbf{Set})$ is the category whose composition operation is SP-composition, defined in §2.2.4, which is proved in the appendix to be symmetric premonoidal.

We give a complete proof for the case of sequential games, that is, perfect information games in which the type of choices of a player may not depend on the values of previous choices (see §1.3.1 and §1.3.4). In §3.2.7 we additionally discuss the representation of imperfect information. Giving a general translation of an arbitrary extensive form game into an abstract scalar of $\mathbf{Game}_{SP}(\mathbf{Set})$ would be more work, and is left for the future.

We will see in §3.3.7 that using the same method to translate extensive form games with mixed strategies into $\mathbf{Game}_{SP}(\mathbf{SRel})$ *goes wrong*, in the sense that it gives an implausible solution concept, and that the most fundamental definitions in §2.1 and §2.2 will need to be changed as a result. This is why we restrict to pure strategies in this chapter.

This section is, technically speaking, the most difficult of this thesis, with §3.2.4 and §3.2.5 being particularly dense. Verifying these proofs is made more difficult by the fact that it is necessary for readability to leave many isomorphisms of sets implicit, such as when using the notation (x_i', x_{-i}) for unilateral deviation. An implementation in a proof assistant such as Coq would be useful, and will be even more important when scaling this style of proof beyond such simple types of games as are considered in this thesis.

3.2.2 Composition with perfect information

We will begin with a sequence of decisions

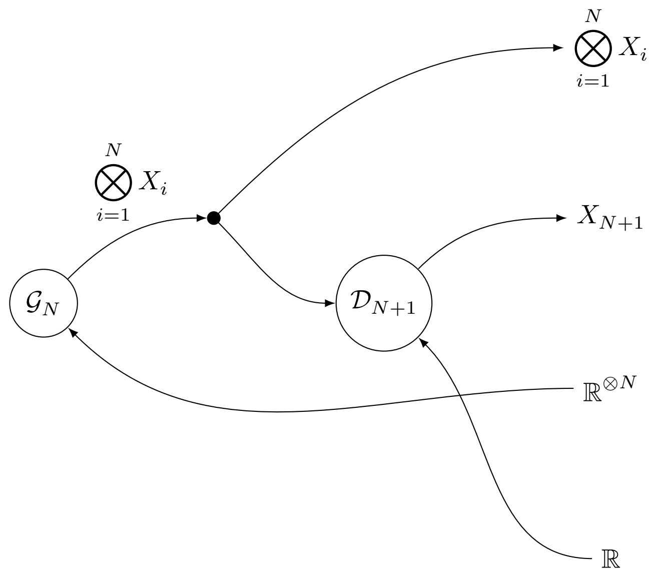
$$\mathcal{D}_i : \bigotimes_{j=1}^{i-1} X_j \to X_i \otimes \mathbb{R}^*$$

in $\mathbf{Game}_{SP}(\mathbf{Set})$. The ith decision represents a player who observes the first $i-1$ choices. Note that since the empty tensor is the identity, the first of these is $\mathcal{D}_1 : I \to X_1 \otimes \mathbb{R}^*$. We will define a game

$$\mathcal{G}_N : I \to \bigotimes_{i=1}^{N} X_i \otimes (\mathbb{R}^*)^{\otimes N}$$

which behaves like their perfect information composition, by induction on N. Like for simultaneous games, the base case is $\mathcal{G}_1 = \mathcal{D}_1$.

For the inductive step, we are given $\mathcal{G}_N$ and $\mathcal{D}_{N+1}$, and we must produce $\mathcal{G}_{N+1}$. In this game, $\mathcal{G}_N$ must play first, and then $\mathcal{D}_{N+1}$ must play observing the choices of $\mathcal{G}_N$. However, the choices of $\mathcal{G}_N$ must also be preserved in the output of $\mathcal{G}_{N+1}$, and hence must be copied in parallel with $\mathcal{D}_{N+1}$. The string diagram to keep in mind is



However, because $\mathbf{Game}_{SP}(\mathbf{Set})$ is not a monoidal category we do not actually have a string diagram language, and so the real definition is given by the algebraic term below.

We will translate this diagram into monoidal category notation in stages. The decision is

$$\mathcal{D}_{N+1} : \bigotimes_{i=1}^{N} X_i \to X_{N+1} \otimes \mathbb{R}^*$$

To add the string above it we tensor with the identity

$$\bigotimes_{i=1}^{N} X_i \otimes \mathcal{D}_{N+1} : \bigotimes_{i=1}^{N} X_i \otimes \bigotimes_{i=1}^{N} X_i \to \bigotimes_{i=1}^{N+1} X_i \otimes \mathbb{R}^*$$

The copying node is the computation

$$\Delta_{\prod_{i=1}^{N} X_i} : \bigotimes_{i=1}^{N} X_i \to \bigotimes_{i=1}^{N} X_i \otimes \bigotimes_{i=1}^{N} X_i$$

The top-right part of the diagram is therefore

$$\left(\bigotimes_{i=1}^{N} X_i \otimes \mathcal{D}_{N+1} \right) \circ_{SP} \Delta_{\prod_{i=1}^{N} X_i} : \bigotimes_{i=1}^{N} X_i \to \bigotimes_{i=1}^{N+1} X_i \otimes \mathbb{R}^*$$

Before the lower-right crossing, we can tensor with the lower $\mathbb{R}^{\otimes N}$-labelled contravariant string to get

$$\left(\left(\bigotimes_{i=1}^{N} X_i \otimes \mathcal{D}_{N+1} \right) \circ_{SP} \Delta_{\prod_{i=1}^{N} X_i} \right) \otimes (\mathbb{R}^*)^{\otimes N}$$

$$: \bigotimes_{i=1}^{N} X_i \otimes (\mathbb{R}^*)^{\otimes N} \to \bigotimes_{i=1}^{N+1} X_i \otimes \mathbb{R}^* \otimes (\mathbb{R}^*)^{\otimes N}$$

The crossing is the dual of the computation

$$\sigma_{\mathbb{R}^N,\mathbb{R}} : \mathbb{R}^{\otimes(N+1)} \to \mathbb{R} \otimes \mathbb{R}^{\otimes N}$$

and we must also tensor with the identity above it, to give

$$\bigotimes_{i=1}^{N+1} X_i \otimes \sigma^*_{\mathbb{R}^N,\mathbb{R}} : \bigotimes_{i=1}^{N+1} X_i \otimes \mathbb{R}^* \otimes (\mathbb{R}^*)^{\otimes N} \to \bigotimes_{i=1}^{N+1} X_i \otimes (\mathbb{R}^*)^{\otimes(N+1)}$$

Finally, we can give the denotation of the entire string diagram as

$$\mathcal{G}_{N+1} = \mathcal{H}_{N+1} \circ_{SP} \mathcal{G}_N : I \to \bigotimes_{i=1}^{N+1} X_i \otimes (\mathbb{R}^*)^{\otimes(N+1)}$$

where

$$\mathcal{H} : \bigotimes_{i=1}^{N} X_i \otimes (\mathbb{R}^*)^{\otimes N} \to \bigotimes_{i=1}^{N+1} X_i \otimes (\mathbb{R}^*)^{\otimes(N+1)}$$

is given by

$$\mathcal{H} = \left(\bigotimes_{i=1}^{N+1} X_i \otimes \sigma^*_{\mathbb{R}^N,\mathbb{R}} \right) \circ_{SP} \left(\left(\left(\bigotimes_{i=1}^{N} X_i \otimes \mathcal{D}_{N+1} \right) \circ_{SP} \Delta_{\prod_{i=1}^{N} X_i} \right) \otimes (\mathbb{R}^*)^{\otimes N} \right)$$

3.2.3 Building the composition

Since the strategy sets of the decisions are

$$\Sigma_{\mathcal{D}_i} = \prod_{j=1}^{i-1} X_j \to X_i$$

it can easily be seen that the strategy sets of the $\mathcal{G}_i$ are isomorphic to the cartesian product of these,

$$\Sigma_{\mathcal{G}_N} = \prod_{i=1}^{N} \left(\prod_{j=1}^{i-1} X_j \to X_i \right)$$

The coplay function of $\mathcal{G}_N$ has type

$$\mathbf{C}_{\mathcal{G}_N} : \prod_{i=1}^{N} \left(\prod_{j=1}^{i-1} X_j \to X_i \right) \to 1 \to 1$$

and hence is uniquely defined.

The play function has type

$$\mathbf{P}_{\mathcal{G}_N} : \prod_{i=1}^{N} \left(\prod_{j=1}^{i-1} X_j \to X_i \right) \to \prod_{i=1}^{N} X_i$$

We will prove that $\mathbf{P}_{\mathcal{G}_N} = \mathbf{P}$ is the play function for sequential games introduced in §1.3.4. Explicitly, we will prove that

$$(\mathbf{P}_{\mathcal{G}_N}\sigma)_i = \sigma_i((\mathbf{P}_{\mathcal{G}_N}\sigma)_1, \ldots, (\mathbf{P}_{\mathcal{G}_N}\sigma)_{i-1})$$

for all $1 \leq i \leq N$. This should be compared to the defining equation of the play function in §1.3.4. In the base case $\mathcal{G}_1 = \mathcal{D}_1$, the play function is by definition

$$(\mathbf{P}_{\mathcal{D}_1}\sigma)_1 = \mathbf{P}_{\mathcal{D}_1}\sigma = \sigma = \sigma_1$$

which satisfies the equation.

For the inductive step, we need to show that

$$\mathbf{P}_{\mathcal{G}_{N+1}}\sigma = (x, \sigma_{N+1}x)$$

where $x = \mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)}$. Since

$$\mathbf{P}_{\mathcal{G}_{N+1}}\sigma = \mathbf{P}_{\mathcal{H}_{N+1}}\sigma_{N+1} \circ \mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)}$$

it suffices to prove that

$$\mathbf{P}_{\mathcal{H}_{N+1}}\sigma_{N+1}x = (x, \sigma_{N+1}x)$$

To begin with, we note that

$$\mathbf{P}_{\bigotimes_{i=1}^N X_i \otimes \sigma^*_{\mathbb{R}^N,\mathbb{R}}} = \mathrm{id}_{\prod_{i=1}^N X_i}$$

and so

$$\begin{aligned}
\mathbf{P}_{\mathcal{H}_{N+1}}\sigma_{N+1} &= \mathbf{P}_{\left((\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}) \circ_{SP} \Delta_{\prod_{i=1}^N X_i}\right) \otimes (\mathbb{R}^*)^{\otimes N}}\sigma_{N+1} \\
&= \mathbf{P}_{(\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}) \circ_{SP} \Delta_{\prod_{i=1}^N X_i}}\sigma_{N+1}
\end{aligned}$$

Then

$$\begin{aligned}
\mathbf{P}_{(\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}) \circ_{SP} \Delta_{\prod_{i=1}^N X_i}}\sigma_{N+1}x &= \mathbf{P}_{\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}}\sigma_{N+1}\left(\mathbf{P}_{\Delta_{\prod_{i=1}^N X_i}} x\right) \\
&= \mathbf{P}_{\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}}\sigma_{N+1}\left(\Delta_{\prod_{i=1}^N X_i} x\right) \\
&= \mathbf{P}_{\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}}(x, x) \\
&= \left(\mathbf{P}_{\bigotimes_{i=1}^N X_i} x, \mathbf{P}_{\mathcal{D}_{N+1}}\sigma_{N+1}x\right) \\
&= (x, \sigma_{N+1}x)
\end{aligned}$$

Having proved this, we have

$$(\mathbf{P}_{\mathcal{G}_{N+1}}\sigma)_i = \begin{cases} (\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)})_i & \text{if } i \leq N \\ \sigma_{N+1}(\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)}) & \text{if } i = N+1 \end{cases}$$

For $i \leq N$, the inductive hypothesis gives

$$\begin{aligned}
(\mathbf{P}_{\mathcal{G}_{N+1}}\sigma)_i &= (\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)})_i \\
&= (\sigma_{-(N+1)})_i((\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)})_1, \ldots, (\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)})_{i-1}) \\
&= \sigma_i((\mathbf{P}_{\mathcal{G}_{N+1}}\sigma)_1, \ldots, (\mathbf{P}_{\mathcal{G}_{N+1}}\sigma)_{i-1})
\end{aligned}$$

and for $i = N + 1$,

$$\begin{aligned}(\mathbf{P}_{\mathcal{G}_{N+1}}\sigma)_{N+1} &= \sigma_{N+1}(\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)}) \\ &= \sigma_{N+1}((\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)})_1, \ldots, (\mathbf{P}_{\mathcal{G}_N}\sigma_{-(N+1)})_N) \\ &= \sigma_{N+1}((\mathbf{P}_{\mathcal{G}_{N+1}}\sigma)_1, \ldots, (\mathbf{P}_{\mathcal{G}_{N+1}}\sigma)_N)\end{aligned}$$

3.2.4 Best response for the composition

We will now prove that our best response function

$$\mathbf{B}_{\mathcal{G}_N} : \left(\prod_{i=1}^{N} X_i \to \mathbb{R}^N\right) \to \Sigma_{\mathcal{G}_N} \to \mathscr{P}\Sigma_{\mathcal{G}_N}$$

where

$$\Sigma_{\mathcal{G}_N} = \prod_{i=1}^{N}\left(\prod_{j=1}^{i-1} X_j \to X_i\right)$$

is such that $\mathbf{B}_{\mathcal{G}_N} k\sigma$ is the set of all $\sigma' : \Sigma_{\mathcal{G}_N}$ such that, for all players $1 \leq i \leq N$ and all partial plays $x = x_1, \ldots, x_{i-1}$ of length $i - 1$, and all possible deviations $x_i : X_i$, we have

$$(\mathscr{U}_x^k \sigma(\sigma_i' x))_i \geq (\mathscr{U}_x^k \sigma x_i)_i$$

where $\mathscr{U}_x^k$ is the unilateral deviation operator for sequential games, defined in §1.3.5.

The proof is again by induction on N. In the base case we have $N = 1$ and we need only check $i = 1$ with the empty partial play and some deviation $x_1 : X_1$. Then

$$(\mathscr{U}^k \sigma\sigma')_1 = k\sigma'$$

and

$$(\mathscr{U}^k \sigma x_1)_1 = kx_1$$

and we are done, because by definition $\mathbf{B}_{\mathcal{G}_1} = \mathbf{B}_{\mathcal{D}_1}$ takes $k : X_1 \to \mathbb{R}$ and $\sigma : X_1$ to the set of all $\sigma' : X_1$ such that

$$k\sigma' \geq kx'$$

for all $x' : X_1$

Next, by the definition of composition,

$$\begin{aligned}\mathbf{B}_{\mathcal{G}_{N+1}} k\sigma = \mathbf{B}_{\mathcal{G}_N} &k_{\sigma_{N+1}\circ}\sigma_{-(N+1)} \\ &\times \bigcap_{\sigma' : \Sigma_{\mathcal{G}}} \mathbf{B}_{\mathcal{H}_{N+1}}(\mathbf{P}_{\mathcal{G}_N}\sigma', k)\sigma_{N+1}\end{aligned}$$

For $\mathbf{B}_{\mathcal{G}_{N+1}} k\sigma$ to contain some σ', it must be that

$$\sigma'_{-(N+1)} \in \mathbf{B}_{\mathcal{G}_N} k_{\sigma_{N+1}\circ}\sigma_{-(N+1)}$$

and for all $\sigma''_{-(N+1)}$,

$$\sigma'_{N+1} \in \mathbf{B}_{\mathcal{H}_{N+1}}(\mathbf{P}_{\mathcal{G}_N}\sigma''_{-(N+1)}, k)\sigma_{N+1}$$

The first condition, by the inductive hypothesis, is equivalent to the claim that for all $1 \leq i \leq N$, partial plays $x = x_1, \ldots, x_{i-1} : \prod_{j<i} X_j$ and devations $x_i : X_i$,

$$(\mathscr{U}_x^{k_{\sigma_{N+1}} \circ} \sigma_{-(N+1)}((\sigma'_{-(N+1)})_i x))_i \geq (\mathscr{U}_x^{k_{\sigma_{N+1}} \circ} \sigma_{-(N+1)} x_i)_i$$

We claim that there is an equality of continuations

$$\mathscr{U}_x^{k_{\sigma_{N+1}} \circ} \sigma_{-(N+1)} = \pi_{-(N+1)} \circ \mathscr{U}_x^k \sigma : X_i \to \mathbb{R}^N$$

which immediately gives us the inductive hypothesis for $N+1$ when $1 \leq i \leq N$. To show this, by definition

$$k_{\sigma_{N+1}} \circ : \prod_{j=1}^N X_j \to \mathbb{R}^N$$

is given by

$$k_{\sigma_{N+1}} \circ x = k_{-(N+1)}(x, \sigma_{N+1} x)$$

Now we have

$$\begin{aligned}\mathscr{U}_x^{k_{\sigma_{N+1}} \circ} \sigma_{-(N+1)} x_i &= k_{\sigma_{N+1}} \circ (\nu_{x,x_i}^{\sigma_{-(N+1)}}) \\ &= k_{-(N+1)}(\nu_{x,x_i}^{\sigma_{-(N+1)}}, \sigma_{N+1}(\nu_{x,x_i}^{\sigma_{-(N+1)}}))\end{aligned}$$

where ν is defined in §1.3.5, because by definition

$$\mathscr{U}_x^k \sigma x_i = k(\nu_{x,x_i}^\sigma)$$

Therefore it suffices to prove that

$$\nu_{x,x_i}^\sigma = (\nu_{x,x_i}^{\sigma_{-(N+1)}}, \sigma_{N+1}(\nu_{x,x_i}^{\sigma_{-(N+1)}}))$$

By definition

$$(\nu_{x,x_i}^\sigma)_{N+1} = \sigma_{N+1}((\nu_{x,x_i}^\sigma)_1, \ldots, (\nu_{x,x_i}^\sigma)_N)$$

so it suffices to prove that

$$(\nu_{x,x_i}^\sigma)_j = (\nu_{x,x_i}^{\sigma_{-(N+1)}})_j$$

for $1 \leq j \leq N$. This is easily proved by strong induction on j.

3.2.5 Best response for the composition, continued

We now return to the second condition for

$$\sigma' \in \mathbf{B}_{\mathcal{G}_{N+1}} k\sigma$$

namely that for all $\sigma''_{-(N+1)} : \Sigma_{\mathcal{G}_N}$,

$$\sigma'_{N+1} \in \mathbf{B}_{\mathcal{H}_{N+1}}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)}, k)\sigma_{N+1}$$

We claim that this is equivalent to

$$\sigma'_{N+1} \in \mathbf{B}_{\mathcal{D}_{N+1}}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)}, k')\sigma_{N+1}$$

where $k' : X_{N+1} \to \mathbb{R}$ is given by

$$k' x_{N+1} = k_{N+1}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)}, x_{N+1})$$

We will prove the slightly more general claim where $\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)}$ is replaced with a general play $x_{-(N+1)}$. To begin, since $\bigotimes_{i=1}^{N+1} X_i \otimes \sigma^*_{\mathbb{R}^N, \mathbb{R}}$ is strategically trivial,

$$\begin{aligned} &\mathbf{B}_{\mathcal{H}_{N+1}}(x_{-(N+1)}, k)\sigma_{N+1} \\ = &\mathbf{B}_{\left((\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}) \circ_{SP} \Delta_{\prod_{i=1}^N X_i} \right) \otimes (\mathbb{R}^*)^{\otimes N}}(x_{-(N+1)}, k_{*\circ})\sigma_{N+1} \end{aligned}$$

where

$$k_{*\circ} : \prod_{i=1}^{N+1} X_i \to \mathbb{R}^{N+1}$$

is the extended continuation defined in §2.2.3, given explicitly by

$$\begin{aligned} k_{*\circ} x &= \mathbf{C}_{\bigotimes_{i=1}^{N+1} X_i \otimes \sigma^*_{\mathbb{R}^N, \mathbb{R}}}(x, k(\mathbf{P}_{\bigotimes_{i=1}^{N+1} X_i \otimes \sigma^*_{\mathbb{R}^N, \mathbb{R}}} x)) \\ &= \sigma_{\mathbb{R}^N, \mathbb{R}}(k(\mathrm{id}_{\prod_{i=1}^N X_i} x)) \\ &= (k_{N+1} x, k_{-(N+1)} x) \end{aligned}$$

Next, because the identity on $(\mathbb{R}^*)^{\otimes N}$ is strategically trivial,

$$\begin{aligned} &\mathbf{B}_{\left((\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}) \circ_{SP} \Delta_{\prod_{i=1}^N X_i} \right) \otimes (\mathbb{R}^*)^{\otimes N}}(x_{-(N+1)}, k_{*\circ})\sigma_{N+1} \\ = &\mathbf{B}_{(\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}) \circ_{SP} \Delta_{\prod_{i=1}^N X_i}}(x_{-(N+1)}, (k_{*\circ})_1)\sigma_{N+1} \end{aligned}$$

where

$$(k_{*\circ})_1 x = k_{N+1} x$$

Therefore

$$\begin{aligned} &\mathbf{B}_{(\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}) \circ_{SP} \Delta_{\prod_{i=1}^N X_i}}(x_{-(N+1)}, k_{N+1})\sigma_{N+1} \\ = &\mathbf{B}_{\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}}(\mathbf{P}_{\Delta_{\prod_{i=1}^N X_i}} x_{-(N+1)}, k_{N+1})\sigma_{N+1} \\ = &\mathbf{B}_{\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}}(\Delta_{\prod_{i=1}^N X_i} x_{-(N+1)}, k_{N+1})\sigma_{N+1} \\ = &\mathbf{B}_{\bigotimes_{i=1}^N X_i \otimes \mathcal{D}_{N+1}}((x_{-(N+1)}, x_{-(N+1)}), k_{N+1})\sigma_{N+1} \\ = &\mathbf{B}_{\mathcal{D}_{N+1}}(x_{-(N+1)}, (k_{N+1})_{*(x_{-(N+1)})\otimes})\sigma_{N+1} \end{aligned}$$

where

$$\begin{aligned} (k_{N+1})_{*(x_{-(N+1)})\otimes} x_{N+1} &= k_{N+1}(\mathbf{P}_{\bigotimes_{i=1}^N X_i} x_{-(N+1)}, x_{N+1}) \\ &= k_{N+1}(x_{-(N+1)}, x_{N+1}) \\ &= k' x_{N+1} \end{aligned}$$

for the definition of k' given earlier.

Now, by definition of $\mathbf{B}_{\mathcal{D}_{N+1}}$, the condition

$$\sigma'_{N+1} \in \mathbf{B}_{\mathcal{D}_{N+1}}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)}, k') \sigma_{N+1}$$

is equivalent to

$$k'(\sigma'_{N+1}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)})) \geq k' x_{N+1}$$

for all $x_{N+1} : X_{N+1}$, which is

$$k_{N+1}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)}, \sigma'_{N+1}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)})) \geq k_{N+1}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)}, x_{N+1})$$

It remains to show that this, universally quantified over $\sigma''_{-(N+1)} : \Sigma_{\mathcal{G}_N}$, implies the remaining case of the inductive hypothesis for $N+1$, namely that for all partial plays $x = x_1, \ldots, x_N : \prod_{j=1}^N X_j$ and deviations $x_{N+1} : X_{N+1}$,

$$(\mathscr{U}_x^k \sigma(\sigma'_{N+1} x))_{N+1} \geq (\mathscr{U}_x^k \sigma x_{N+1})_{N+1}$$

Notice that

$$(\mathscr{U}_x^k \sigma x_{N+1})_{N+1} = k_{N+1}(\nu^\sigma_{x, x_{N+1}}) = k_{N+1}(x, x_{N+1}) = k' x_{N+1}$$

so we must show

$$k_{N+1}(x, \sigma'_{N+1} x) \geq k_{N+1}(x, x_{N+1})$$

for all partial plays x.

Given such an x, we take the strategy profile

$$(\sigma''_{-(N+1)})_i x' = x_i$$

which ignores its observation, and simply plays the move from x. We claim that $\mathbf{P}_{\mathcal{G}_N} \sigma'' = x$. Using the characterisation of $\mathbf{P}_{\mathcal{G}_N}$ in §3.2.3, for each $1 \leq i \leq N$ we have

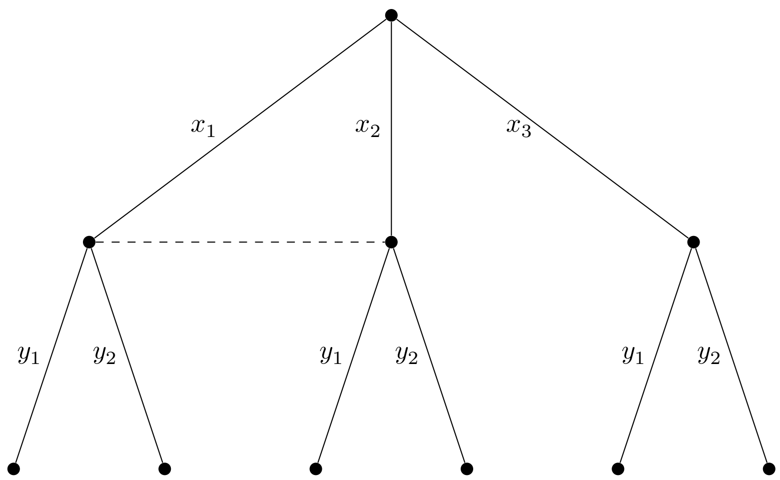
$$\begin{aligned}(\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)})_i &= (\sigma''_{-(N+1)})_i((\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)})_1, \ldots, (\mathbf{P}_{\mathcal{G}_N} \sigma''_{-(N+1)})_{i-1}) \\ &= x_i\end{aligned}$$

This completes the proof.

3.2.6 Information sets

Information sets are used in [vNM44] to give a general theory of imperfect information, including games that are intermediate between sequential and simultaneous. In general, a player may be able to observe partial information about another player's earlier move. That is, they may be able to distinguish some pairs of moves, but not others.

For simplicity, we will begin with a concrete two-player game. Suppose the set of moves for the first player is $X = \{x_1, x_2, x_3\}$ and the set of moves for the second player is $Y = \{y_1, y_2\}$. The game tree for this game is



An **information set** for the second player is simply a partitioning of X, that is to say, an equivalence relation on X. The intention is that two elements of X that are in the same equivalence class cannot be distinguished by the second player, whereas elements in different equivalence classes can be. In this example, the dashed line is the usual notation for specifying that two nodes are in the same equivalence class. Thus, our partitioning of X is $X/{\sim} = \{\{x_1, x_2\}, \{x_3\}\}$.

For simplicity, we will focus on pure strategies, although sequential equilibrium [LBS08] is a more common solution concept for games with nontrivial information sets (and see §3.3.7). The informal condition that the second player cannot distinguish between x_1 and x_2 can be interpreted formally as a restriction on the allowed strategies, namely that every strategy for player 2 must be compatible with the equivalence relation. Thus, a function $\sigma_2 : X \to Y$ is an allowed strategy for player 2 iff $\sigma_2 x_1 = \sigma_2 x_2$. This is equivalent to saying that σ_2 is actually a function $\sigma_2 : X/{\sim} \to Y$. The set of strategies of this game, therefore, is

$$\Sigma = X \times (X/{\sim} \to Y)$$

The general definition of a subgame perfect equilibrium, including with nontrivial information sets, is a strategy that induces a Nash equilibrium on every subgame. For this two-stage game, this is the same as a fixpoint of the best response function $\mathbf{B} : \Sigma \to \mathscr{P}\Sigma$, where $\mathbf{B}(\sigma_1, \sigma_2)$ is the set of all (σ'_1, σ'_2) satisfying the two conditions

$$q_1(\sigma'_1, \sigma_2[\sigma'_1]) \geq q_1(x', \sigma_2[x'])$$

for all $x' : X$, and

$$q_2(x', \sigma'_2[x']) \geq q_2(x', y')$$

for all $x' : X$ and $y' : Y$. Equivalently, we can give the two conditions as

$$\sigma'_1 = \arg\max_{x' : X} q_1(x', \sigma_2[x'])$$

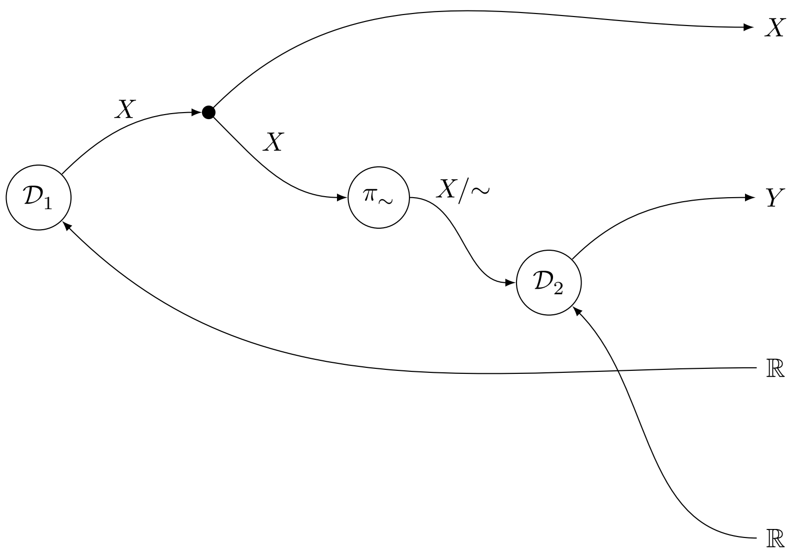
and for all $x' : X$,

$$\sigma'_2[x'] = \arg\max_{y' : Y} q_2(x', y')$$

3.2.7 Imperfect information via open games

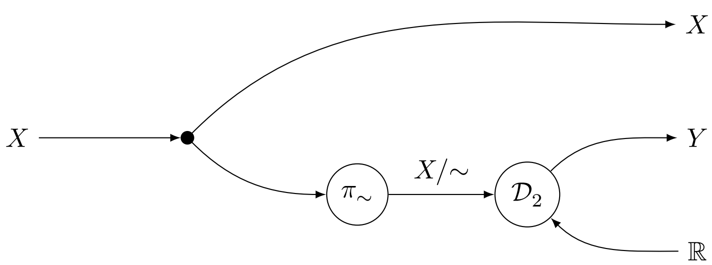
We will end this section by taking a more general two player game involving imperfect information, and showing how it can be formalised in $\mathbf{Game}_{SP}(\mathbf{Set})$. The sets of moves are X and Y, and we have an equivalence relation $\sim$ on X. A strategy profile is a pair $(\sigma_1, \sigma_2) : X \times (X/{\sim} \to Y)$. The definition of subgame perfect equilibrium is as before.

Intuitively, we use the string diagram



where $\pi_\sim : X \to X/{\sim}$ is the projection onto information sets, $\pi_\sim x = [x]$. However, as before, this cannot be formalised because $\mathbf{Game}_{SP}(\mathbf{Set})$ is not monoidal. This should be compared to the string diagrams in §2.3.10 and §3.2.2, which denote perfect information games in which $\sim$ is the identity relation, and so $\pi_\sim$ is the identity on X. Notice how we can explicitly see that the information flowing from $\mathcal{D}_1$ to $\mathcal{D}_2$ passes through the projection $\pi_\sim$, where it is partially hidden.

Since most of the analysis is the same as for the perfect information case, we will focus on the sub-diagram



which is given algebraically by

$$\mathcal{G} = (X \otimes (\mathcal{D}_2 \circ_{SP} \pi_\sim)) \circ_{SP} \Delta_X : X \to X \otimes Y \otimes \mathbb{R}^*$$

The set of strategy profiles is $\Sigma_\mathcal{G} = X/{\sim} \to Y$, with play function

$$\mathbf{P}_\mathcal{G} \sigma x = (x, \sigma[x])$$

The best response function

$$\mathbf{B}_\mathcal{G} : X \times (X \times Y \to \mathbb{R}) \to \Sigma_\mathcal{G} \to \mathscr{P}\Sigma_\mathcal{G}$$

is calculated as

$$\begin{aligned}
\mathbf{B}_\mathcal{G}(h, k)\sigma &= \mathbf{B}_{X \otimes (\mathcal{D}_2 \circ_{SP} \pi_\sim)}((h, h), k)\sigma \\
&= \mathbf{B}_{\mathcal{D}_2 \circ_{SP} \pi_\sim}(h, k')\sigma \\
&= \mathbf{B}_{\mathcal{D}_2}([h], k')\sigma
\end{aligned}$$

where $k'y = k(h, y)$.

Since $\mathbf{B}_{\mathcal{D}_2}$ is a utility-maximising decision, this is

$$\mathbf{B}_\mathcal{G}(h, k)\sigma = \{\sigma' : X/{\sim} \to Y \mid k(h, \sigma[h]) \geq k(h, y') \text{ for all } y' : Y\}$$

and the condition can be equivalently written

$$\sigma[h] = \arg\max_{y' : Y} k(h, y')$$

This corresponds exactly to the second condition for the best response function in §3.2.6.

3.3 Solvable games

3.3.1 Discussion

In this section we will introduce a suitable solution concept for open games, called solvability. Although the best response function of a game contains all of the information needed to define equilibria, it does not seem to be directly usable as a solution concept, because there is nothing that corresponds directly to an 'equilibrium of an open game'. A solution of a game, on the other hand, is defined in terms of the best response function and can be directly read as an analogue of Nash or subgame perfect equilibria for open games.

The solvability of a game should be seen as a *witness for an existence theorem*. This is because a solution of an open game implies the existence of equilibria for a class of closed games obtained by varying its continuation, that is, for a class of games whose outcome function varies. For example, as mentioned in §3.3.6 there is a single open game whose solvability is equivalent to the Nash existence theorem for bimatrix games (of fixed size).

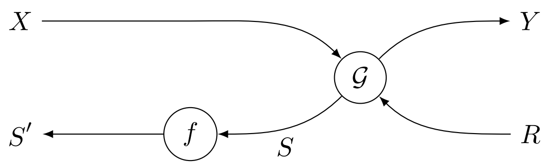
The ultimate aim is to show that solvability is respected by the categorical operations used to construct games, which will allow us to prove existence theorems by structural induction, and compute equilibria recursively. In §3.3.4 we will show that when applied to categorical composition this yields something akin to *backward induction* in classical game theory, and thereby connect solutions with selection functions. In §3.3.6 and §3.3.7, we will discuss how solvability behaves with respect to tensor products, which is more subtle and is ongoing work.

3.3.2 The definition of solvability

Let $\mathcal{G} : X \otimes S^* \to Y \otimes R^*$ be a game in $\mathbf{Game}_N(\mathcal{C})$ or $\mathbf{Game}_{SP}(\mathcal{C})$. We will call $\mathcal{G}$ **solvable** if, for every continuation $k : \hom_{\mathcal{C}}(Y, R)$, there exists a strategy $\sigma : \Sigma_{\mathcal{G}}$, such that for all histories $h : \hom_{\mathcal{C}}(I, X)$, we have $\sigma \in \mathbf{B}_{\mathcal{G}}(h, k)\sigma$.

A **solution** of $\mathcal{G}$ is a function $s : \hom_{\mathcal{C}}(Y, R) \to \Sigma_{\mathcal{G}}$ such that sk is a fixpoint of $\mathbf{B}_{\mathcal{G}}(h, k)$ for all contexts $(h, k) : \hom_{\mathcal{C}}(I, X) \times \hom_{\mathcal{C}}(Y, R)$. By the axiom of choice, a game is solvable iff it has a solution.

A first technical issue with solvability is that it is not 'aware' of coutility, because the definition of solvability of $\mathcal{G}$ does not refer to S or $\mathbf{C}_{\mathcal{G}}$. Consequently, the solutions of



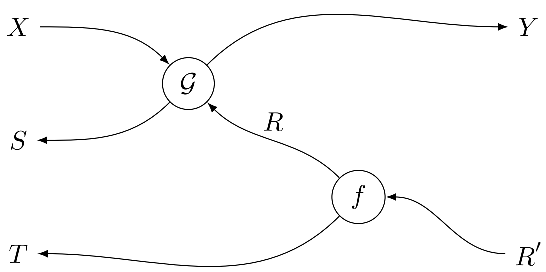
are independent of the computation $f : S \to S'$.

Notice that all strategically trivial games (§2.2.11) are solvable. Take a strategically trivial game $\mathcal{G} : X \otimes S^* \to Y \otimes R^*$, so we have $\Sigma_{\mathcal{G}} = 1$, and for every $h : \hom_{\mathcal{C}}(I, X)$ and $k : \hom_{\mathcal{C}}(Y, R)$ we have

$$* \in \mathbf{B}_{\mathcal{G}}(h, k)*$$

Then the unique function $s : \hom_{\mathcal{C}}(Y, R) \to \Sigma_{\mathcal{G}}$ is trivially a solution.

The following is a simple but useful way to obtain new solvable games from old. Let $\mathcal{G} : X \otimes S^* \to Y \otimes R^*$ be solvable, and let $f : R' \to R \otimes T$ be a computation. Then



is also solvable. When solutions are connected with selection functions in §3.3.5, this is analogous to modifying the outcome type of a selection function (§1.1.8). Algebraically, the game denoted by this string diagram is

$$(Y \otimes f^*) \circ_{SP} (\mathcal{G} \otimes T^*) : X \otimes S^* \otimes T^* \to Y \otimes R'^*$$

(and the same holds if $\circ_{SP}$ is replaced with $\circ_N$.) It is obtained from $\mathcal{G}$ by composing and tensoring with strategically trivial games, and so

$$\Sigma_{(Y \otimes f^*) \circ_{SP} (\mathcal{G} \otimes T^*)} = \Sigma_{\mathcal{G}}$$

Its best response function is

$$\mathbf{B}_{(Y\otimes f^*)\circ_{SP}(\mathcal{G}\otimes T^*)} : \hom_{\mathcal{C}}(I,X)\times\hom_{\mathcal{C}}(Y,R') \to \Sigma_{\mathcal{G}} \to \mathscr{P}\Sigma_{\mathcal{G}}$$

given by

$$\mathbf{B}_{(Y\otimes f^*)\circ_{SP}(\mathcal{G}\otimes T^*)}(h,k)\sigma = \mathbf{B}_{\mathcal{G}}(h,k')\sigma$$

where

$$k' : Y \xrightarrow{k} R' \xrightarrow{f} R\otimes T \xrightarrow{\pi_1} R$$

Let $k : \hom_{\mathcal{C}}(Y,R)$ be a continuation for $(Y\otimes f^*)\circ_{SP}(\mathcal{G}\otimes T^*)$. By the solvability of $\mathcal{G}$ with the continuation k', we have $\sigma : \Sigma_{\mathcal{G}}$ such that, for all $h : \hom_{\mathcal{C}}(I,X)$,

$$\sigma \in \mathbf{B}_{\mathcal{G}}(h,k')\sigma$$

Thus $\sigma \in \mathbf{B}_{(Y\otimes f^*)\circ_{SP}(\mathcal{G}\otimes T^*)}(h,k)\sigma$, as required.

3.3.3 Solvable decisions

We will now provide our first nontrivial examples of a solvable games, namely the decisions over **Set** defined in §2.1.8 that maximise with respect to a rational preference relation $\succeq$ on a set R. These are

$$\mathcal{D} : X \to Y \otimes R^*$$

defined by the rationality function

$$\mathbf{R} : X \times (Y \to R) \to \mathscr{P}(X \to Y)$$

given by

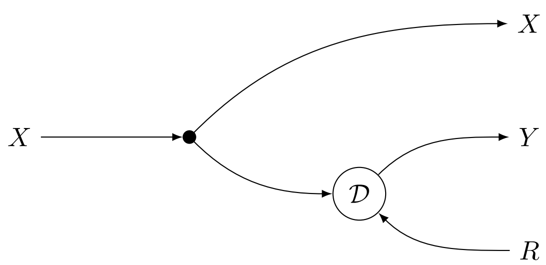
$$\mathbf{R}(x,k) = \{\sigma : X \to Y \mid k(\sigma x) \succeq ky \text{ for all } y : Y\}$$

We will prove that this decision is solvable, if Y is finite.

Let $k : Y \to R$ be a continuation. Then since Y is finite and $\succeq$ is a rational preference relation there is some $y : Y$ for which ky is maximal. Pick the constant strategy $\sigma : X \to Y$ given by $\sigma x = y$. Then for all $h : X$ we have $\sigma \in \mathbf{B}_{\mathcal{D}}(h,k)\sigma$ because for all $y' : Y$ we have

$$k(\sigma x) = ky \succeq ky'$$

In $\mathbf{Game}_{SP}(\mathbf{Set})$ we can prove a stronger ad-hoc result, namely that



is also solvable. Algebraically, this game is $(X \otimes \mathcal{D}) \circ_{SP} \Delta_X$, which is obtained from $\mathcal{D}$ by tensoring and composing with strategically trivial games. Thus, the strategy set is

$$\Sigma_{(X \otimes \mathcal{D}) \circ_{SP} \Delta_X} = \Sigma_{\mathcal{D}} = X \to Y$$

Its best response function, by similar reasoning as in §3.2.4, is

$$\mathbf{B}_{(X \otimes \mathcal{D}) \circ_{SP} \Delta_X}(h, k)\sigma = \{\sigma' : X \to Y \mid \sigma' \in \mathbf{B}_{\mathcal{D}}(h, k')\sigma\}$$

where $k' : Y \to R$ is given by $k'y = k(h, y)$. Explicitly, this is

$$\mathbf{B}_{(X \otimes \mathcal{D}) \circ_{SP} \Delta_X}(h, k)\sigma = \{\sigma' : X \to Y \mid k(h, \sigma' h) \succeq k(h, y) \text{ for all } y : Y\}$$

Now let $k : X \times Y \to R$ be a continuation for $(X \otimes \mathcal{D}) \circ_{SP} \Delta_X$. Define a strategy $\sigma : X \to Y$ by

$$\sigma x = \arg\max_{y:Y} k(x, y)$$

where $\arg\max$ picks some y that attains the $\succeq$-maximum. Then for every history $h : X$ we have

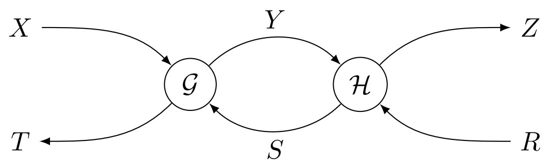
$$k(h, \sigma x) \succeq k(h, y)$$

by construction, and hence $\sigma \in \mathbf{B}_{(X \otimes \mathcal{D}) \circ_{SP} \Delta_X}(h, k)\sigma$.

3.3.4 Backward induction for open games

We will now prove that solvable games are closed under SP-composition. The proof of this result amounts to *backward induction*, just as does the product of selection functions (§1.3.6). This is the main piece of evidence thus far that solvable games are mathematically natural.

Suppose we have games



with solutions

$$s_{\mathcal{G}} : \hom_{\mathcal{C}}(Y, S) \to \Sigma_{\mathcal{G}}$$

$$s_{\mathcal{H}} : \hom_{\mathcal{C}}(Z, R) \to \Sigma_{\mathcal{H}}$$

We define $s_{\mathcal{H} \circ_{SP} \mathcal{G}} : \hom_{\mathcal{C}}(Z, R) \to \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$ by

$$s_{\mathcal{H} \circ_{SP} \mathcal{G}} k = (s_{\mathcal{G}} k_{s_{\mathcal{H}} k \circ}, s_{\mathcal{H}} k)$$

where $k_{s_{\mathcal{H}} k \circ} : \hom_{\mathcal{C}}(Y, S)$ is the extended continuation defined in §2.2.3. We must prove that $s_{\mathcal{H} \circ_{SP} \mathcal{G}}$ is a solution of $\mathcal{H} \circ_{SP} \mathcal{G}$.

Let $(h, k) : \hom_{\mathcal{C}}(I, X) \times \hom_{\mathcal{C}}(Z, R)$ be a context for $\mathcal{H} \circ_{SP} \mathcal{G}$. We must prove that

$$s_{\mathcal{H} \circ_{SP} \mathcal{G}} k \in \mathbf{B}_{\mathcal{H} \circ_{SP} \mathcal{G}}(h, k)(s_{\mathcal{H} \circ_{SP} \mathcal{G}} k)$$

This unwinds to the two conditions

$$s_{\mathcal{G}} k_{s_{\mathcal{H}} k \circ} \in \mathbf{B}_{\mathcal{G}}(h, k_{s_{\mathcal{H}} k \circ})(s_{\mathcal{G}} k_{s_{\mathcal{H}} k \circ})$$

and

$$s_{\mathcal{H}}k \in \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma \circ h, k)(s_{\mathcal{H}}k)$$

for all $\sigma : \Sigma_{\mathcal{G}}$. Both of these follow immediately from the fact that $s_{\mathcal{G}}$ and $s_{\mathcal{H}}$ are solutions of $\mathcal{G}$ and $\mathcal{H}$ respectively.

3.3.5 Selection functions and open games

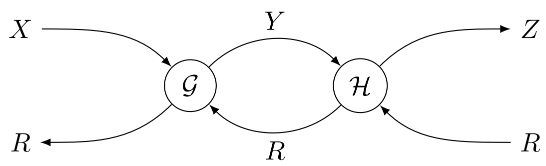
Given a solution $s : \hom_{\mathcal{C}}(Y, R) \to \Sigma_{\mathcal{G}}$ of a game $\mathcal{G} : X \otimes S^* \to Y \otimes R^*$, we have an associated selection function

$$\mathbf{P}_{\mathcal{G}} \circ s : \hom_{\mathcal{C}}(Y, R) \to \hom_{\mathcal{C}}(X, Y)$$

which is a morphism in $\hom_{\mathscr{J}_R\mathcal{C}}(X, Y)$, where $\mathscr{J}_R\mathcal{C}$ is the category of selection functions defined in §1.3.2. We will call ε a selection function for $\mathcal{G}$ if it has the form $\varepsilon = \mathbf{P}_{\mathcal{G}} \circ s$ for a solution s of $\mathcal{G}$.

We will now for the first time relate the composition in the category of selection functions, or equivalently the Kleisli extension of the selection monad, with game theory. Unfortunately, this connection does not behave well with respect to coplay; it remains to be seen whether or not it is a useful idea.

Suppose we have games



in $\mathbf{Game}_{SP}(\mathcal{C})$, whose coplay functions are the identity on R. We will prove that if ε is a selection function for $\mathcal{G}$ and δ is a selection function for $\mathcal{H}$, then $\delta \circ \varepsilon$ is a selection function for $\mathcal{H} \circ_{SP} \mathcal{G}$, where the composition $\delta \circ \varepsilon$ is taken in the category $\mathscr{J}_R\mathcal{C}$.

By hypothesis, we have a solution $s_{\mathcal{G}} : \hom_{\mathcal{C}}(Y, R) \to \Sigma_{\mathcal{G}}$ of $\mathcal{G}$, and a solution $s_{\mathcal{H}} : \hom_{\mathcal{C}}(Z, R) \to \Sigma_{\mathcal{H}}$ of $\mathcal{H}$, such that $\varepsilon = \mathbf{P}_{\mathcal{G}} \circ s_{\mathcal{G}}$ and $\delta = \mathbf{P}_{\mathcal{H}} \circ s_{\mathcal{H}}$. The composition of these is

$$(\delta \circ \varepsilon)k = \delta k \circ \varepsilon(k \circ \delta k) = \mathbf{P}_{\mathcal{H}}(s_{\mathcal{H}}k) \circ \mathbf{P}_{\mathcal{G}}(s_{\mathcal{G}}(k \circ \mathbf{P}_{\mathcal{H}}(s_{\mathcal{H}}k)))$$

The solution $s_{\mathcal{H} \circ_{SP} \mathcal{G}}$ of $\mathcal{H} \circ_{SP} \mathcal{G}$ is defined by

$$s_{\mathcal{H} \circ_{SP} \mathcal{G}}k = (s_{\mathcal{G}}k_{s_{\mathcal{H}}k\circ}, s_{\mathcal{H}}k)$$

The selection function associated to this solution is

$$\mathbf{P}_{\mathcal{H} \circ_{SP} \mathcal{G}}(s_{\mathcal{H} \circ_{SP} \mathcal{G}}k) = \mathbf{P}_{\mathcal{H}}(s_{\mathcal{H}}k) \circ \mathbf{P}_{\mathcal{G}}(s_{\mathcal{G}}(k_{s_{\mathcal{H}}k\circ}))$$

We get that $\delta \circ \varepsilon$ is precisely this, and so it is a selection function for $\mathcal{H} \circ_{SP} \mathcal{G}$. Here we need to use the restricted form of the coplay function of $\mathcal{H}$ in order to get that

$$k_{s_{\mathcal{H}}k\circ} = k \circ \mathbf{P}_{\mathcal{H}}(s_{\mathcal{H}}k)$$

rather than (in pure strategies)

$$k_{s_{\mathcal{H}}k\circ}y = \mathbf{C}_{\mathcal{H}}(s_{\mathcal{H}}k)(y, k(\mathbf{P}_{\mathcal{H}}(s_{\mathcal{H}}k)y))$$

This provides us with a high level way to think about solvability. Given an object R of $\mathcal{C}$, we consider the subcategory $\mathbf{Game}_{SP}(\mathcal{C}, R)$ whose objects are of the form $X \otimes R^*$ for X an object of $\mathcal{C}$, and whose morphisms are solvable games whose coplay function is constant. Then a (compatible) choice of solution $s_\mathcal{G}$ for each game $\mathcal{G}$ defines a functor $F : \mathbf{Game}_{SP}(\mathcal{C}, R) \to \mathscr{J}_R \mathcal{C}$ whose action on objects is $F(X \otimes R^*) = X$ and on morphisms is $F\mathcal{G} = \mathbf{P}_\mathcal{G} \circ s_\mathcal{G}$. Although it is possible to do a certain amount of game theory in $\mathbf{Game}_{SP}(\mathcal{C}, R)$ when R is suitably chosen, it is unlikely that this way of thinking has enough benefit to offset the restriction of not having coplay available, which limits compositionality in practice. One way in which this is useful, however, is to understand the relationship between selection functions and open games.

3.3.6 Tensor does not preserve solvability

In general, the class of solvable games is not closed under tensor. This is a high level way of saying that simultaneous games do not have equilibria for every solution concept, and in particular they do not have pure strategy Nash equilibria.

We will give a counterexample using matching pennies, which is the simplest game with no pure strategy equilibrium. In this game two players each simultaneously choose H ('heads') or T ('tails'). If both players make the same choice then player 1 wins, and if the choices are different player 2 wins. This game has no pure strategy Nash equilibrium, because when playing a strategy in which both choices are the same player 2 has incentive to deviate, and when playing a strategy in which the choices are different player 1 has incentive to deviate. (This game has a unique mixed strategy equilibrium, in which both players mix with equal probability.)

We will formalise this game by working in $\mathbf{Game}SP(\mathbf{Set})$. Let $X = \{H, T\}$, and let R be the poset $\{0 \prec 1\}$, where 1 indicates winning and 0 indicates losing. The maximising decision

$$\mathcal{D} : I \to X \otimes R^*$$

was proved to be solvable in §3.3.3.

We will now prove that the tensor product $\mathcal{D} \otimes \mathcal{D}$ is not solvable. To show this, we must exhibit a particular continuation $k : X \times X \to R \times R$ with the property that

$$\mathbf{B}_{\mathcal{D} \otimes \mathcal{D}} k : X \times X \to \mathscr{P}(X \times X)$$

has no fixpoints. This is equivalent to saying that the game with outcome function k has no pure strategy Nash equilibrium. Therefore, we take k to be the outcome function of matching pennies, namely

$$k(x, y) = \begin{cases} (1, 0) & \text{if } x = y \\ (0, 1) & \text{if } x \neq y \end{cases}$$

This gives

$$\mathbf{B}_{\mathcal{D} \otimes \mathcal{D}} k(x, y) = \{(y, \overline{x})\}$$

where $\overline{\cdot} : X \to X$ interchanges H and T. This function indeed has no fixpoint.

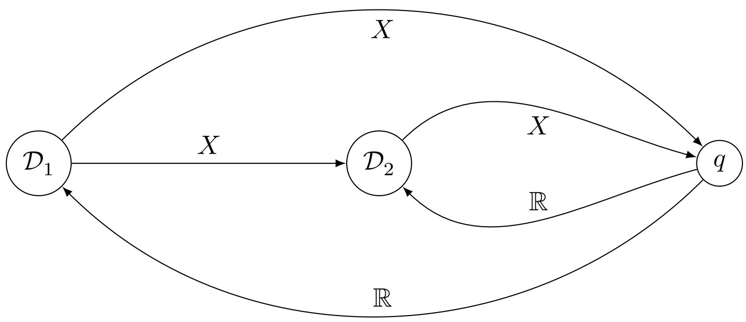
If we reasoned in $\mathbf{Game}_{SP}(\mathbf{SRel})$ instead of $\mathbf{Game}_{SP}(\mathbf{Set})$ then this does not happen: by Nash's theorem, the tensor product of finitely many decisions over finite sets of choices in $\mathbf{Game}_{SP}(\mathbf{SRel})$ is solvable.

3.3.7 Failure of compositional Nash's theorem

An early motivation for introducing solvable games, besides connecting open games and selection functions (§3.3.5), was to prove a suitable generalisation of Nash's theorem for open games. The fact that solvable games are closed under composition (§3.3.4), but fail to be closed under tensor because pure strategy Nash equilibria need not exist (§3.3.6), strongly suggests the conjecture that in $\mathbf{Game}_{SP}(\mathbf{SRel})$, *all freely generated games are solvable.*

This fails, however, because it is not the case that arbitrary solvable games in $\mathbf{Game}_{SP}(\mathbf{SRel})$ are closed under tensor product. This appears to result from a much deeper issue with the definitions in §2.1 and §2.2. In §3.1 it was shown that normal form games with mixed strategies Nash equilibria can be faithfully represented by open games, and similarly for extensive form games and pure strategy subgame perfect equilibria in §3.2. However, if the two are combined, we end up with a uselessly implausible solution concept. The previous conjecture, that all freely generated games are solvable, can be taken as a requirement for a repaired theory.

We will end with a specific counterexample to the conjecture. Consider the game in $\mathbf{Game}_{SP}(\mathbf{SRel})$ denoted by



or $\mathcal{G} = q \circ_{SP} (X \otimes \mathcal{D}_2 \otimes \mathbb{R}^*) \circ_{SP} \mathcal{D}_1$, where $X = \{H, T\}$. $\mathcal{D}_2$ is an ordinary decision that maximises expected utility, and $\mathcal{D}_1$ also maximises expected utility but chooses a pair from $X \times X$ with a mixed strategy. The strategically trivial game q (which could be decomposed into computations and counits) implements the outcome function, which is again that of matching pennies. Specifically, the coplay function $\mathbf{C}_q : X \times X \to \mathscr{D}(\mathbb{R} \times \mathbb{R})$ is given by

$$\mathbf{C}_q(x, y) = \begin{cases} \delta(0,1) & \text{if } x = y \\ \delta(1,0) & \text{if } x \neq y \end{cases}$$

where δ is the unit of the monad $\mathscr{D}$ (see §2.1.3).

Since $\mathcal{G}$ is a scalar, its best response function has the form

$$\mathbf{B}_{\mathcal{G}} : \Sigma_{\mathcal{G}} \to \mathscr{P}\Sigma_{\mathcal{G}}$$

Despite the fact that we are working over $\mathbf{SRel}$ and so we have mixed strategies available, and the fact that $\mathcal{G}$ has been built from only expectation-maximising decisions, computations and counits using composition and tensor, we will prove that $\mathbf{B}_\mathcal{G}$ has no fixpoints.

The set of strategies is

$$\Sigma_\mathcal{G} = \Sigma_{\mathcal{D}_1} \times \Sigma_{\mathcal{D}_2} = \mathscr{D}(X \times X) \times (X \to \mathscr{D}X)$$

Geometrically, this is the product of a tetrahedron and a square, and hence is closed, convex and finite-dimensional, satisfying the hypotheses of the Kakutani fixpoint theorem.

To begin with, since q is strategically trivial we have

$$\mathbf{B}_\mathcal{G}(*, *)(\sigma, \tau) = \mathbf{B}_{(X \otimes \mathcal{D}_2 \otimes \mathbb{R}^*) \circ_{SP} \mathcal{D}_1}(*, k)(\sigma, \tau)$$

where $k : X \times X \to \mathscr{D}(\mathbb{R} \times \mathbb{R})$ is the continuation $k = *_{*\circ} = q$. This is

$$\mathbf{B}_{\mathcal{D}_1}(*, k_{\tau\circ})\sigma \times \bigcap_{\sigma' : \mathscr{D}(X \times X)} \mathbf{B}_{X \otimes \mathcal{D}_2 \otimes \mathbb{R}^*}(\mathbf{P}_{\mathcal{D}_1}\sigma' \circ *, k)\tau$$

We will only need to focus on the second part of this. Since $\mathcal{D}_1$ is a decision we have $\mathbf{P}_{\mathcal{D}_1}\sigma' \circ * = \sigma'$. Then

$$\mathbf{B}_{X \otimes \mathcal{D}_2 \otimes \mathbb{R}^*}(\sigma', k)\tau = \mathbf{B}_{X \otimes \mathcal{D}_2}(\sigma', k_1)\tau$$

where $k_1 : X \times X \to \mathscr{D}\mathbb{R}$ is given by

$$k(x_1, x_2) = \begin{cases} \delta 0 & \text{if } x_1 = x_2 \\ \delta 1 & \text{if } x_1 \neq x_2 \end{cases}$$

The game $X \otimes \mathcal{D}_2 : X \otimes X \to X \otimes X \otimes \mathbb{R}^*$ is essentially the component of $\mathcal{G}$ that causes the problems. Its best response function is

$$\mathbf{B}_{X \otimes \mathcal{D}_2}(h, k_1)\tau = \mathbf{B}_{\mathcal{D}_2}(h_1, k')\tau$$

The continuation $k' = (k_1)_{*(h_1)\otimes} : X \to \mathscr{D}\mathbb{R}$ is given by $k'x_2 = k_1(h_1, x_2)$, where we are abusing notation as in §3.1.7. Explicitly, using the particular k we defined,

$$\mathbb{P}\left[k'x_2 = 1\right] = \mathbb{P}\left[h_1 \neq x_2\right]$$

Now using the definition of $\mathbf{B}_{\mathcal{D}_2}$,

$$\mathbf{B}_{X \otimes \mathcal{D}_2}(h_1, k_1)\tau = \{\tau' : X \to \mathscr{D}X \mid \mathbb{E}[k'(\tau' h_1)] \geq \mathbb{E}[k'x_2] \text{ for } x_2 = H, T\}$$

Putting this together, if we had a fixpoint (σ, τ) of $\mathbf{B}_\mathcal{G}$, then for every possible history $h : \mathscr{D}(X \times X)$ we would have

$$\mathbb{E}[k_1(h_1, \tau h_1)] \geq \mathbb{E}[k_1(h_1, x_2)]$$

for $x_2 = H, T$. We will now prove that this is impossible.

Pick some $x_1 \in \operatorname{supp}(\tau H)$, so $\mathbb{P}[\tau H = x_1] > 0$. We will take $h = \delta x_1 \otimes \delta H$, that is, h is the probability distribution with $\mathbb{P}[h = (x_1, H)] = 1$. Then

$$k'(\tau h_1) = k'(\tau x_1) = k_1(x_1, \tau H)$$

and since $\mathbb{P}[\tau H = x_1] > 0$ we have $\mathbb{E}[k_1(x_1, \tau H)] < 1$. This means that player 2 has incentive to unilaterally deviate to $x_2 \neq x_1$, which has

$$\mathbb{E}[k'x_2] = \mathbb{E}[k'(x_1, x_2)] = 1$$

Conclusion

The future of compositional game theory

Suppose an economist wants to create a mathematical model of some economic system, say, a new market. Using her experience as a working economist, she analyses the market and divides it into a number of interacting components. Most of these component markets are well known and have been extensively studied, and she recognises the last component as behaving similarly to part of another market that was recently studied by a colleague.

She opens her software suite and loads these known models, most of which come pre-installed, and the last downloaded from a source repository accompanying her colleague's research paper. Each of these specifies an open game, whose ports specify how it communicates with an arbitrary environment, which is graphically represented on the screen by a bead with open strings. Although she has never heard the phrase 'category theory' except in a cryptic footnote in the software's documentation, she draws strings connecting the beads, in a way corresponding to her intuition about how the component markets are communicating and influencing each other.

Behind the scenes, the software compiles the string diagram to an intermediate representation, and then to Haskell, which is compiled and optimised by `ghc`.

The economist begins by specifying a plausible strategy profile, for example by drawing a finite state automaton, and then plays the game with the strategy profile to obtain a plot showing a probability distribution on outcomes. She tests this against real econometric data and finds that it does not fit the data. She then replaces some parameter in the strategy profile by a variable, and runs an automated optimisation procedure. This takes 30 minutes to complete, and still does not fit the data well.

By tracing the results through the string diagram, viewing plots of the data flowing along the intermediate strings, she determines that her colleague's model is at fault. She 'zooms in' to that model in the editor, seeing visually the information flow inside that model. It consists of many standard components interacting, each of which itself can be opened as a string diagram for editing. She manually tunes several quantitative parameters, and qualitatively changes the logic of the information flow in a few places, in an iterative process, at each stage comparing the simulation results to the econometric data.

Finally, she obtains a model and a strategy profile that is a good fit for the data. The next day, a three-hour computation confirms that the strategy profile is an equilibrium, suggesting that the model is stable. From here, she can answer questions of economic interest. For example, she can simulate economic events such as changing the structure of the market in some way, or changing the response of a neighbouring market. She can test the existing strategy profile with the modified game, and the software will determine which agents now have incentive to deviate. In some cases, she is able, through a mixture of intuition and automatic optimisation procedures, to find a new equilibrium to which the old strategy profile will plausibly settle given the changes made to the game. This constitutes an economic prediction, about how the market will respond to a given event once it settles back into equilibrium.

Realistically, of course, this is very optimistic. However I believe the description of the technology is entirely plausible; the main question is whether nontrivial and reasonably accurate predictions about macroeconomic systems can be made using such a compositional game model, and whether compositionality of games is a good model for compositionality of macroeconomic systems. I conjecture that computations made from a user-supplied strategy profile can be done efficiently, in time linear in the size of the game; although to compute an approximate equilibrium from nothing we run into the fundamental problem that equilibria of both perfect information and normal form games are expensive to compute.

I will now discuss what needs to be done for this vision of the future to come true, with the exception of the economic questions, which are far outside of my expertise. These divide into theoretical problems and implementation.

A direct translation of the definitions in §2.1 and §2.2 into a programming language will consist of combinators for composing games, and each game will implement the play, coplay and best response functions, each taking as one input a strategy profile implemented as a tuple of functions. In particular, the play function will convert a strategy profile into a distribution of strategic plays (in the case of probabilistic choice), and the best response function can be used to decide whether a given strategy is a Nash equilibrium. The prototype described in §0.2 already has these features, and it is not difficult to imagine other features such as search and optimisation algorithms sitting above these primitive features, although it might be more difficult to make these run in a feasible amount of time.

The software Quantomatic [KMS14] is currently able to reason about string diagrams using graph rewrite rules, but it does not convert the string diagram into the logical language of composition and tensor, which is what is needed in game theory. I propose an extensible, language-independent API implementing the logical language of morphisms of monoidal categories, which can be used as an intermediate language between Quantomatic (or similar software) and several backends. For example, different code generators could target MATLAB for **FVect**, Maple for **Rel** and Haskell for $\mathbf{Game}_N(\mathcal{C})$.

Far more work is needed on the theoretical side. A typical example of a game in practical economics may have several awkward features simultaneously: dependent types, infinite repetition, incomplete information, learning, irrationality, ...

For such practical examples, each of these needs to be represented in compositional game theory, in compatible ways. For several of these problems, my preferred approach is to apply intuition from functional programming and change the underlying category to successively add new features. Informally, I think of this in terms of 'effects stacks'. A simple example may have probabilistic choice at the bottom of the stack, following by learning, followed by nondeterminism, followed by selection. As the first step, probabilistic choice is the monad

$$T_1 X = \mathscr{D} X$$

introduced in §2.1.3. The second step is to apply the state monad transformer

$$T_2 X = \mathrm{St}_A^{T_1} X = A \to T_1(X \times A) = A \to \mathscr{D}(X \times A)$$

Alternatively, this point can be reached using Lawvere theories rather than monad transformers. Adding nondeterminism to a stack involving probability is not straightforward, but there are two approaches discussed in [VW06]. An alternative possibility is to use synthetically compact sets [Esc04] to represent nondeterminism, which leads to

$$T_3 X = \mathscr{J}_{\mathbb{B}}(T_2 X) = ((A \to \mathscr{D}(X \times A)) \to \mathbb{B}) \to A \to \mathscr{D}(X \times A)$$

which is a monad due to the unpublished fact (observed by Fiore and Griffin in 2011) that for every monad T there is a distributive law

$$\lambda : T \circ \mathscr{J}_{\mathbb{B}} \to \mathscr{J}_{\mathbb{B}} \circ T$$

Finally, selections can be added using the generalised selection monad in [EO15], resulting in

$$\begin{aligned} T_4 X &= \mathscr{J}_R^{T_3} X \\ &= (X \to R) \to T_4 X \\ &= (X \to R) \to ((A \to \mathscr{D}(X \times A)) \to \mathbb{B}) \to A \to \mathscr{D}(X \times A) \end{aligned}$$

where R is a suitable algebra of T_4, ultimately derived from the expectation operator $\mathbb{E} : \mathscr{D}\mathbb{R} \to \mathbb{R}$.

Although this looks complicated, machinery exists in functional programming to work with compound monads like this. I think it will be necessary to make a systematic study of this before serious applications of compositional game theory in economics become possible. In general, the Kleisli categories of noncommutative monads are symmetric premonoidal (and moreover closed Freyd). It may be that for premonoidal categories, string diagrams in the sense of [Jef97] are impractical and should be replaced with a form of arrow notation [Pat01]. However, in an effects stack in which each player has their own state variable that is private to the other players (such as Bayesian beliefs), any computations done by different players will commute past each other. This results in far more allowed topological moves on string diagrams than for general premonoidal categories, and so I suggest that the software will also need to keep track of allowed moves on a string diagram, based on which morphisms of the category commute.

The status of string diagrams

Although it is often emphasised that string diagrams are a fully formal algebraic language, as was pointed out in §2.3.1, this is not yet the case for string diagrams in game theory. In general, string diagrams are made fully formal by a *coherence theorem*, which is a statement of both soundness and completeness. The soundness part says that if two diagrams are equivalent under a certain class of topological moves then their denotations are naturally isomorphic, and completeness is the converse. A survey of many such theorems can be found in [Sel11]. Perhaps the most elegant such theorem is that for compact closed categories, for which the allowed moves are precisely the topological deformations.

A key difference in game theory is that we have a single, concretely defined category, rather than a axiomatically defined family. In this situation it should be expected that we do not have completeness, roughly because there are equivalences between games that hold for 'game theoretic reasons' rather than 'structural reasons'. For example in classical game theory, games are invariant under any affine transformation of the utilities, and there is no reason to expect that these invariances can be visualised with a string diagram.

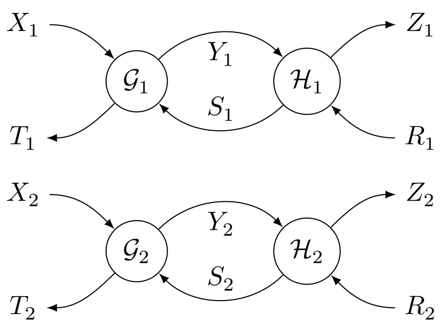
Although this thesis does prove that many topological moves are sound, most of them coming from the symmetric monoidal category structure, we have not proved a sharp theorem giving an explicit list of allowed moves. As a starting point I make the following conjecture: if two cap-free (that is, unit-free, see §2.3.6) are equivalent by arbitrary topological deformations then their denotations are naturally isomorphic. That is to say, the language of string diagrams in game theory is precisely the language of compact closed categories, with caps disallowed. Proving this would require a careful geometric analysis, classifying the possible moves as those that only involve covariant parts, those that only involve contravariant parts, and those that involve an interaction between the two (such as the counit law §2.3.6, or a covariant-contravariant crossing §2.3.7). For example, it would have to be argued that if two cap-free diagrams are equivalent, then they are equivalent without going via an intermediate point involving a cap.

Another complication to consider is the behaviour of the copying and deleting operators. For computations we have a naturality condition: copying the output of a computation is equivalent to copying the input and applying the computation to each copy. For players making choices, however, this will not hold in general, because two copies of the player could choose differently from multiple equilibria. This is reminiscent of the behaviour of effects in functional programming: running an effectful computation and copying the result is not equivalent to running the computation twice, whereas for pure functions they are equivalent.

On the two composition operators

An accidental theme in this thesis has been the contrast between the categories $\mathbf{Game}_N(\mathcal{C})$ and $\mathbf{Game}_{SP}(\mathcal{C})$. This was never intended, however. Originally the thesis used only $\mathbf{Game}_{SP}(\mathcal{C})$, called simply $\mathbf{Game}(\mathcal{C})$, and incorrectly claimed it was symmetric monoidal; my examiners noticed that I had (accidentally) omitted the check that $\otimes$ was a bifunctor, which turned out to be false. As a result, the structure of the thesis is somewhat awkward, because of this fairly major change during the corrections process.

My view currently is that the definition of $\mathbf{Game}_{SP}(\mathcal{C})$ is *wrong*, and its use in this thesis now amounts to a historical accident. There is no particular reason why the use of subgame perfect equilibrium should lead to only a premonoidal category, and intuitively, string diagrams should still be valid. The fundamental problem is the interpretation in $\mathbf{Game}_{SP}(\mathcal{C})$ of the following string diagram:



The immediate problem is that the two readings of this, namely

$$(\mathcal{H}_1 \circ_{SP} \mathcal{G}_1) \otimes (\mathcal{H}_2 \circ_{SP} \mathcal{G}_2)$$

and

$$(\mathcal{H}_1 \otimes \mathcal{H}_2) \circ_{SP} (\mathcal{G}_1 \otimes \mathcal{G}_2)$$

are different. The equality between these two readings is precisely the condition that $\otimes$ is a bifunctor on the category whose composition is $\circ_{SP}$, and so is precisely what fails when trying to interpret string diagrams in a premonoidal category.

We can see precisely what fails by carefully stepping through the proofs in §2.2.8, §2.2.9 and §A.4. The difference between the two readings relates to how the players in $\mathcal{G}_1$ reason about those in $\mathcal{G}_2$, and vice versa. We fix a strategy profile

$$(\sigma_1, \tau_1, \sigma_2, \tau_2) : \Sigma_{\mathcal{G}_1} \times \Sigma_{\mathcal{H}_1} \times \Sigma_{\mathcal{G}_2} \times \Sigma_{\mathcal{H}_2}$$

and consider the best responses to it. For each pair of components, the players in the first component can either reason about those in the second as though they are using the fixed strategy profile, or as though they can deviate to an arbitrary strategy profile. In the first reading, $\mathcal{H}_1$ reasons as though $\mathcal{G}_1$ is deviating, and $\mathcal{H}_2$ reasons as though $\mathcal{G}_2$ is deviating, and every other pair reasons as though the strategy profile is fixed. In the second reading, however, $\mathcal{G}_1$ and $\mathcal{G}_2$ additionally reason about each other as though they are deviating.

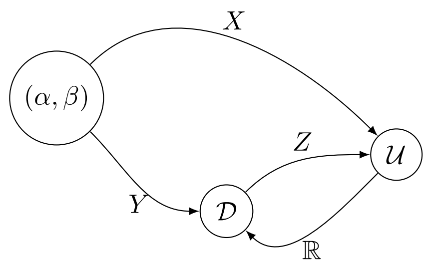
My current opinion is that one of these two readings should be universally *correct* on game theoretic grounds, and the other should be wrong. Since $\mathcal{G}_1$ and $\mathcal{G}_2$ happen simultaneously, by analogy to the simpler game $\mathcal{G}_1 \otimes \mathcal{G}_2$ it seems more likely that the first reading is correct: $\mathcal{G}_1$ and $\mathcal{G}_2$ should not consider each other to be deviating. The challenge is then to find a new composition operator in which both readings of the diagram are equal, and are equal to the first reading with $\circ_{SP}$.

One direct way to do this is to extend the 4-tuple definition of an open game in §2.1.4 with a fifth component, a recursively-built collection of subgames $U_{\mathcal{G}}$. Then the best response function should take an element of $U_{\mathcal{G}}$ as an additional parameter, indicating that the best response condition for Nash is being checked for the given subgame. The best response condition for subgame perfect equilibria is then equivalent to this holding for every element of $U_{\mathcal{G}}$. This approach is also better from a computational perspective: the prototype mentioned in §0.2 always used $\circ_N$, even when this thesis exclusively used $\circ_{SP}$, because the intersection over all strategies is not a constructive operation.

This idea, of recursively building a type of subgames, occurred to me early in the development of compositional game theory; it predates the definition of $\circ_{SP}$, which was intended to be a simplification that achieved the same thing without needing an additional piece of data defining open games.

Mixed strategies and Bayesian reasoning

The problem discussed in §3.3.7 can be simplified to a single-player situation, by replacing the first player with a pair of dependent random variables. Consider the string diagram



Here $(\alpha, \beta) : \mathscr{D}(X \times Y)$ is a pair of dependent random variables, $\mathcal{D}$ is an expected utility-maximising decision, and $\mathcal{U}$ is a utility function combined with a counit.

Before making a decision, the player at $\mathcal{D}$ concretely (non-probabilistically) observes a $y : Y$. Since the structure of the diagram is common knowledge, including the exact distributions of the random variables, she can use Bayes' theorem to obtain a new distribution for α. Most naively, this Bayesian updating operation could be manually built into the definition of categorical composition.

The main technical difficulty is that the behaviour must be well-defined even if the concrete observation made is not in the support of the random variable. Put another way, the problem is what the posterior probability should be, after observing an event of prior probability zero. A solution to this problem is provided by sequential equilibrium, which could also naively be built in to the definition categorical composition.

For purposes of proving theorems (as well as mathematical elegance) it would be very useful to express the sequential equilibrium condition in a more abstract way, whereas it is usually written in terms of perturbations, or limits of sequences of distributions. A nice possibility would be to use †-categories, in which a morphism $f : \hom(X, Y)$ can be reversed to a morphism $f^\dagger : \hom(Y, X)$. In the †-category of sets and relations, seen as the kleisli category of the powerset monad, the converse of a morphism $f : X \to \mathscr{P}Y$ is

$$f^\dagger y = \{x : X \mid y \in fx\}$$

This is a sort of possibilistic analogue of Bayesian updating: $f^\dagger y$ is the set of possibilities for x, given that the observed value of fx is y. The category of stochastic relations fails to be a †-category in the obvious way precisely because of the problem of observing an event of prior probability zero. It would be most ideal if the modified Bayesian updating used in sequential equilibrium can be described as a particular †-category, because then it would be possible to reason quite abstractly.

Morphisms between games

One of the clear next stages in the development of open games, for several reasons, is the consideration of morphisms between open games. The primary technical reason is that it would be natural to define infinite games as final coalgebras of a functor which composes one additional stage onto a finite approximation of the infinite game. Since games are themselves morphisms, this brings us into the realm of higher category theory.

Given a pair of games $\mathcal{G}, \mathcal{H} : X \otimes S^* \to Y \otimes R^*$ of the same type, there is an obvious definition of 2-cell between them: namely, a function $f : \Sigma_\mathcal{G} \to \Sigma_\mathcal{H}$ which commutes with the other data specifying $\mathcal{G}$ and $\mathcal{H}$. If we no longer quotient games by isomorphisms of sets as we did in §2.2.2, it appears likely that the resulting structure will be a monoidal bicategory [SP09], because the categorical and monoidal structure at the 1-level holds up to isomorphism at the 2-level.

We can also consider morphisms between open games with different types, resulting in a structure called a monoidal double category [Shu10]. To quote from that paper: "There is a good case to be made ... that often the extra morphisms should *not* be discarded ... in many cases symmetric monoidal bicategories are a red herring, and really we should be studying symmetric monoidal double categories."

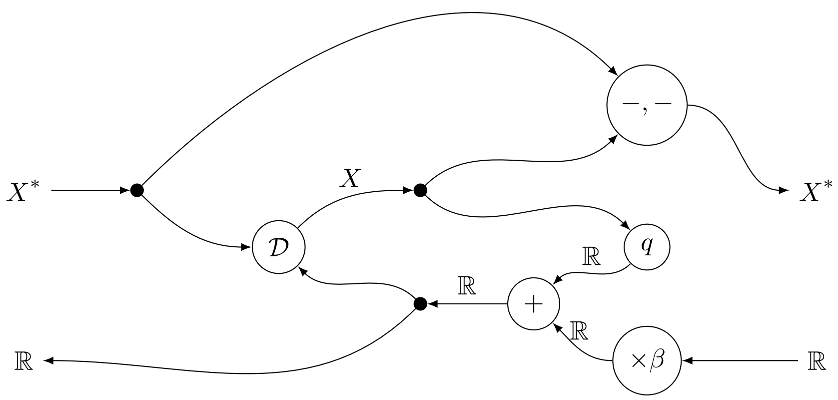
However, finding a more general definition of 2-cells is not immediately straightforward. Suppose we would like to define 2-cells $\alpha : \mathcal{G} \implies \mathcal{H}$, where $\mathcal{G} : X_1 \otimes S_1^* \to Y_1 \otimes R_1^*$ and $\mathcal{H} : X_2 \otimes S_2^* \to Y_2 \otimes R_2^*$, beginning with a function $\alpha_\Sigma : \Sigma_\mathcal{G} \to \Sigma_\mathcal{H}$. The obvious next step is to choose functions between X_1 and X_2 etc., either in one direction or the other, and demand that these functions commute with the data specifying $\mathcal{G}$ and $\mathcal{H}$. Unfortunately, this puts demands on the directions of these functions that seem to be impossible to satisfy. A relation-based definition may still work, however.

The meaning of coplay

The most important test-case for such a definition is the ability to define repeated games. For simplicity, we will consider a repeated decision problem: a player repeatedly chooses an element of a set X, having observed the history (of type X^*, the finite lists of X). The utility function for the player in the stage decision is $q : X \to \mathbb{R}$, which is discounted by a factor $0 < \beta < 1$. Thus, the utility function $\widetilde{q} : X^\omega \to \mathbb{R}$ for the repeated decision is

$$\widetilde{q}x = \sum_{i=0}^{\infty} \beta^i \cdot qx_i$$

Consider the string diagram



This is possibly the most complicated string diagram appearing in this thesis. More importantly, it is the only example given of a fully-fledged game (rather than a component of a game) that uses coplay in a nontrivial way, and so it is key to gaining an intuitive understanding of coplay. We will now discuss this game in some detail.

The forward-flowing part of the game is relatively straightforward. A finite history of type $x : X^*$ is observed from the past. The player observes this and chooses a next move x'. This new move is appended to the end of the history, to give x, x' (the $-,- : X^* \otimes X \to X^*$ operator is reverse cons, sometimes called 'snoc'), which is then outputted for future observation.

In the reverse direction, the real number that is inputted from the future is interpreted by the model as the influence on this stage's utility from all future stages. The discounting of each stage is determined, however, not from the beginning of the game, but relative to the current stage: the immediately following stage is not discounted, the next stage after that is discounted by a factor of β, the next stage by β^2, etc. Suppose the stage depicted in the string diagram is the nth, then the value inputted is

$$\sum_{i=0}^{\infty} \beta^i \cdot qx_{n+i+1}$$

The first thing done to this value is to multiply the whole by β, which shifts the discounting amount 'one stage into the future', giving

$$\beta \cdot \sum_{i=0}^{\infty} \beta^i \cdot qx_{n+i+1} = \sum_{i=1}^{\infty} \beta^i \cdot qx_{n+i}$$

This is then added to the utility from the current stage, giving

$$qx_n + \sum_{i=1}^{\infty} \beta^i \cdot qx_{n+i} = \sum_{i=0}^{\infty} \beta^i \cdot qx_{n+i}$$

The decision in the current stage acts to maximise this value, which is equivalent to maximising the entire sum, because the utility from earlier stages is independent of the current choice. Additionally this value, which is the utility from this stage and all future stages, is outputted as coutility for the $(n-1)$th stage to input as its own future utility. The value can be equivalently written as

$$\sum_{i=0}^{\infty} \beta^i \cdot qx_{n+i} = \sum_{i=0}^{\infty} \beta^i \cdot qx_{(n-1)+i+1}$$

and hence is in the form that the $(n-1)$th stage is expecting.

Appendix: The structure of $\mathbf{Game}_{SP}(\mathcal{C})$

A.1 Discussion

This appendix closely mirrors §2.2, starting with §2.2.5, but uses SP-composition rather than N-composition. The definition of $\mathcal{H} \circ_{SP} \mathcal{G}$ differs from $\mathcal{H} \circ_N \mathcal{G}$ only in the best response function, but for completeness we will reproduce the entire definition here. The set of strategy profiles is

$$\Sigma_{\mathcal{H} \circ_{SP} \mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}$$

The play function is

$$\mathbf{P}_{\mathcal{H} \circ_{SP} \mathcal{G}}(\sigma, \tau) = \mathbf{P}_{\mathcal{H}} \tau \circ \mathbf{P}_{\mathcal{G}} \sigma$$

The coplay function $\mathbf{C}_{\mathcal{H} \circ_{SP} \mathcal{G}}(\sigma, \tau)$ is the composition

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{\mathcal{G}} \sigma \otimes R} X \otimes Y \otimes R \xrightarrow{X \otimes \mathbf{C}_{\mathcal{H}} \tau} X \otimes S \xrightarrow{\mathbf{C}_{\mathcal{G}} \sigma} T$$

The best response function, which is different to that for N-composition, is

$$\mathbf{B}_{\mathcal{H} \circ_{SP} \mathcal{G}}(h, k)(\sigma, \tau) = \mathbf{B}_{\mathcal{G}}(h, k_{\tau \circ})\sigma \times \bigcap_{\sigma' : \Sigma_{\mathcal{G}}} \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}} \sigma' \circ h, k)\tau$$

Because we are using a different composition operator we have a different category, which we call $\mathbf{Game}_{SP}(\mathcal{C})$. All other definitions remain unchanged, including the definition of the tensor product operator in §2.2.7. However one of the conditions for a monoidal category, namely that the tensor product is a bifunctor (which was proved for N-composition in §2.2.8 and §2.2.9) fails for SP-composition. As a result, the category $\mathbf{Game}_{SP}(\mathcal{C})$ is only a premonoidal category [PR93], not a monoidal category.

The purpose of this appendix is to give a complete, self-contained proof of this fact. It is designed to be independent of the proofs in §2.2 both logically and in presentation. The cost of this is that a large amount of material from that section has been duplicated here.

A.2 The identity laws

We begin with the identity laws of a category. Let $\mathcal{G} : (X, S) \to (Y, R)$ be a game. We first prove that $(\mathrm{id}_Y, \mathrm{id}_R) \circ_{SP} \mathcal{G} \sim \mathcal{G}$, and hence they are equal after quotienting.

For the strategy sets,

$$\Sigma_{(\mathrm{id}_Y,\,\mathrm{id}_R)\circ_{SP}\mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{(\mathrm{id}_Y,\,\mathrm{id}_R)} = \Sigma_{\mathcal{G}} \times 1$$

and so we take the isomorphism $i : \Sigma_{\mathcal{G}} \times 1 \to \Sigma_{\mathcal{G}}$. For the play function,

$$\mathbf{P}_{(\mathrm{id}_Y,\,\mathrm{id}_R)\circ_{SP}\mathcal{G}}(i\sigma) = \mathbf{P}_{(\mathrm{id}_Y,\,\mathrm{id}_R)}* \circ \mathbf{P}_{\mathcal{G}}\sigma = \mathrm{id}_Y \circ \mathbf{P}_{\mathcal{G}}\sigma = \mathbf{P}_{\mathcal{G}}\sigma$$

For coplay, by definition $\mathbf{C}_{(\mathrm{id}_Y,\,\mathrm{id}_R)\circ_{SP}\mathcal{G}}(i\sigma)$ is the composition

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{\mathcal{G}}\sigma \otimes R} X \otimes Y \otimes R$$
$$\xrightarrow{X \otimes \mathbf{C}_{(\mathrm{id}_Y,\,\mathrm{id}_R)}*} X \otimes R \xrightarrow{\mathbf{C}_{\mathcal{G}}\sigma} S$$

which, expanding the definitions further, is

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{\mathcal{G}}\sigma \otimes R} X \otimes Y \otimes R \xrightarrow{X \otimes \pi_2} X \otimes R \xrightarrow{\mathbf{C}_{\mathcal{G}}\sigma} S$$

The first part of this is the identity on $X \otimes R$, so it is equal to $\mathbf{C}_{\mathcal{G}}\sigma$.

For best response we have

$$\begin{aligned} & i\sigma' \in \mathbf{B}_{(\mathrm{id}_Y,\mathrm{id}_R)\circ_{SP}\mathcal{G}}(h,k)(i\sigma) \\ \iff & (\sigma', *) \in \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \times \bigcap_{\sigma'' : \Sigma_{\mathcal{G}}} \mathbf{B}_{(\mathrm{id}_Y,\mathrm{id}_R)}(\mathbf{P}_{\mathcal{G}}\sigma'' \circ h, k)* \\ \iff & \sigma' \in \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \end{aligned}$$

The continuation $k_{*\circ}$ is given by

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes \mathbf{P}_{(\mathrm{id}_Y,\,\mathrm{id}_R)}*} Y \otimes Y \xrightarrow{Y \otimes k} Y \otimes R \xrightarrow{\mathbf{C}_{(\mathrm{id}_Y,\,\mathrm{id}_R)}*} R$$

Expanding the definitions and simplifying, this is

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes k} Y \otimes R \xrightarrow{\pi_2} R$$

which is equal to k, and we are done.

For the other identity law, we will prove that $\mathcal{G} \circ_{SP} (\mathrm{id}_X, \mathrm{id}_S) \sim \mathcal{G}$. For the strategy sets,

$$\Sigma_{\mathcal{G}\circ_{SP}(\mathrm{id}_X,\,\mathrm{id}_S)} = \Sigma_{(\mathrm{id}_X,\,\mathrm{id}_S)} \times \Sigma_{\mathcal{G}} = 1 \times \Sigma_{\mathcal{G}} \cong \Sigma_{\mathcal{G}}$$

now with the isomorphism $i : \Sigma_{\mathcal{G}} \to 1 \times \Sigma_{\mathcal{G}}$. For the play function,

$$\mathbf{P}_{\mathcal{G}\circ_{SP}(\mathrm{id}_X,\,\mathrm{id}_S)}(i\sigma) = \mathbf{P}_{\mathcal{G}}\sigma \circ \mathbf{P}_{(\mathrm{id}_X,\,\mathrm{id}_S)}* = \mathbf{P}_{\mathcal{G}}\sigma \circ \mathrm{id}_X = \mathbf{P}_{\mathcal{G}}\sigma$$

For coplay we have that $\mathbf{C}_{\mathcal{G}\circ_{SP}(\mathrm{id}_X,\,\mathrm{id}_S)}(i\sigma)$ is the composition

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{(\mathrm{id}_X,\,\mathrm{id}_S)}* \otimes R} X \otimes X \otimes R$$
$$\xrightarrow{X \otimes \mathbf{C}_{\mathcal{G}}\sigma} X \otimes S \xrightarrow{\mathbf{C}_{(\mathrm{id}_X,\,\mathrm{id}_S)}*} S$$

Expanding and simplifying, this is

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{C}_{\mathcal{G}}\sigma} X \otimes S \xrightarrow{\pi_2} S$$

which is equal to $\mathbf{C}_\mathcal{G}\sigma$.

For best response we have

$$\begin{aligned}
& i\sigma' \in \mathbf{B}_{\mathcal{G}\circ_{SP}(\mathrm{id}_X,\mathrm{id}_S)}(h,k)(i\sigma) \\
\iff & (*,\sigma') \in \mathbf{B}_{(\mathrm{id}_X,\mathrm{id}_S)}(h,k_{\sigma\circ})* \times \bigcap_{\sigma'':\Sigma_{(\mathrm{id}_X,\mathrm{id}_S)}} \mathbf{B}_\mathcal{G}(\mathbf{P}_{(\mathrm{id}_X,\mathrm{id}_S)}\sigma''\circ h,k)\sigma \\
\iff & \sigma' \in \mathbf{B}_\mathcal{G}(\mathbf{P}_{(\mathrm{id}_X,\mathrm{id}_S)}*\circ h,k)\sigma \\
\iff & \sigma' \in \mathbf{B}_\mathcal{G}(h,k)\sigma
\end{aligned}$$

because $\mathbf{P}_{(\mathrm{id}_X,\mathrm{id}_S)}* = \mathrm{id}_X$.

In summary, we have proved that the identity for SP-composition on an object (X,R) is $(\mathrm{id}_X,\mathrm{id}_R)$.

A.3 Associativity

Consider games

$$(W,U) \xrightarrow{\mathcal{G}} (X,T) \xrightarrow{\mathcal{H}} (Y,S) \xrightarrow{\mathcal{I}} (Z,R)$$

We will prove that

$$\mathcal{I}\circ_{SP}(\mathcal{H}\circ_{SP}\mathcal{G}) \sim (\mathcal{I}\circ_{SP}\mathcal{H})\circ_{SP}\mathcal{G}$$

For SP-composition, more than for N-composition, this associativity law is not obvious.

The two sets of strategy profiles are $\Sigma_{\mathcal{I}\circ_{SP}(\mathcal{H}\circ_{SP}\mathcal{G})} = (\Sigma_\mathcal{G}\times\Sigma_\mathcal{H})\times\Sigma_\mathcal{I}$ and $\Sigma_{(\mathcal{I}\circ_{SP}\mathcal{H})\circ_{SP}\mathcal{G}} = \Sigma_\mathcal{G}\times(\Sigma_\mathcal{H}\times\Sigma_\mathcal{I})$, so we take the isomorphism

$$i : (\Sigma_\mathcal{G}\times\Sigma_\mathcal{H})\times\Sigma_\mathcal{I} \to \Sigma_\mathcal{G}\times(\Sigma_\mathcal{H}\times\Sigma_\mathcal{I})$$

The case for the play function follows immediately from associativity of composition in the underlying category $\mathcal{C}$:

$$\begin{aligned}
\mathbf{P}_{\mathcal{I}\circ_{SP}(\mathcal{H}\circ_{SP}\mathcal{G})}((\sigma,\tau),\upsilon) &= \mathbf{P}_\mathcal{I}\upsilon\circ\mathbf{P}_{\mathcal{H}\circ_{SP}\mathcal{G}}(\sigma,\tau) \\
&= \mathbf{P}_\mathcal{I}\upsilon\circ\mathbf{P}_\mathcal{H}\tau\circ\mathbf{P}_\mathcal{G}\sigma \\
&= \mathbf{P}_{\mathcal{I}\circ_{SP}\mathcal{H}}(\tau,\upsilon)\circ\mathbf{P}_\mathcal{G}\sigma \\
&= \mathbf{P}_{(\mathcal{I}\circ_{SP}\mathcal{H})\circ_{SP}\mathcal{G}}(i((\sigma,\tau),\upsilon))
\end{aligned}$$

For coplay, by definition $\mathbf{C}_{\mathcal{I}\circ_{SP}(\mathcal{H}\circ_{SP}\mathcal{G})}((\sigma,\tau),\upsilon)$ is the composition

$$W\otimes R \xrightarrow{\Delta_W\otimes R} W\otimes W\otimes R \xrightarrow{W\otimes\mathbf{P}_{\mathcal{H}\circ_{SP}\mathcal{G}}(\sigma,\tau)\otimes R} W\otimes Y\otimes R$$
$$\xrightarrow{W\otimes\mathbf{C}_\mathcal{I}\upsilon} W\otimes S \xrightarrow{\mathbf{C}_{\mathcal{H}\circ_{SP}\mathcal{G}}(\sigma,\tau)} U$$

which is

$$W\otimes R \xrightarrow{\Delta_W\otimes R} W\otimes W\otimes R \xrightarrow{W\otimes\mathbf{P}_\mathcal{G}\sigma\otimes R} W\otimes X\otimes R$$
$$\xrightarrow{W\otimes\mathbf{P}_\mathcal{H}\tau\otimes R} W\otimes Y\otimes R \xrightarrow{W\otimes\mathbf{C}_\mathcal{I}\upsilon} W\otimes S \xrightarrow{\Delta_W\otimes S} W\otimes W\otimes S$$

$$\xrightarrow{W\otimes\mathbf{P}_{\mathcal{G}}\sigma\otimes S} W\otimes X\otimes S \xrightarrow{W\otimes\mathbf{C}_{\mathcal{H}}\tau} W\otimes T \xrightarrow{\mathbf{C}_{\mathcal{G}}\sigma} U$$

On the other hand $\mathbf{C}_{(\mathcal{I}\circ_{SP}\mathcal{H})\circ_{SP}\mathcal{G}}(i((\sigma,\tau),\upsilon))$ is

$$W\otimes R \xrightarrow{\Delta_W\otimes R} W\otimes W\otimes R \xrightarrow{W\otimes\mathbf{P}_{\mathcal{G}}\sigma\otimes R} W\otimes X\otimes R$$
$$\xrightarrow{W\otimes\mathbf{C}_{\mathcal{I}\circ_{SP}\mathcal{H}}(\tau,\upsilon)} W\otimes T \xrightarrow{\mathbf{C}_{\mathcal{G}}\sigma} U$$

which is

$$W\otimes R \xrightarrow{\Delta_W\otimes R} W\otimes W\otimes R \xrightarrow{W\otimes\mathbf{P}_{\mathcal{G}}\sigma\otimes R} W\otimes X\otimes R$$
$$\xrightarrow{W\otimes\Delta_X\otimes R} W\otimes X\otimes X\otimes R \xrightarrow{W\otimes X\otimes\mathbf{P}_{\mathcal{H}}\tau\otimes R} W\otimes X\otimes Y\otimes R$$
$$\xrightarrow{W\otimes X\otimes\mathbf{C}_{\mathcal{I}}\upsilon} W\otimes X\otimes S \xrightarrow{W\otimes\mathbf{C}_{\mathcal{H}}\tau} W\otimes T \xrightarrow{\mathbf{C}_{\mathcal{I}}\sigma} U$$

and these two morphisms are equal by the comonoid laws for Δ.

For best response, we have

$$\begin{aligned}
& ((\sigma',\tau'),\upsilon') \in \mathbf{B}_{\mathcal{I}\circ_{SP}(\mathcal{H}\circ_{SP}\mathcal{G})}(h,k)((\sigma,\tau),\upsilon) \\
\iff & ((\sigma',\tau'),\upsilon') \in \mathbf{B}_{\mathcal{H}\circ_{SP}\mathcal{G}}(h,k_{\upsilon\circ})(\sigma,\tau) \\
& \times \bigcap_{(\sigma'',\tau''):\Sigma_{\mathcal{H}\circ_{SP}\mathcal{G}}} \mathbf{B}_{\mathcal{I}}(\mathbf{P}_{\mathcal{H}\circ_{SP}\mathcal{G}}(\sigma'',\tau'')\circ h,k)\upsilon \\
\iff & (\sigma',\tau',\upsilon') \in \mathbf{B}_{\mathcal{G}}(h,(k_{\upsilon\circ})_{\tau\circ})\sigma \\
& \times \bigcap_{\sigma'':\Sigma_{\mathcal{G}}} \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma''\circ h,k_{\upsilon\circ})\tau \times \bigcap_{\substack{\sigma'':\Sigma_{\mathcal{G}}\\ \tau'':\Sigma_{\mathcal{H}}}} \mathbf{B}_{\mathcal{I}}(\mathbf{P}_{\mathcal{H}}\tau''\circ\mathbf{P}_{\mathcal{G}}\sigma''\circ h,k)\upsilon
\end{aligned}$$

and

$$\begin{aligned}
& i((\sigma',\tau'),\upsilon') \in \mathbf{B}_{(\mathcal{I}\circ_{SP}\mathcal{H})\circ_{SP}\mathcal{G}}(h,k)(i((\sigma,\tau),\upsilon)) \\
\iff & (\sigma',(\tau',\upsilon')) \in \mathbf{B}_{\mathcal{G}}(h,k_{(\tau,\upsilon)\circ})\sigma \times \bigcap_{\sigma'':\Sigma_{\mathcal{G}}} \mathbf{B}_{\mathcal{I}\circ_{SP}\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma''\circ h,k)(\tau,\upsilon) \\
\iff & (\sigma',(\tau',\upsilon')) \in \mathbf{B}_{\mathcal{G}}(h,k_{(\tau,\upsilon)\circ})\sigma \\
& \times \bigcap_{\sigma'':\Sigma_{\mathcal{G}}} \left(\mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma''\circ h,k_{\upsilon\circ})\tau \times \bigcap_{\tau'':\Sigma_{\mathcal{H}}} \mathbf{B}_{\mathcal{I}}(\mathbf{P}_{\mathcal{H}}\tau''\circ\mathbf{P}_{\mathcal{G}}\sigma''\circ h,k)\upsilon \right) \\
\iff & (\sigma',\tau',\upsilon') \in \mathbf{B}_{\mathcal{G}}(h,k_{(\tau,\upsilon)\circ})\sigma \\
& \times \bigcap_{\sigma'':\Sigma_{\mathcal{G}}} \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma''\circ h,k_{\upsilon\circ})\tau \times \bigcap_{\substack{\sigma'':\Sigma_{\mathcal{G}}\\ \tau'':\Sigma_{\mathcal{H}}}} \mathbf{B}_{\mathcal{I}}(\mathbf{P}_{\mathcal{H}}\tau''\circ\mathbf{P}_{\mathcal{G}}\sigma''\circ h,k)\upsilon
\end{aligned}$$

Here $k_{\upsilon\circ}$ is the composition

$$Y \xrightarrow{\Delta_Y} Y\otimes Y \xrightarrow{Y\otimes\mathbf{P}_{\mathcal{I}}\upsilon} Y\otimes Z \xrightarrow{Y\otimes k} Y\otimes R \xrightarrow{\mathbf{C}_{\mathcal{I}}\upsilon} S$$

and $(k_{\upsilon\circ})_{\tau\circ}$ is the composition

$$X \xrightarrow{\Delta_X} X\otimes X \xrightarrow{X\otimes\mathbf{P}_{\mathcal{H}}\tau} X\otimes Y \xrightarrow{X\otimes k_{\upsilon\circ}} X\otimes S \xrightarrow{\mathbf{C}_{\mathcal{H}}\tau} T$$

which expands to

$$X \xrightarrow{\Delta_X} X \otimes X \xrightarrow{X \otimes \mathbf{P}_{\mathcal{H}}\tau} X \otimes Y \xrightarrow{X \otimes \Delta_Y} X \otimes Y \otimes Y$$
$$\xrightarrow{X \otimes Y \otimes \mathbf{P}_{\mathcal{I}}\upsilon} X \otimes Y \otimes Z \xrightarrow{X \otimes Y \otimes k} X \otimes Y \otimes R \xrightarrow{X \otimes \mathbf{C}_{\mathcal{I}}\upsilon} X \otimes S \xrightarrow{\mathbf{C}_{\mathcal{H}}\tau} T$$

On the other hand $k_{(\tau,\upsilon)\circ}$ is the composition

$$X \xrightarrow{\Delta_X} X \otimes X \xrightarrow{X \otimes \mathbf{P}_{\mathcal{I} \circ \mathcal{H}}(\tau,\upsilon)} X \otimes Z \xrightarrow{X \otimes k} X \otimes R \xrightarrow{\mathbf{C}_{\mathcal{I} \circ \mathcal{H}}(\tau,\upsilon)} T$$

which expands to

$$X \xrightarrow{\Delta_X} X \otimes X \xrightarrow{X \otimes \mathbf{P}_{\mathcal{H}}\tau} X \otimes Y \xrightarrow{X \otimes \mathbf{P}_{\mathcal{I}}\upsilon} X \otimes Z \xrightarrow{X \otimes k} X \otimes R$$
$$\xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{\mathcal{H}}\tau \otimes R} X \otimes Y \otimes R \xrightarrow{\mathbf{P}_{\mathcal{I}}\upsilon} X \otimes S \xrightarrow{\mathbf{P}_{\mathcal{H}}\tau} T$$

Then $(k_{\upsilon\circ})_{\tau\circ} = k_{(\tau,\upsilon)\circ}$, and we are done.

A.4 Functoriality of the tensor product

We will now prove that $\otimes$, defined in §2.2.7, is individually functorial on $\mathbf{Game}_{SP}(\mathcal{C})$ in each of its two arguments, that is to say, it is a premonoidal product.

The action on objects of $\mathbf{Game}_{SP}(\mathcal{C})$ is to pairwise apply the monoidal product of $\mathcal{C}$, so

$$(X_1, R_1) \otimes (X_2, R_2) = (X_1 \otimes X_2, R_1 \otimes R_2)$$

The unit is (I, I), where I is the monoidal unit of $\mathcal{C}$.

We will first prove the identity law, namely

$$\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)} \sim \mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}$$

Since $\Sigma_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}} = 1 \times 1$ and $\Sigma_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}} = 1$, we take the isomorphism $i(*,*) = *$. For the play function,

$$\begin{aligned}
\mathbf{P}_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}}(*,*) &= \mathbf{P}_{\mathrm{id}_{(X_1,R_1)}} * \otimes \mathbf{P}_{\mathrm{id}_{(X_2,R_2)}} * \\
&= \mathrm{id}_{X_1} \otimes \mathrm{id}_{X_2} \\
&= \mathrm{id}_{X_1 \otimes X_2} \\
&= \mathbf{P}_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}} *
\end{aligned}$$

For coplay,

$$\begin{aligned}
&\mathbf{C}_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}}(*,*) \\
&= (\mathbf{C}_{\mathrm{id}_{(X_1,R_1)}} * \otimes \mathbf{C}_{\mathrm{id}_{(X_2,R_2)}} *) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= ((\mathrm{id}_{R_1} \circ \pi_{X_1 \otimes X_2 \to R_1}) \otimes (\mathrm{id}_{R_2} \circ \pi_{X_2 \otimes R_2 \to R_2})) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= (\mathrm{id}_{R_1} \otimes \mathrm{id}_{R_2}) \circ (\pi_{X_1 \otimes R_1 \to R_1} \otimes \pi_{X_2 \otimes R_2 \to R_2}) \circ (X_1 \otimes s_{X_2,R_1} \otimes R_2) \\
&= \mathrm{id}_{R_1 \otimes R_2} \circ \pi_{X_1 \otimes X_2 \otimes R_1 \otimes R_2 \to R_1 \otimes R_2} \\
&= \mathbf{C}_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}} *
\end{aligned}$$

where the canonical projections have been labelled with their types for clarity. For best response, we note that

$$* \in \mathbf{B}_{\mathrm{id}_{(X_1 \otimes X_2, R_1 \otimes R_2)}}(h,k)*$$

always holds, and so does

$$\begin{aligned}(*,*) &\in \mathbf{B}_{\mathrm{id}_{(X_1,R_1)} \otimes \mathrm{id}_{(X_2,R_2)}}(h,k)(*,*)\\ &= \mathbf{B}_{\mathrm{id}_{(X_1,R_1)}}(h_1, k_{\otimes *(h_2)})* \times \mathbf{B}_{\mathrm{id}_{(X_2,R_2)}}(h_2, k_{*(h_1)\otimes})*\end{aligned}$$

The distributivity laws for a premonoidal product are

$$(\mathcal{H} \circ_{SP} \mathcal{G}) \otimes \mathrm{id}_{(X',R')} \sim (\mathcal{H} \otimes \mathrm{id}_{(X',R')}) \circ_{SP} (\mathcal{G} \otimes \mathrm{id}_{(X',R')})$$

and

$$\mathrm{id}_{(X',R')} \otimes (\mathcal{H} \circ_{SP} \mathcal{G}) \sim (\mathrm{id}_{(X',R')} \otimes \mathcal{H}) \circ_{SP} (\mathrm{id}_{(X',R')} \otimes \mathcal{G})$$

Because the definition of $\otimes$ is completely symmetrical, we will prove the case for best response only for the first.

For the strategy profiles we have

$$\Sigma_{(\mathcal{H} \circ_{SP} \mathcal{G}) \otimes \mathrm{id}_{(X',R')}} = (\Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{H}}) \times 1$$

and

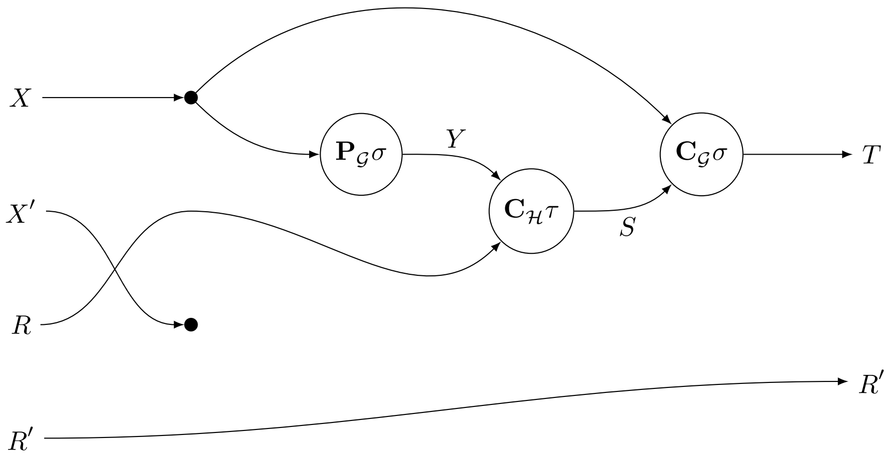
$$\Sigma_{(\mathcal{H} \otimes \mathrm{id}_{(X',R')}) \circ_{SP} (\mathcal{G} \otimes \mathrm{id}_{(X',R')})} = (\Sigma_{\mathcal{G}} \times 1) \times (\Sigma_{\mathcal{H}} \times 1)$$

and so we take the isomorphism $i((\sigma,\tau),*) = ((\sigma,*),(\tau,*))$.

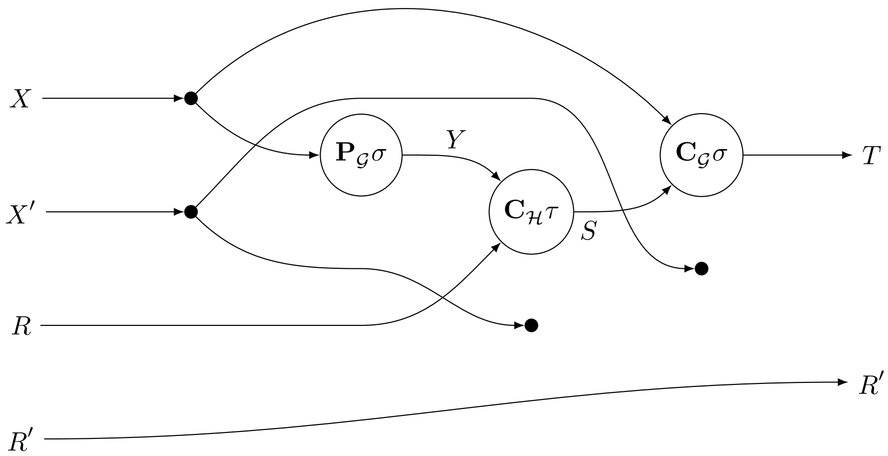
For the play function we have

$$\begin{aligned}\mathbf{P}_{(\mathcal{H} \circ_{SP} \mathcal{G}) \otimes \mathrm{id}_{(X',R')}}((\sigma,\tau),*) &= \mathbf{P}_{\mathcal{H} \circ_{SP} \mathcal{G}}(\sigma,\tau) \otimes \mathbf{P}_{id_{(X',R')}}*\\ &= (\mathbf{P}_{\mathcal{H}}\tau \circ \mathbf{P}_{\mathcal{G}}\sigma) \otimes \mathrm{id}_{X'}\\ &= (\mathbf{P}_{\mathcal{H}}\tau \otimes \mathrm{id}_{X'}) \circ (\mathbf{P}_{\mathcal{G}}\sigma \otimes \mathrm{id}_{X'})\\ &= (\mathbf{P}_{\mathcal{H}}\tau \otimes \mathbf{P}_{\mathrm{id}_{(X',R')}}*) \circ (\mathbf{P}_{\mathcal{G}}\sigma \otimes \mathbf{P}_{\mathrm{id}_{(X',R')}}*)\\ &= \mathbf{P}_{\mathcal{H} \otimes \mathrm{id}_{(X',R')}}(\tau,*) \circ \mathbf{P}_{\mathcal{G} \otimes \mathrm{id}_{(X',R')}}(\sigma,*)\\ &= \mathbf{P}_{(\mathcal{H} \otimes \mathrm{id}_{(X',R')}) \circ_{SP} (\mathcal{G} \otimes \mathrm{id}_{(X',R')})}((\sigma,*),(\tau,*))\end{aligned}$$

For the coplay functions, we draw string diagrams in $\mathcal{C}$ again, as we did in §2.2.8. The string diagram representation of $\mathbf{C}_{(\mathcal{H} \circ_{SP} \mathcal{G}) \otimes \mathrm{id}_{(X',R')}}((\sigma,\tau),*)$ is



and that for $\mathbf{C}_{(\mathcal{H}\otimes\mathrm{id}_{(X',R')})\circ_{SP}(\mathcal{G}\otimes\mathrm{id}_{(X',R')})}$ is



It can be seen that each can be deformed into the other.

Finally, we come to the best response functions. We must show that

$$((\sigma',\tau'),*)\in\mathbf{B}_{(\mathcal{H}\circ_{SP}\mathcal{G})\otimes\mathrm{id}_{(X',R')}}(h,k)((\sigma,\tau),*)$$

is equivalent to

$$((\sigma',*),(\tau',*))\in\mathbf{B}_{(\mathcal{H}\otimes\mathrm{id}_{(X',R')})\circ_{SP}(\mathcal{G}\otimes\mathrm{id}_{(X',R')})}(h,k)((\sigma,*),(\tau,*))$$

The former expands as

$$\begin{aligned}
&((\sigma',\tau'),*)\in\mathbf{B}_{(\mathcal{H}\circ_{SP}\mathcal{G})\otimes\mathrm{id}_{(X',R')}}(h,k)((\sigma,\tau),*)\\
\iff\ &((\sigma',\tau'),*)\in\mathbf{B}_{\mathcal{H}\circ_{SP}\mathcal{G}}(h,k_{\otimes*(h_2)})(\sigma,\tau)\times\mathbf{B}_{\mathrm{id}_{(X',R')}}(h_2,k_{(\sigma,\tau)(h_1)\otimes})*\\
\iff\ &(\sigma',\tau')\in\mathbf{B}_{\mathcal{G}}(h_1,(k_{\otimes*(h_2)})_{\tau\circ})\sigma\times\bigcap_{\sigma'':\Sigma_{\mathcal{G}}}\mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}}\sigma''\circ h_1,k_{\otimes*(h_2)})\tau
\end{aligned}$$

and the latter as

$$\begin{aligned}
&((\sigma', *), (\tau', *)) \in \mathbf{B}_{(\mathcal{H}\otimes\mathrm{id}_{(X',R')})\circ_{SP}(\mathcal{G}\otimes\mathrm{id}_{(X',R')})}(h,k)((\sigma,*),(\tau,*)) \\
\iff\ &((\sigma', *), (\tau', *)) \in \mathbf{B}_{\mathcal{G}\otimes\mathrm{id}_{(X',R')}}(h, k_{(\tau,*)\circ})(\sigma,*) \\
&\times \bigcap_{\substack{(\sigma'',*):\\ \Sigma_{\mathcal{G}}\times 1}} \mathbf{B}_{\mathcal{H}\otimes\mathrm{id}_{(X',R')}}(\mathbf{P}_{\mathcal{G}\otimes\mathrm{id}_{(X',R')}}(\sigma'',*)\circ h, k)(\tau,*) \\
\iff\ &(\sigma',\tau') \in \mathbf{B}_{\mathcal{G}}(h_1, (k_{(\tau,*)\circ})_{\otimes *(h_2)})\sigma \\
&\times \bigcap_{\sigma'':\Sigma_{\mathcal{G}}} \mathbf{B}_{\mathcal{H}}((\mathbf{P}_{\mathcal{G}\otimes\mathrm{id}_{(X',R')}}(\sigma'',*)\circ h)_1, k_{\otimes *((\mathbf{P}_{\mathcal{G}\otimes\mathrm{id}_{(X',R')}}(\sigma'',*)\circ h)_2)})\tau
\end{aligned}$$

Comparing these, we first note that we have equalities of histories

$$\mathbf{P}_{\mathcal{G}}\sigma''\circ h_1 = (\mathbf{P}_{\mathcal{G}\otimes\mathrm{id}_{(X',R')}}(\sigma'',*)\circ h)_1$$

and

$$h_2 = (\mathbf{P}_{\mathcal{G}\otimes\mathrm{id}_{(X',R')}}(\sigma'',*)\circ h)_2$$

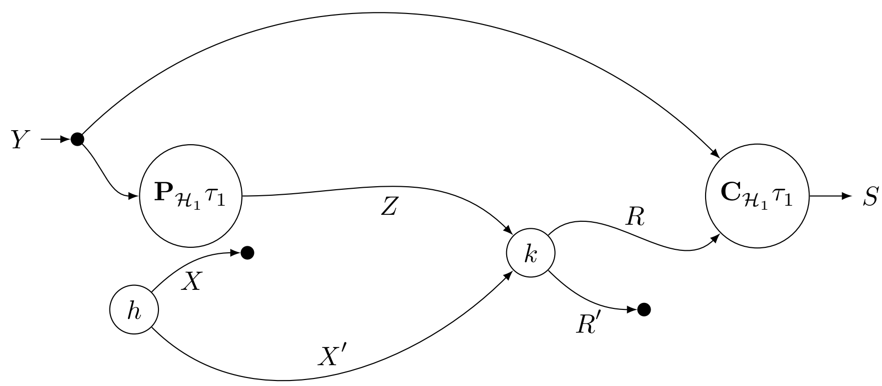
The equality of continuations

$$k_{\otimes *(h_2)} = k_{\otimes *((\mathbf{P}_{\mathcal{G}\otimes\mathrm{id}_{(X',R')}}(\sigma'',*)\circ h)_2)}$$

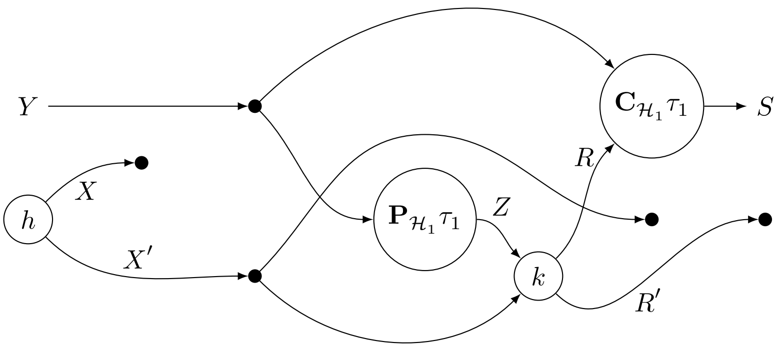
follows immediately from the latter. For the remaining equality

$$(k_{\otimes *(h_2)})_{\tau\circ} = (k_{(\tau,*)\circ})_{\otimes *(h_2)}$$

we note the equivalence between the diagrams



and



A.5 The monoidal category axioms

The remaining work in proving that $\mathbf{Game}_{SP}(\mathcal{C})$ is premonoidal is to prove the monoidal category axioms. (To be clear, the axioms of a premonoidal category are still usually called the 'monoidal category axioms' because they are identical to those of a monoidal category; the only difference in a premonoidal category is that $\otimes$ is not a bifunctor.)

In general, proving these axioms takes a significant amount of work. We must define three families of morphisms, the left and right unitors and the associators, prove their naturality, and then prove the commutativity of two diagrams including the Mac Lane pentagon. To prove that a premonoidal category is symmetric we must additionally define the braiding morphisms, prove their naturality, and prove commutativity of an additional three diagrams.

Most of this work can be avoided by appealing to Mac Lane's coherence theorem [Mac78] and replacing $\mathcal{C}$ with a monoidally equivalent strict monoidal category. In that case we have equalities of objects

$$(I, I) \otimes (X, R) = (I \otimes X, I \otimes R) = (X, R) = (X \otimes I, R \otimes I) = (X, R) \otimes (I, I)$$

and so we can take all of the unitors to be the identity morphisms (that is, computations formed of pairs of identities, see §2.2.3), which are automatically natural and satisfy the commutative diagrams, simply by the fact that $\mathbf{Game}_{SP}(\mathcal{C})$ is a category. Similarly we have equalities

$$\begin{aligned}((X_1, R_1) \otimes (X_2, R_2)) \otimes (X_3, R_3) &= (X_1 \otimes X_2 \otimes X_3, R_1 \otimes R_2 \otimes R_3) \\ &= (X_1, R_1) \otimes ((X_2, R_2) \otimes (X_3, R_3))\end{aligned}$$

and so we can also take the associators to be identities.

For the braiding morphisms we take (the equivalence class of) the computation

$$\begin{gathered} s_{(X_1,R_1),(X_2,R_2)} = (s_{X_1,X_2}, s_{R_2,R_1}) \\ : (X_1 \otimes X_2, R_1 \otimes R_2) \to (X_2 \otimes X_1, R_2 \otimes R_1) \end{gathered}$$

For a strict premonoidal category the the unit law becomes trivial, so we must prove only the associativity and inverse laws. For this, we will use the result from §A.7, that computations respect SP-composition and tensor. This is not circular, because we will only use the part of the result that does not already assume that $\mathbf{Game}_{SP}(\mathcal{C})$ is premonoidal, and is really shorthand for copying special cases of that proof into this section.

We will begin with the inverse law. For an arbitrary symmetric premonoidal category this is

$$s_{B,A} \circ s_{A,B} = \mathrm{id}_{A \otimes B}$$

We take $A = (X_1, R_1)$ and $B = (X_2, R_2)$, and so this is

$$(s_{X_2,X_1}, s_{R_1,R_2}) \circ_{SP} (s_{X_1,X_2}, s_{R_2,R_1})$$
$$: (X_1 \otimes X_2, R_1 \otimes R_2) \to (X_1 \otimes X_2, R_1 \otimes R_2)$$

Since computation is functorial (§A.7), this is

$$(s_{X_2,X_1} \circ s_{X_1,X_2}, s_{R_2,R_1} \circ s_{R_1,R_2})$$

and we can apply the inverse law of $\mathcal{C}$.

The associativity axiom for a strict symmetric premonoidal category is

$$\begin{tikzcd}
A \otimes B \otimes C \arrow[rr, "s_{A,B} \otimes C"] \arrow[d, "s_{A,B\otimes C}"'] & & B \otimes A \otimes C \arrow[dll, "B \otimes s_{A,C}"] \\
B \otimes C \otimes A & &
\end{tikzcd}$$

In $\mathbf{Game}_{SP}(\mathcal{C})$, we need to take $A = (X_1, R_1)$, $B = (X_2, R_2)$ and $C = (X_3, R_3)$.

As a lemma, we need the equations

$$(X_2, R_2) \otimes s_{(X_1,R_1),(X_3,R_3)} = (X_2 \otimes s_{X_1,X_3}, R_2 \otimes s_{R_3,R_1})$$

and

$$s_{(X_1,R_1),(X_2,R_2)} \otimes (X_3, R_3) = (s_{X_1,X_2} \otimes X_3, s_{R_2,R_1} \otimes R_3)$$

which are special cases of the fact that computation is a monoidal functor (§A.7).

Now, by functoriality, we have that the computations

$$((X_2, R_2) \otimes s_{(X_1,R_1),(X_3,R_3)}) \circ_{SP} (s_{(X_1,R_1),(X_2,R_2)} \otimes (X_3, R_3))$$

and

$$((X_2 \otimes s_{X_1,X_3}) \circ (s_{X_1,X_2} \otimes X_3), (s_{R_2,R_1} \otimes R_3) \circ (R_2 \otimes s_{R_3,R_1}))$$

are equal. Therefore we need only check the equations

$$(X_2 \otimes s_{X_1,X_3}) \circ (s_{X_1,X_2} \otimes X_3) = s_{X_1,X_2 \otimes X_3}$$

and

$$(s_{R_2,R_1} \otimes R_3) \circ (R_2 \otimes s_{R_3,R_1}) = s_{R_2 \otimes R_3,R_1}$$

in $\mathcal{C}$, which both hold because $\mathcal{C}$ is symmetric monoidal.

A.6 Strategic triviality

The definition of strategic triviality was given in §2.2.11. We will now prove that strategically trivial games are closed under SP-composition.

Suppose we have strategically trivial games

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

We must prove that $\mathcal{H} \circ_{SP} \mathcal{G}$ has one strategy, which behaves trivially. The strategy profiles are

$$\Sigma_{\mathcal{H} \circ_{SP} \mathcal{G}} = \Sigma_{\mathcal{G}} \times \Sigma_{\mathcal{G}} = 1 \times 1$$

The best response function is

$$\begin{aligned}\mathbf{B}_{\mathcal{H} \circ_{SP} \mathcal{G}}(h, k)(*, *) &= \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})* \times \bigcap_{*:1} \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}} * \circ h, k)* \\ &= \{*\} \times \{*\} = \{(*, *)\}\end{aligned}$$

We also note the equivalent for SP-composition of the useful result from §2.2.11, namely that if we have a composition

$$(X', S') \xrightarrow{\mathcal{H}_1} (X, S) \xrightarrow{\mathcal{G}} (Y, R) \xrightarrow{\mathcal{H}_2} (Y', R')$$

in $\mathbf{Game}_{SP}(\mathcal{C})$, where $\mathcal{H}_1$ and $\mathcal{H}_2$ are strategically trivial, then

$$\Sigma_{\mathcal{H}_2 \circ_{SP} \mathcal{G} \circ_{SP} \mathcal{H}_1} \cong \Sigma_{\mathcal{G}}$$

and

$$\mathbf{B}_{\mathcal{H}_2 \circ_{SP} \mathcal{G} \circ_{SP} \mathcal{H}_1}(h, k) = \mathbf{B}_{\mathcal{G}}(\mathbf{P}_{\mathcal{H}_1} \circ h, k')$$

where

$$k' : Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes \mathbf{P}_{\mathcal{H}_2}} Y \otimes Y' \xrightarrow{Y \otimes k} Y \otimes R' \xrightarrow{\mathbf{C}_{\mathcal{H}_2}} R$$

As before, $\mathcal{H}_1$ affects only the history, and $\mathcal{H}_2$ affects only the continuation.

It is also useful to note that if either $\mathcal{G}$ or $\mathcal{H}$ is strategically trivial, then $\mathcal{H} \circ_{SP} \mathcal{G} = \mathcal{H} \circ_N \mathcal{G}$. This is extremely useful because it will quite often be the case that one of the games being composed is strategically trivial. The two cases have different proofs. If $\mathcal{G}$ is strategically trivial the result holds because the intersection over the strategies of $\mathcal{G}$ is trivial:

$$\begin{aligned}\mathbf{B}_{\mathcal{H} \circ_{SP} \mathcal{G}}(h, k)(*, \sigma) &= \mathbf{B}_{\mathcal{G}}(h, k_{\sigma\circ})* \times \bigcap_{*:1} \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}} * \circ h, k)\sigma \\ &= \mathbf{B}_{\mathcal{G}}(h, k_{\sigma\circ})* \times \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}} * \circ h, k)\sigma \\ &= \mathbf{B}_{\mathcal{H} \circ_N \mathcal{G}}(h, k)(*, \sigma)\end{aligned}$$

On the other hand, if $\mathcal{H}$ is strategically trivial then

$$\begin{aligned}\mathbf{B}_{\mathcal{H} \circ_{SP} \mathcal{G}}(h, k)(\sigma, *) &= \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \times \bigcap_{\sigma' : \Sigma_{\mathcal{G}}} \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}} \sigma' \circ h, k)* \\ &= \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \times \bigcap_{\sigma' : \Sigma_{\mathcal{G}}} \{*\} \\ &= \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \times \{*\} \\ &= \mathbf{B}_{\mathcal{G}}(h, k_{*\circ})\sigma \times \mathbf{B}_{\mathcal{H}}(\mathbf{P}_{\mathcal{G}} \sigma \circ h, k)* \\ &= \mathbf{B}_{\mathcal{H} \circ_N \mathcal{G}}(h, k)(\sigma, *)\end{aligned}$$

A.7 Computations as a monoidal functor

We will now prove that computation, defined in §2.1.9, gives us a faithful monoidal functor

$$(-,-) : \mathcal{C} \times \mathcal{C}^{\mathrm{op}} \hookrightarrow \mathbf{Game}_{SP}(\mathcal{C})$$

We will first prove that we have a bifunctor $\mathcal{C} \times \mathcal{C}^{\mathrm{op}} \to \mathbf{Game}_{SP}(\mathcal{C})$. In the product category $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ the objects are pairs of objects of $\mathcal{C}$, and the morphisms are pairs of morphisms with the second reversed. The identity morphism on the object (X, R) of $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ is $(\mathrm{id}_X, \mathrm{id}_R)$ which, lifted to a computation, is also the the identity game on (X, R) in $\mathbf{Game}_{SP}(\mathcal{C})$ (see §A.2).

The composition

$$(X, T) \xrightarrow{(f_1, f_2)} (Y, S) \xrightarrow{(g_1, g_2)} (Z, R)$$

in $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ is, by definition,

$$(g_1, g_2) \circ (f_1, f_2) = (g_1 \circ f_1, f_2 \circ g_2)$$

and we must prove that the games denoted by these two expressions are equal. Since both are strategically trivial (§A.6), we need only check the play and coplay functions. The play functions are

$$\mathbf{P}_{(g_1 \circ f_1, f_2 \circ g_2)} : X \xrightarrow{g_1 \circ f_1} Z$$

and

$$\mathbf{P}_{(g_1, g_2) \circ_{SP} (f_1, f_2)} : X \xrightarrow{\mathbf{P}_{(f_1, f_2)}} Y \xrightarrow{\mathbf{P}_{(g_1, g_2)}} Z$$

which are equal. The coplay functions are

$$\mathbf{P}_{(g_1 \circ f_1, f_2 \circ g_2)} : X \otimes R \xrightarrow{\pi_2} R \xrightarrow{f_2 \circ g_2} T$$

and

$$\mathbf{C}_{(g_1, g_2) \circ_{SP} (f_1, f_2)} : X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes \mathbf{P}_{(f_1, f_2)} \otimes R} X \otimes Y \otimes R$$
$$\xrightarrow{X \otimes \mathbf{C}_{(g_1, g_2)}} X \otimes S \xrightarrow{\mathbf{C}_{(f_1, f_2)}} T$$

The latter is

$$X \otimes R \xrightarrow{\Delta_X \otimes R} X \otimes X \otimes R \xrightarrow{X \otimes f_1 \otimes R} X \otimes Y \otimes R$$
$$\xrightarrow{X \otimes \pi_2} X \otimes R \xrightarrow{X \otimes g_2} X \otimes S \xrightarrow{\pi_2} S \xrightarrow{f_2} T$$

and these are equal.

Next, we must prove that the embedding also respects the monoidal structure. The monoidal unit of the product monoidal category $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ is (I, I), which is also the monoidal unit of $\mathbf{Game}_{SP}(\mathcal{C})$ (see §A.5).

Suppose $(f_1, f_2) : (X_1, S_1) \to (Y_1, R_1)$ and $(g_1, g_2) : (X_2, S_2) \to (Y_2, R_2)$ are morphisms of $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$. Their monoidal product is

$$(f_1 \otimes g_1, f_2 \otimes g_2) : (X_1 \otimes X_2, S_1 \otimes S_2) \to (Y_1 \otimes Y_2, R_1 \otimes R_2)$$

We must therefore prove the equality of games

$$(f_1, f_2) \otimes (g_1, g_2) = (f_1 \otimes g_1, f_2 \otimes g_2)$$

Again both are strategically trivial, so we need only work with the play and coplay functions. For the play function we have

$$\mathbf{P}_{(f_1,f_2)\otimes(g_1,g_2)} = \mathbf{P}_{(f_1,f_2)} \otimes \mathbf{P}_{(g_1,g_2)} = f_1 \otimes g_1 = \mathbf{P}_{(f_1\otimes g_1, f_2\otimes g_2)}$$

For coplay, the former is

$$\mathbf{C}_{(f_1,f_2)\otimes(g_1,g_2)} : X_1 \otimes X_2 \otimes R_1 \otimes R_2 \xrightarrow{\cong} X_1 \otimes R_1 \otimes X_2 \otimes R_2$$
$$\xrightarrow{\mathbf{C}_{(f_1,f_2)}\otimes \mathbf{C}_{(g_1,g_2)}} S_1 \otimes S_2$$

which is

$$X_1 \otimes X_2 \otimes R_1 \otimes R_2 \xrightarrow{\cong} X_1 \otimes R_1 \otimes X_2 \otimes R_2$$
$$\xrightarrow{\pi_2 \otimes \pi_2} R_1 \otimes R_2 \xrightarrow{f_2 \otimes g_2} S_1 \otimes S_2$$

and the latter is

$$\mathbf{C}_{(f_1\otimes g_1, f_2\otimes g_2)} : X_1 \otimes X_2 \otimes R_1 \otimes R_2 \xrightarrow{\pi_2} R_1 \otimes R_2 \xrightarrow{f_2 \otimes g_2} S_1 \otimes S_2$$

which is equal.

A.8 The counit law

Let $f : \hom_{\mathcal{C}}(X, Y)$. Then the following diagram commutes in $\mathbf{Game}_{SP}(\mathcal{C})$:

$$\begin{tikzcd}
(X, Y) \arrow[r, "(\mathrm{id}_X, f)"] \arrow[d, "(f, \mathrm{id}_Y)"'] & (X, X) \arrow[d, "\tau_X"] \\
(Y, Y) \arrow[r, "\tau_Y"] & (I, I)
\end{tikzcd}$$

Since both games are strategically trivial, we need only check the behaviour of the play and coplay functions.

The definitions in §2.1.9 give us

$$\mathbf{P}_{(\mathrm{id}_X, f)} : X \xrightarrow{\mathrm{id}_X} X \qquad \mathbf{C}_{(\mathrm{id}_X, f)} : X \otimes X \xrightarrow{\pi_2} X \xrightarrow{f} Y$$
$$\mathbf{P}_{(f, \mathrm{id}_Y)} : X \xrightarrow{f} Y \qquad \mathbf{C}_{(f, \mathrm{id}_Y)} : X \otimes Y \xrightarrow{\pi_2} Y \xrightarrow{\mathrm{id}_Y} Y$$
$$\mathbf{P}_{\varepsilon_X} : X \xrightarrow{!_X} I \qquad \mathbf{C}_{\varepsilon_X} : X \otimes I \xrightarrow{\pi_1} X$$
$$\mathbf{P}_{\varepsilon_Y} : Y \xrightarrow{!_Y} I \qquad \mathbf{C}_{\varepsilon_Y} : Y \otimes I \xrightarrow{\pi_1} Y$$

Composing these sequentially gives us the play functions

$$\mathbf{P}_{\tau_X \circ_{SP} (\mathrm{id}_X, f)} : X \xrightarrow{\mathrm{id}_X} X \xrightarrow{!_X} I$$

$$\mathbf{P}_{\tau_Y \circ_{SP} (f, \mathrm{id}_Y)} : X \xrightarrow{f} Y \xrightarrow{!_Y} I$$

which are equal, and the coplay functions

$$\mathbf{C}_{\varepsilon_X \circ_{SP} (\mathrm{id}_X, f)} : X \otimes I \xrightarrow{\Delta_X \otimes I} X \otimes X \otimes I \xrightarrow{X \otimes \pi_1} X \otimes X \xrightarrow{\pi_2} X \xrightarrow{f} Y$$

and

$$\mathbf{C}_{\varepsilon_Y \circ_{SP} (f, \mathrm{id}_Y)} : X \otimes I \xrightarrow{\Delta_X \otimes I} X \otimes X \otimes I \xrightarrow{X \otimes f \otimes I} X \otimes Y \otimes I$$

$$\xrightarrow{X \otimes \pi_1} X \otimes Y \xrightarrow{\pi_2} Y$$

which are both equal to

$$X \otimes I \xrightarrow{\pi_1} X \xrightarrow{f} Y$$

The game denoted by these two equal expressions is important, because when post-composed with another game it will behave like a continuation, and we will generally use it when f is an outcome function. Let $\mathcal{G} : (X, S) \to (Y, R)$ be a game, and let $k : \mathrm{hom}_{\mathcal{C}}(Y, R)$ be a continuation for $\mathcal{G}$. Consider the game

$$(X, S) \xrightarrow{\mathcal{G}} (Y, R) \xrightarrow{(k, \mathrm{id}_R)} (R, R) \xrightarrow{\varepsilon_R} (I, I)$$

in $\mathbf{Game}_{SP}(\mathcal{C})$, which by the counit law, can be equivalently written

$$(X, S) \xrightarrow{\mathcal{G}} (Y, R) \xrightarrow{(\mathrm{id}_Y, k)} (Y, Y) \xrightarrow{\varepsilon_Y} (I, I)$$

Then

$$\Sigma_{\varepsilon_R \circ_{SP} (k, \mathrm{id}_R) \circ_{SP} \mathcal{G}} \cong \Sigma_{\mathcal{G}}$$

and for any $h : \mathrm{hom}_{\mathcal{C}}(I, X)$ and $\sigma : \Sigma_{\mathcal{G}}$ we have

$$\mathbf{B}_{\mathcal{G}}(h, k)\sigma \cong \mathbf{B}_{\varepsilon_R \circ_{SP} (k, \mathrm{id}_R) \circ_{SP} \mathcal{G}}(h, \mathrm{id}_I)\sigma$$

under the same isomorphism.

To see this, the right hand side is

$$\begin{aligned} &\mathbf{B}_{\varepsilon_R \circ_{SP} (k, \mathrm{id}_R) \circ_{SP} \mathcal{G}}(h, \mathrm{id}_I)\sigma \\ &= \mathbf{B}_{\mathcal{G}}(h, (\mathrm{id}_I)_{*\circ})\sigma \times \bigcap_{\sigma' : \Sigma_{\mathcal{G}}} \mathbf{B}_{\varepsilon_R \circ_{SP} (k, \mathrm{id}_R)}(\mathbf{P}_{\mathcal{G}}\sigma' \circ h, \mathrm{id}_I)* \end{aligned}$$

and, since $\varepsilon_R \circ_{SP} (k, \mathrm{id}_R)$ is strategically trivial, this is

$$\mathbf{B}_{\mathcal{G}}(h, (\mathrm{id}_I)_{*\circ})\sigma \times \{*\}$$

The final step is to see that $(\mathrm{id}_I)_{*\circ} = k$. By definition it is

$$(\mathrm{id}_I)_{*\circ} : Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes \mathbf{P}_{\varepsilon_R \circ_{SP} (k, \mathrm{id}_R)}} Y \otimes I \xrightarrow{Y \otimes \mathrm{id}_I} Y \otimes I \xrightarrow{C_{\varepsilon_R \circ_{SP} (k, \mathrm{id}_R)}} R$$

which reduces to

$$Y \xrightarrow{\Delta_Y} Y \otimes Y \xrightarrow{Y \otimes !_Y} Y \otimes I \xrightarrow{\pi_1} Y \xrightarrow{k} R$$

and hence to k.

Bibliography

[Abr05] Samson Abramsky. Abstract scalars, loops, and free traced and strongly compact closed categories. In *Proceedings of CALCO'05*, volume 3629 of *Lectures notes in computer science*, pages 1–29, 2005.

[Abr15] Samson Abramsky. Arrow's theorem by arrow theory. In *Logic without borders: essays on set theory, model theory, philosophical logic and philosophy of mathematics*, volume 5 of *Ontos mathematical logic*, pages 15–30. De Gruyter, 2015.

[AC04] Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science: LICS 2004*, 2004.

[AF98] Jeremy Avigad and Solomon Feferman. Gödel's functional ("Dialectica") interpretation. In S. Buss, editor, *Handbook of proof theory*, volume 137 of *Studies in logic and the foundations of mathematics*, pages 337–405. North Holland, Amsterdam, 1998.

[AGH+06] Thorsten Altenkirch, Neil Ghani, Peter Hancock, Conor McBride, and Peter Morris. Indexed containers. Unpublished manuscript, 2006.

[AKLW15] Samson Abramsky, Alexander Kurz, Pierre Lescanne, and Viktor Winschel. Coalgebraic semantics of reflexive economics (Dagstuhl seminar 15042). *Dagstuhl reports*, 5(1), 2015.

[Ati08] Michael Atiyah. Advice to a young mathematician. In Timothy Gowers, editor, *The Princeton companion to mathematics*. Princeton university press, 2008.

[AW15] Samson Abramsky and Viktor Winschel. Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. To appear in *Mathematical Structures in Computer Science*, 2015.

[BIB13] Nicola Botta, Cezar Ionescu, and Edwin Brady. Sequential decision problems, dependently typed solutions. In *Proeedings of PLMMS13*, 2013.

[BKP15] Pierre Boudes, Antoine Kaszczyc, and Luc Pellisser. Monetary economics simulation: Stock-flow consistent invariance, monadic style. HAL ID: hal-01181278, 2015.

[Bla97] Andreas Blass. Some semantical aspects of linear logic. *Logic journal of the interest group in pure and applied logic*, 5(4):487–503, 1997.

[BMI+11] Nicola Botta, Antoine Mandel, Cezar Ionescu, Mareen Hofmann, Daniel Lincke, Sibylle Schupp, and Carlo Jaeger. A functional framework for agent-based models of exchange. *Applied mathematics and computation*, 218(8):4025–4040, 2011.

[BS10] John Baez and Mike Stay. Physics, topology, logic and computation: a Rosetta stone. In Bob Coecke, editor, *New structures for physics*, pages 95–172. Springer, 2010.

[BW13] Achim Blumensath and Viktor Winschel. A coalgebraic framework for games in economics. Working paper, 2013.

[Coe05] Bob Coecke. Kindergarten quantum mechanics. ArXiV, 2005.

[Coe06] Bob Coecke. Introducing categories to the practicing physicist. In *What is category theory? Advanced studies in mathematics and logic*, volume 30, pages 45–74. Polimetrica, 2006.

[Coe11] Bob Coecke, editor. *New structures for physics*. Springer, 2011.

[DGP06] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *STOC '06 Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 71–79, 2006.

[dP91] Valeria de Paiva. The dialectica categories. Technical report, University of Cambridge, 1991.

[dP07] Valeria de Paiva. Dialectica and Chu construtions: Cousins? *Theory and applications of categories*, 17(7):127–152, 2007.

[EK06] Martin Erwig and Steve Kollmansberger. Probabilistic functional programming in Haskell. *Journal of functional programming*, 16(1):21–34, 2006.

[EO10a] Martin Escardó and Paulo Oliva. Selection functions, bar recursion and backward induction. *Mathematical structures in computer science*, 20(2):127–168, 2010.

[EO10b] Martin Escardó and Paulo Oliva. What sequential games, the Tychonoff theorem and the double-negation shift have in common. In *Proceedings of MSFP'10*, 2010.

[EO11] Martin Escardó and Paulo Oliva. Sequential games and optimal strategies. *Proceedings of the Royal Society A*, 467:1519–1545, 2011.

[EO12] Martin Escardó and Paulo Oliva. Computing Nash equilibria of unbounded games. *Proceedings of the Turing centenary conference*, 2012.

[EO15] Martin Escardó and Paulo Oliva. The Herbrand interpretation of the double negation shift. Submitted, 2015.

[Esc04] Martin Escardó. Synthetic topology of data types and classical spaces. *ENTCS*, 87:21–156, 2004.

[Fri09] Tobias Fritz. Convex spaces I: Definitions and examples. ArXiV, 2009.

[FT91] Drew Fudenberg and Jean Tirole. *Game theory*. MIT Press, 1991.

[GH16] Neil Ghani and Jules Hedges. A compositional approach to economic game theory. Submitted, 2016.

[Gir82] Michèlle Giry. A categorical approach to probability theory. *Categorical aspects of topology and analysis*, pages 68–85, 1982.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical computer science*, 1987.

[Gir01] Jean-Yves Girard. Locus solum: from the rules of logic to the logic of rules. *Mathematical structures in computer science*, 11(3):301–506, 2001.

[Hed13] Jules Hedges. A generalisation of Nash's theorem with higher-order functionals. *Proceedings of the Royal Society A*, 469, 2013.

[Hed14] Jules Hedges. Monad transformers for backtracking search. In *Proceedings of MSFP'14*, EPTCS, pages 31–50, 2014.

[Hed15a] Jules Hedges. Dialectica categories and games with bidding. In *Post-proceedings of TYPES'14*, volume 39 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 89–110, 2015.

[Hed15b] Jules Hedges. The selection monad as a CPS translation. ArXiV, 2015.

[HOS+15a] Jules Hedges, Paulo Oliva, Evguenia Sprits, Viktor Winschel, and Philipp Zahn. Higher-order decision theory. ArXiV, 2015.

[HOS+15b] Jules Hedges, Paulo Oliva, Evguenia Sprits, Viktor Winschel, and Philipp Zahn. Higher-order game theory. Submitted, 2015.

[Hoy08] Doug Hoyte. *Let over lambda*. lulu.com, 2008.

[HS16] Jules Hedges and Mehrnoosh Sadrzadeh. A generalised quantifier theory of natural language in categorical compositional distributional semantics with bialgebras. Submitted, 2016.

[HSG13] Chris Heunen, Mehrnoosh Sadrzadeh, and Edward Grefenstette, editors. *Quantum physics and linguistics*. Oxford University Press, 2013.

[Hyv14] Pierre Hyvernat. A linear category of polynomial diagrams. *Mathematical structures in computer science*, 24(1), 2014.

[IJ13] Cezar Ionescu and Patrik Jansson. Dependently-typed programming in scientific computing: examples from economic modelling. In *Implementation and application of functional languages*, Lectures notes in computer science, pages 140–156. Springer, 2013.

[Jef97] Alan Jeffrey. Premonoidal categories and a graphical view of programs. Unpublished manuscript, 1997.

[Joh02] Peter Johnstone. *Sketches of an Elephant: a topos theory compendium*, volume 1. Oxford University Press, 2002.

[JS91] André Joyal and Ross Street. The geometry of tensor calculus I. *Advances in mathematics*, 88(1):55–112, 1991.

[JSV96] André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. *Mathematical proceedings of the Cambridge philosophical society*, 119:447–468, 1996.

[JWW15] Bart Jacobs, Bas Westerbaan, and Bram Westerbaan. States of convex sets. In *Proceedings of FoSSaCS'15*, Lectures notes in computer science, pages 87–101, 2015.

[Kak41] Shizuo Kakutani. A generalisation of brouwer's fixed point theorem. *Duke Math. J.*, 8(3):457–459, 1941.

[Key36] John Maynard Keynes. *The general theory of employment, interest and money.* Macmillan, 1936.

[KL80] G. M. Kelly and M. L. Laplaza. Coherence for compact closed categories. *Journal of pure and applied algebra*, 19(193–213), 1980.

[KM03] Daphne Koller and Brian Milch. Multi-agent influence diagrams for representing and solving games. *Games and economic behaviour*, 45(1):181–221, 2003.

[KMS14] Aleks Kissinger, Alex Merry, and Matvey Soloviev. Pattern graph rewrite systems. *Proceedings of DCM'12*, 143:54–66, 2014.

[Koc71] Anders Kock. On double dualisation monads. *Mathematica Scandinavica*, 27(2):151–165, 1971.

[Koc72] Anders Kock. Strong functors and monoidal monads. *Archiv der Mathematik*, 23(1):113–120, 1972.

[Kre90] David Kreps. *A course in microeconomic theory.* Princeton university press, 1990.

[Kre91] David Kreps. *Game theory and economic modelling.* Clarendon lectures in economics. Oxford University Press, 1991.

[LaV06] Steven M. LaValle. *Planning algorithms.* Cambridge University Press, 2006.

[LBS08] Kevin Leyton-Brown and Yoav Shoham. *Essentials of game theory: a concice, multidisciplinary introduction.* Morgan and Claypool, 2008.

[LP12] Pierre Lescanne and Matthieu Perrinel. "Backward" coinduction, Nash equilibrium and the rationality of escalation. *Acta Informatica*, 49(3):117–137, 2012.

[lR14] Stephane le Roux. From winning strategy to Nash equilibrium. *Mathematical logic quarterly*, 60(4–5):354–371, 2014.

[lRP14] Stéphane le Roux and Arno Pauly. Infinite sequential games with real-valued payoffs. In *Proceedings of CSL-LICS'14*, pages 62:1–62:10, 2014.

[Mac78] Saunders Mac Lane. *Categories for the working mathematician.* Springer, 1978.

[McC12] Micah McCurdy. Graphical methods for Tannaka duality of weak bialgebras and weak Hopf algebras. *Theory and applications of categories*, 26(9):233–280, 2012.

[MI04] Lawrence Moss and Ignacio Iglizzo. Harsanyi type spaces and final coalgebras constructed from satisfied theories. In *Proceedings of CMCS'04*, volume 106 of *ENTCS*, pages 279–295, 2004.

[Mog89] Eugenio Moggi. Computational lambda calculus and monads. In *Proceedings of fourth annual symposium on logic in computer science*, 1989.

[Mog91] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[Mos57] A. Mostowski. On a generalisation of quantifiers. *Fundamenta mathematicae*, 44:12–36, 1957.

[Nas50] John Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[Nas51] John Nash. Non-cooperative games. *Annals of mathematics*, 54(2):286–295, 1951.

[NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay Vazirani, editors. *Algorithmic game theory.* Cambridge University Press, 2007.

[OP14] Paulo Oliva and Thomas Powell. A constructive interpretation of Ramsey's theorem via the product of selecion functions. *Mathematical structures in computer science*, 2014.

[OP15] Paulo Oliva and Thomas Powell. A game-theoretic computational interpretation of proofs in classical analysis. In *Gentzen's centenary: The quest for consistency*, pages 501–531. Springer, 2015.

[Pan99] Prakash Panangaden. The category of Markov kernels. *Proceedings of the first international workshop on probabilistic methods in verification*, 22:171–187, 1999.

[Pat01] Ross Paterson. A new notation for arrows. In *Proceedings of the sixth ACM SIGPLAN international conference on functional programming*, pages 229–240. ACM, 2001.

[Pav09] Dusko Pavlovic. A semantical approach to equilibria and rationality. In *Algebra and coalgebra in computer science*, volume 5728 of *Lectures notes in computer science*, pages 317–334. Springer, 2009.

[Pav13] Dusko Pavlovic. Monoidal computer I: Basic computability by string diagrams. *Information and Computation*, 226:94–116, 2013.

[Pen71] Roger Penrose. Applications of negative dimensional tensors. In D. J. A. Welsh, editor, *Combinatorial mathematics and its applications*, pages 221–244. Academic Press, 1971.

[Pow13] Thomas Powell. *On Bar Recursive Interpretations of Analysis*. PhD thesis, Queen Mary University of London, 2013.

[PP02] Gordon Plotkin and John Power. Notions of computation determine monads. In *Proceedings of FoSSaCS'02*, volume 2303 of *Lectures notes in computer science*, pages 342–356. Springer, 2002.

[PR93] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Mathematical structures in computer science*, 1993.

[PT99] John Power and Hayo Thielecke. Closed Freyd and κ-categories. In *Proceedings of the 26th International Colloquium on Automata, Languages and Programming*, pages 625–634. Springer-Verlag, 1999.

[Sea13] Gavin Seal. Tensors, monads and actions. *Theory and applications of categories*, 28(15):403–434, 2013.

[Sel11] Peter Selinger. A survey of graphical languages for monoidal categories. In Bob Coecke, editor, *New structures for physics*, pages 289–355. Springer, 2011.

[Shu10] Mike Shulman. Constructing symmetric monoidal bicategories. ArXiV, 2010.

[SP09] Chris Schommer-Pries. *The classification of two-dimensional extended topological field theories*. PhD thesis, University of California, Berkeley, 2009.

[vNM44] John von Neumann and Oskar Morgenstern. *Theory of games and economic behaviour*. Princeton university press, 1944.

[VW06] Daniele Varacca and Glynn Winskel. Distributing probability over nondeterminism. *Mathematical structures in computer science*, 16(1):87–113, 2006.