

Sequential Games and Optimal Strategies

BY MARTÍN ESCARDÓ AND PAULO OLIVA

University of Birmingham, Birmingham B15 2TT, UK

Queen Mary University of London, London E1 4NS, UK

This article gives an overview of recent work on the theory of selection functions. We explain the intuition behind these higher-type objects, and define a general notion of sequential game whose optimal strategies can be computed via a certain product of selection functions. Several instances of this game are considered in a variety of areas such as fixed point theory, topology, game theory, higher-type computability, and proof theory. These examples are intended to illustrate how the fundamental construction of optimal strategies based on products of selection functions permeates several research areas.

Keywords: Selection function, Nash equilibrium, backward induction, backtracking, bar recursion, Bekić’s lemma, Tychonoff theorem

1. Introduction

Life is the sum of all your choices, so said Albert Camus. But what does “choice” mean? One could say that to choose is to select one element x out of a set X of possible candidates. The set X could be the collection of all the petrol stations on the way to work, or the different types of fuel available at a particular station. Our choice x is which petrol station to use when the fuel tank empties or which types of fuel we prefer.

So, how do we *make* choices? What leads us to pick one particular x over all others? If we ignore the fact that in some cases we do like to pick things at random, it is fair to say that we normally decide on a few criteria before making a choice. For instance, quite often we implicitly assign a value (say, a real number \mathbb{R}) to each element of the set, representing the *cost* of making that particular choice. If X is the set of fuel types at a particular petrol station, then our mapping $p: X \rightarrow \mathbb{R}$ could be the cost of a litre of petrol of each particular type x . We then select an $x \in X$ which has minimal cost or the best cost-benefit. In fact, we are only using the real numbers as an ordered set. Any ordered set (R, \leq) would be sufficient. The order does not even need to be total, meaning that for some two candidates x and x' we might not be able to tell which one is better.

Anyway, once we have fixed a more structured set R , our “criteria” mapping $p: X \rightarrow R$ propagates the structure of R to the elements of X . For instance, if R is an ordered set, we immediately get an order on X by saying that a petrol station x is better than another petrol station y if $p(x) \geq p(y)$. Thus, we might be interested in picking an $x \in X$ which either maximises or minimises the value of $p(x)$, depending on whether $p: X \rightarrow R$ represents the quality of fuel or the cost of a litre of that particular fuel type, respectively.

Although the mapping $p: X \rightarrow R$ depends on the criteria used by a particular individual, the fact that we pick an x which minimises the cost is much more uni-

versal. We use the same minimisation function when choosing the cheapest hotel or the cheapest pair of trousers. Any function, such as the minimisation or maximisation functions, that in a uniform way picks an $x \in X$ when given a mapping $p: X \rightarrow R$ we shall call a *selection function*. These are functions of type

$$(X \rightarrow R) \rightarrow X.$$

The “minimisation” selection function, which we use when p describes the *cost* associated with each particular choice, is

$$\varepsilon(p) = \text{any } x \in X \text{ such that } p(x) \leq p(y) \text{ for all } y \in X.$$

whereas the “maximisation” selection functions, which we use when p represents the *payoff* obtained by making a particular choice, is

$$\delta(p) = \text{any } x \in X \text{ such that } p(x) \geq p(y) \text{ for all } y \in X.$$

Both ε and δ have type $(X \rightarrow R) \rightarrow X$.

Note that selection functions decide which element to choose for each given “criteria” mapping $p: X \rightarrow R$. If the cost of the different fuel types change, we might be led to make a different choice, although we are still using the same selection function ε . Only the criteria mapping p will have changed.

Once we have the chosen element $x = \varepsilon(p)$, we can in particular calculate the optimal cost (or payoff) we have achieved as

$$r = p(x) = p(\varepsilon(p)).$$

For a given $p: X \rightarrow R$, we say that $x = \varepsilon(p)$ is an *optimal choice*, whereas $r = p(\varepsilon(p))$ is the *optimal outcome*. We can also define the optimal outcome directly. For instance, in the case of the petrol station, the optimal outcome is the minimum value of the mapping $p: X \rightarrow R$. Since ε picks an x where p has least value, we have that

$$\min(p) = p(\varepsilon(p)),$$

for all $p: X \rightarrow R$. A similar equation to the above also holds for the relation between max and the selection function δ , i.e.

$$\max(p) = p(\delta(p)).$$

In general, we call a function

$$\phi : (X \rightarrow R) \rightarrow R$$

a *quantifier*. When for a given quantifier ϕ a selection function ε always picks $x = \varepsilon(p)$ such that $\phi(p) = p(x)$ we say that ε is a *selection function for ϕ* . In this case ϕ can be defined from epsilon by the equation

$$\phi(p) = p(\varepsilon(p)).$$

Intuitively, both selection functions and quantifiers operate on the criteria mapping $p: X \rightarrow R$. A quantifier ϕ says what is the desired outcome for a given $p: X \rightarrow R$, whereas a selection function ε for ϕ calculates an $x \in X$ which achieves that

outcome. Note that our concept of “quantifier” is more general than Mostowski’s notion of a generalised quantifier (Mostowski 1957), which is the case when R is the set of truth-values or booleans.

For the rest of this review we shall explore the connection between quantifiers and selection functions, and define algebraic operations on them. We will then define a notion of sequential game, and show that the product of selection functions allows us to compute optimal strategies in such games. In the final section we show, by picking particular instantiations from different areas, that this notion of game is very general. This review mostly covers recent work of the authors which has appeared in (Escardó & Oliva 2010a, Escardó & Oliva 2010b, Escardó & Oliva 2010c), although some concepts have been simplified and generalised.

λ -notation. In the following sections we rely heavily on the use of λ -notation to describe functions. For instance, the function “two to the power of n ” can be described as $\lambda n.2^n$. When this is applied to 3, for instance, it gives $(\lambda n.2^n)(3) = 2^3 = 8$. The λ -notation is particularly useful when working with higher-type objects such as functions ε from $X \rightarrow R$ to X , since its argument is itself a function.

Models of higher-type functions. For the first part of the paper (Sections 2, 3, 4, and 5), which concerns finite games of fixed length, the reader can safely think of X and Y as ordinary finite or infinite sets, and the mappings $X \rightarrow Y$ as the set of all functions from X to Y . In Section 6, however, which concerns games of unbounded length, one has to be careful and work with (non-standard) models of functionals, so as to ensure that the infinitely iterated product is well-defined. The model we should bear in mind in this case is that of the continuous functionals (Normann 1999), and in some cases also the model of majorisable functionals (Bezem 1985).

2. The Product of Quantifiers

Continuing from the examples given in the introduction, suppose now that we could only choose which petrol station to stop at, and the owner of the petrol stations then chooses the type of fuel we must use at that particular station we have picked. As we know that the owner will try to maximise his profit and will offer us the most expensive fuel, we have to choose the petrol station with the cheapest most expensive fuel. In other words, if X is the set of petrol stations, Y is the set of different fuel types, and $q: X \times Y \rightarrow \mathbb{R}$ gives the cost of fuel y at station x , then the best outcome for us is to pay

$$r = \inf_{x \in X} \sup_{y \in Y} q(x, y).$$

Interestingly, this is also the optimal outcome for the owner of the petrol stations, as he has no control over which station we choose.

Let us use the abbreviations $K_R X = (X \rightarrow R) \rightarrow R$ and $J_R X = (X \rightarrow R) \rightarrow X$ for the types of quantifiers and selection functions, respectively.

Definition 1 (Binary product of quantifiers). *Given quantifiers $\phi: K_R X$ and $\psi: K_R Y$ we define their product $(\phi \otimes \psi): K_R(X \times Y)$ as*

$$(\phi \otimes \psi)(q^{X \times Y \rightarrow R}) = \phi(\lambda x^X. \psi(\lambda y^Y. q(x, y))).$$

For instance, given the quantifiers $\text{inf}: K_{\mathbb{R}}X$ and $\text{sup}: K_{\mathbb{R}}Y$ discussed above we have that their product is

$$(\text{sup} \otimes \text{inf})(q^{X \times Y \rightarrow \mathbb{R}}) \stackrel{\mathbb{R}}{=} \sup_{x \in X} \inf_{y \in Y} q(x, y),$$

where, for clarity, we indicate above the equality sign the type of the two objects. Also, given the quantifiers $\exists: K_{\mathbb{B}}X$ and $\forall: K_{\mathbb{B}}Y$, corresponding to the existential and universal quantifiers, we have that their product concisely describes the nesting of quantifiers, i.e.

$$(\exists \otimes \forall)(q^{X \times Y \rightarrow \mathbb{B}}) \stackrel{\mathbb{B}}{=} \exists x^X \forall y^Y q(x, y).$$

We can iterate this binary product of quantifiers and obtain an n -ary product as follows:

Definition 2 (n -ary product of quantifiers). *Given a sequence of quantifiers $\phi_i: K_R X_i$, $0 \leq i \leq n$, define their product inductively by iterating the binary product as*

$$\bigotimes_{i=k}^n \phi_i = \phi_k \otimes \left(\bigotimes_{i=k+1}^n \phi_i \right)$$

where $(\bigotimes_{i=n}^n \phi_i) = \phi_n$.

Let us show how the concept of quantifiers allows us to define a very general notion of sequential games. We will also show that the product of quantifiers described above computes the optimal outcome of such games.

3. Finite Games of Fixed Length

Games often simulate life. In particular, a play in a game can be viewed as a sequence of choices that lead to some outcome. Hence, we can think of a sequence of choices as a sequence of moves in some sequential game. In the case of choosing which petrol station to use, this is a simple game with one player and one round. In the case we must also choose which fuel type to use, this is a one-player two-round game. If the owner of the station chooses the type of fuel we must use, then this is a two-player two-round game. In all cases, the aim of the game for us (the first player in the last example) is to minimise the cost (or maximise the payoff) obtained from a particular choice, and the actual move in the game is the particular choice we commit ourselves to.

In the following we explore the concept of a quantifier introduced above and generalise these scenarios to games with a finite number of rounds.

Definition 3 (Finite game). *Any tuple $(R, (X_i)_{i=0}^{n-1}, (\phi_i)_{i=0}^{n-1}, q)$ defines an n -round sequential game, where*

1. R is the set of possible outcomes of the game,
2. X_i is the set of possible moves at round i ,
3. $\phi_i: K_R X_i$ is the quantifier for round i , and
4. $q: \prod_{i=0}^{n-1} X_i \rightarrow R$ is the outcome function.

A sequence $x_0, \dots, x_{n-1} \in \prod_{i=0}^{n-1} X_i$ is called a play of the game.

Note that we have said nothing about the number of *players* in the game. Instead, the definition of the game includes a quantifier ϕ_i for each round. This function ϕ_i picks a particular outcome $\phi_i(p) = r \in R$ when given a mapping $p: X_i \rightarrow R$. Intuitively, once we know how each choice of move in X_i will impact the outcome of the game, then ϕ_i tells us what outcome would be desirable.

For instance, if $R = \{-1, 0, 1\}$ and ϕ_i are the minimum-value function for odd rounds and the maximum-value function for even rounds, then we have a two player game where one player is aiming for the outcome -1 whereas the other is aiming for the outcome 1 . Thus, the three possible outcomes correspond to one of the players winning, or the game ending in a draw.

The example in Section 2 corresponds to the case when $n = 2$, i.e. a game with two rounds. X is the set of petrol stations and Y is the collection of different fuel types. The outcome function $q: X \times Y \rightarrow \mathbb{R}$ gives the cost of a particular fuel at a particular station. Since we are choosing which station to use, and the owner of the stations chooses the fuel to be used, we have that $\phi_0 = \inf$ whereas $\phi_1 = \sup$.

Definition 4 (Strategy). A strategy for the game $(R, (X_i)_{i=0}^{n-1}, (\phi_i)_{i=0}^{n-1}, q)$ is a family of mappings $\text{next}_k: \prod_{i=0}^{k-1} X_i \rightarrow X_k$, for each $0 \leq k < n$. Given a partial play $\vec{a} = a_0 \dots, a_{k-1}$, any strategy defines an strategic extension of \vec{a} as

$$b_j^{\vec{a}} \stackrel{X_j}{=} \text{next}_j(a_0, \dots, a_{k-1}, b_k^{\vec{a}}, \dots, b_{j-1}^{\vec{a}})$$

for $k \leq j < n$.

Given a strategy $(\text{next}_k)_{k < n}$, the functions next_k compute which move should be played at each round k , i.e. when the game is at position $\vec{a} = a_0, \dots, a_{k-1}$ the next move selected is given by $a_k = \text{next}_k(\vec{a})$. The strategic extension of a partial play is simply the play obtained by following the strategy at each round.

Definition 5 (Optimal strategy). A strategy $(\text{next}_k)_{k < n}$ is said to be optimal if for every partial play $\vec{a} = a_0, \dots, a_{k-1}$, we have

$$q(\vec{a}, b_k^{\vec{a}}, \dots, b_{n-1}^{\vec{a}}) \stackrel{R}{=} \phi_k(\lambda x_k. q(\vec{a}, x_k, b_{k+1}^{\vec{a}, x_k}, \dots, b_{n-1}^{\vec{a}, x_k})).$$

Intuitively, a strategy is optimal if the outcome obtained by following the strategy $q(\vec{a}, b_k^{\vec{a}}, \dots, b_{n-1}^{\vec{a}})$ is the best possible outcome as described by the quantifier ϕ_k , for each round k . For instance, considering again the example from Section 2, the optimal strategy would consist of the functions

$$\text{next}_0() = \text{any } x \in X \text{ such that } \sup q_x \leq \sup q_{x'} \text{ for any other } x' \in X$$

$$\text{next}_1(x) = \text{any } y \in Y \text{ such that } q(x, y) \geq q(x, y') \text{ for any other } y' \in Y$$

where $\sup q_x$ abbreviates $\sup_{y \in Y} q(x, y)$. In general, we will write q_x for the function $\lambda y. q(x, y)$.

Theorem 3.1 (Optimal outcome). For any optimal strategy $(\text{next}_k)_{k < n}$ let \vec{b} be its strategic play. We have that

$$w := \left(\bigotimes_{i=0}^{n-1} \phi_i \right) (q) = q(\vec{b}).$$

We call w the optimal outcome of the game. Note that this is unique and independent of the optimal strategy.

Proof. Let an optimal strategy $(\text{next}_k)_{k < n}$ be fixed. By the definition of optimal strategy (Definition 5) we have

$$\begin{aligned} q(\vec{b}) &= \phi_0(\lambda x_0 \cdot q(x_0, b_1^{x_0}, \dots, b_{n-1}^{x_0})) \\ &= \phi_0(\lambda x_0 \cdot \phi_1(\lambda x_1 \cdot q(x_0, x_1, b_2^{x_0, x_1}, \dots, b_{n-1}^{x_0, x_1}))) \\ &\dots \\ &= \left(\bigotimes_{i=0}^{n-1} \phi_i \right) (q). \end{aligned}$$

□

As discussed in Section 2, the optimal outcome of the game between us and the owner of the petrol station is for us to choose the station with cheapest most expensive fuel, whereas the owner then picks the most expensive fuel at the station we have chosen. So the optimal outcome of the game is

$$w = (\inf \otimes \sup)(q).$$

If on the other hand we could choose not only which station to use, but also which fuel type to use (and if we were interested in minimising our cost) that is a different game where $\phi_0 = \phi_1 = \inf$ and the optimal outcome of the game is

$$w = (\inf \otimes \inf)(q),$$

where we pick the station which has the cheapest fuel, and then pick the cheapest fuel at that chosen station.

Definition 6 (Subgames). *A partial play $\vec{a}: \prod_{i=0}^{k-1} X_i$, for $k \leq n$, defines the subgame $(R, (X_i)_{i=k}^{n-1}, (\phi_i)_{i=k}^{n-1}, q_{\vec{a}})$ where $q_{\vec{a}}: \prod_{i=k}^{n-1} X_i \rightarrow R$ is the partial outcome function defined as*

$$q_{\vec{a}}(x_k, \dots, x_{n-1}) := q(a_0, \dots, a_{k-1}, x_k, \dots, x_{n-1}),$$

or, more concisely $q_{\vec{a}}(\vec{x}) := q(\vec{a} * \vec{x})$, where $*$ denotes concatenation of finite sequences.

The subgame $(R, (X_i)_{i=k}^{n-1}, (\phi_i)_{i=k}^{n-1}, q_{\vec{a}})$ is like the original game except that we are starting at the position determined by the initial moves \vec{a} . Notice that if $k = n$ then q is constant, and when $k = 0$ this is the same as the full game. The optimal outcome of the subgame defined by \vec{a} can similarly be computed as

$$w_{\vec{a}} := \left(\bigotimes_{i=k}^{n-1} \phi_i \right) (q_{\vec{a}}).$$

If $k = n$ this corresponds to the subgame where there are no more moves to be made – the end of the game. Therefore, the outcome can be computed directly from

the moves already made \vec{a} , i.e. $w_{\vec{a}} = q(\vec{a})$. If $k < n$ then we have a proper subgame, and the optimal outcome of this proper subgame is

$$w_{\vec{a}} = \phi_k \left(\lambda x_k \cdot \left(\bigotimes_{i=k+1}^{n-1} \phi_i \right) (q_{\vec{a}*x_k}) \right) = \phi_k (\lambda x_k \cdot w_{\vec{a}*x_k}).$$

Hence, the optimal outcome of round k is determined by the quantifier for round k together with the mapping $\lambda x_k \cdot w_{\vec{a}*x_k}$ computing the optimal outcome at round $k+1$ given what is played at round k . Considering again the example from Section 2, when choosing which petrol station x to stop at, it is enough to look at the most expensive fuel type w_x at each station, because we know that this is the fuel type the owner of the stations will offer us. Therefore, the optimal outcome for us is $\inf(\lambda x \cdot w_x)$. The notion of optimal outcome for subgames leads to the notion of optimal move.

Definition 7 (Optimal move and optimal play). *For a partial play $\vec{a} = a_0, \dots, a_{k-1}$ a move x_k at round k is said to be an optimal move if it preserves the optimal outcome, i.e. if the optimal outcome $w_{\vec{a}}$ of the subgame determined by \vec{a} is the same as the optimal outcome $w_{\vec{a}*x_k}$ of the new subgame determined by $\vec{a} * x_k$. A whole play $\vec{x} = x_0, \dots, x_{n-1}$ is called an optimal play if each x_k is an optimal move given the partial play x_0, \dots, x_{k-1} .*

In other words, a move at round k is optimal if it leads to a subgame where the optimal outcome of the game is still achievable. As one would expect, if we play optimally at each round we will end up with the optimal outcome of the game.

Lemma 3.2. *If $\vec{x} = x_0, \dots, x_{n-1}$ is an optimal play then*

$$q(\vec{x}) = w = \left(\bigotimes_{i=0}^{n-1} \phi_i \right) (q),$$

i.e. the outcome of an optimal play is the optimal outcome.

Proof. Since \vec{x} is an optimal play we have that (for all $0 \leq i < n$) x_i is an optimal move in the subgame x_0, \dots, x_{i-1} , i.e.

$$w = w_{x_0} = w_{x_0, x_1} = \dots = w_{x_0, \dots, x_{n-1}}.$$

The result follows since $w_{x_0, \dots, x_{n-1}} = q(\vec{x})$ whereas $w = \left(\bigotimes_{i=0}^{n-1} \phi_i \right) (q)$. \square

Moreover, the next lemma shows that plays obtained by following optimal strategies are indeed optimal.

Lemma 3.3 (Optimal strategies and optimal plays). *For any optimal strategy its strategic play is an optimal play.*

Proof. Assume $(\text{next}_k)_{k < n}$ is an optimal strategy, and let $\vec{b} = b_0, \dots, b_{n-1}$ be the strategic play. We must show, for $0 \leq k < n-1$, that $w_{\vec{a}} = w_{\vec{a}*b_k}$, where

$\vec{a} = b_0, \dots, b_{k-1}$. By (i) the definition of optimal outcome and (ii) the fact that $(\text{next}_k)_{k < n}$ is an optimal strategy we have

$$\begin{aligned}
w_{\vec{a}} &\stackrel{(i)}{=} \left(\bigotimes_{i=k}^{n-1} \phi_i \right) (q_{\vec{a}}) \\
&\stackrel{\text{T3.1}}{=} q(\vec{a} * \vec{b}^{\vec{a}}) \\
&= q(\vec{a} * b_k * \vec{b}^{\vec{a} * b_k}) \\
&\stackrel{(ii)}{=} \phi_{k+1}(\lambda x_{k+1}. q(\vec{a} * b_k * x_{k+1} * \vec{b}^{\vec{a} * b_k * x_{k+1}})) \\
&\stackrel{\text{T3.1}}{=} \phi_{k+1} \left(\lambda x_{k+1}. \left(\bigotimes_{i=k+2}^{n-1} \phi_i \right) (q_{\vec{a} * b_k * x_{k+1}}) \right) \\
&= \left(\bigotimes_{i=k+1}^{n-1} \phi_i \right) (q_{\vec{a} * b_k}) \\
&\stackrel{(i)}{=} w_{\vec{a} * b_k}.
\end{aligned}$$

□

Note that optimal strategies do not exist in general, since optimal outcomes are not necessarily achievable. For instance, consider the game with a single round having quantifier $\phi_0: K_{\{0,1\}}\{0,1\}$

$$\phi_0(p) = 0,$$

and assume the outcome function is $q(x) = 1$. The optimal outcome defined by ϕ_0 is 0, whereas the only achievable outcome is 1. Nevertheless, in Section 5 we will show that optimal strategies always exist when the quantifiers that define the game are *attainable*, i.e. have corresponding selection functions. Moreover, a suitably defined product of selection functions calculates optimal strategies.

4. Games with Multiple Optimal Outcomes

Before we proceed to show how a product of selection functions computes optimal strategies, let us first generalise the notion of game introduced in Section 3. For that, let us consider the space

$$S_R X \equiv (X \rightarrow R) \rightarrow 2^R$$

of *multi-valued quantifiers*. Similarly to before, we say that a multi-valued quantifier $\phi: S_R X$ is *attainable* if there exists a selection function $\varepsilon: J_R X$ such that

$$p(\varepsilon(p)) \in \phi(p),$$

for all $p: X \rightarrow R$. Intuitively, ϕ describes a set of optimal outcomes for any given criteria mapping, and the selection function ε for ϕ picks an $x \in X$ which achieves one of the optimal outcomes.

Definition 8 (Finite games with multiple optimal outcomes). *An n -round sequential game with multiple optimal outcomes is a tuple $(R, (X_i)_{i=0}^{n-1}, (\phi_i)_{i=0}^{n-1}, q)$ where*

1. R is the set of possible outcomes of the game

2. X_i is the set of possible moves at round i ,
3. $\phi_i: S_R X_i$ is the multi-valued quantifier for round i , and
4. $q: \prod_{i=0}^{n-1} X_i \rightarrow R$ is the outcome function.

A sequence $x_0, \dots, x_{n-1} \in \prod_{i=0}^{n-1} X_i$ is called a play of the game.

Just as in Section 3, every partial play $\vec{a}: \prod_{i=0}^{k-1} X_i$ defines a subgame. In games with multiple optimal outcomes, however, we cannot define “the” optimal outcome as we did in the previous section, simply because there might not be a single optimal outcome for the whole game. Moreover, there seems to be no way of defining a product of multi-valued quantifiers, as we did for the single-valued quantifiers. Nevertheless, we can still define a notion of an optimal strategy, slightly generalising Definition 5.

Definition 9 (Optimal strategy). Recall that a partial play $\vec{a} = a_0, \dots, a_{k-1}$ and a strategy $(\text{next}_k: \prod_{i=0}^{k-1} X_i \rightarrow X_k)_{k < n}$ defines a strategic extension of \vec{a} as

$$b_j^{\vec{a}} = \text{next}_j(a_0, \dots, a_{k-1}, b_k^{\vec{a}}, \dots, b_{j-1}^{\vec{a}})$$

for $k \leq j < n$. A strategy $(\text{next}_k)_{k < n}$ is said to be optimal if for every partial play $\vec{a} = a_0, \dots, a_{k-1}$, we have

$$q(\vec{a}, b_k^{\vec{a}}, \dots, b_{n-1}^{\vec{a}}) \in \phi_k(\lambda x_k. q(\vec{a}, x_k, b_{k+1}^{\vec{a}, x_k}, \dots, b_{n-1}^{\vec{a}, x_k})).$$

Compare this with Definition 5, where we have equality (=) instead of \in .

In the previous section we said that a strategy was optimal if the outcome of a strategic play $q(\vec{a}, b_k^{\vec{a}}, \dots, b_{n-1}^{\vec{a}})$ was precisely the optimal outcome according to the quantifier ϕ_k . By moving to multi-valued quantifiers, what we must say instead is that a strategy is optimal if the outcome of a strategic play is among the possible optimal outcomes for each of the multi-valued quantifier ϕ_k . It is clear that when multi-valued quantifiers are in fact single-valued the two notions of optimal strategy (Definitions 5 and 9) coincide.

5. Calculating Optimal Strategies

We have explained at the end of Section 3 why optimal outcomes are not always achievable. Note also that even when an optimal outcome is achievable, optimal strategies are not necessarily unique. There might be several choices which are all equally optimal, if all lead to an optimal outcome of the game. Nevertheless, what we show next is that if the (multi-valued) quantifiers ϕ_k for each round k have associated selection functions ε_k , then we can always compute optimal strategies (in the more general sense of Definition 9) in the game defined by such ϕ_k 's. Intuitively, having a selection function ε is a stronger requirement than simply having a quantifier ϕ , because every selection function ε defines a (single-valued) quantifier as

$$\phi(p) := p(\varepsilon(p)).$$

Recall that $J_R X$ abbreviates $(X \rightarrow R) \rightarrow X$.

Definition 10 (From J_RX to K_RX). Denote by $\overline{(\cdot)}: J_RX \rightarrow K_RX$ the following mapping from selection functions to quantifiers

$$\overline{\varepsilon}(p) := p(\varepsilon(p)). \quad (5.1)$$

Going back to the example from Section 2, we are in a good position because the quantifiers \inf and \sup in this case have selection functions ε and δ respectively (as defined in the introduction). And it is for that reason that we can find a play which is optimal and hence leads to the optimal outcome

$$\inf_{x \in X} \sup_{y \in Y} q(x, y).$$

But how do we compute the optimal play x, y which leads to this optimal outcome? The solution is to turn the game upside down. Instead of trying to find the correct x first, we instead look for the best choice of $y \in Y$ for any given $x \in X$. That can be done very easily using the selection function δ for the quantifier \max , i.e.

$$y_x := \delta(\lambda y. q(x, y)).$$

Hence, for each of our choices of petrol station $x \in X$, we know what the optimal move $y_x \in Y$ of the second player is. Once we have this mapping $x \mapsto y_x$ from our choice of station x to the fuel type $y_x \in Y$ the owner of the petrol station will offer us, we can choose the petrol station with cheapest most expensive fuel. For that we use the selection function for the minimisation quantifier, i.e.

$$a := \varepsilon(\lambda x. q(x, y_x)).$$

Finally, once we have computed our best choice of petrol station a as above, we can then go back to y_x and compute the optimal move for the second player as $y = y_a$ (where a is now the optimal move for player one). Such construction works in general, as we have only used the fact that ε and δ are selection functions for the quantifiers \min and \max , respectively.

Definition 11 (Binary product of selection functions). Given selection functions $\varepsilon: J_RX$ and $\delta: J_RY$, define their product $(\varepsilon \otimes \delta): J_R(X \times Y)$ using λ -notation as

$$(\varepsilon \otimes \delta)(q^{X \times Y \rightarrow R}) = (a, y_a),$$

where $a = \varepsilon(\lambda x. q(x, y_x))$ and $y_x = \delta(\lambda y. q(x, y))$. We can then equivalently define the product above using function composition as

$$(\varepsilon \otimes \delta)(q^{X \times Y \rightarrow R}) = (\varepsilon(p), (\delta \circ \hat{q} \circ \varepsilon)(p)),$$

where $p \stackrel{X \rightarrow R}{\cong} \overline{\delta} \circ \hat{q}$ and $\hat{q}: X \rightarrow (Y \rightarrow R)$ is the curried form of $q: X \times Y \rightarrow R$. This is illustrated diagrammatically as

$$\begin{array}{ccccccc} (X \rightarrow R) & \xrightarrow{\varepsilon} & X & \xrightarrow{\hat{q}} & (Y \rightarrow R) & \xrightarrow{\delta} & Y \\ & & & & & & \\ X & \xrightarrow{\hat{q}} & (Y \rightarrow R) & \xrightarrow{\overline{\delta}} & R. & & \end{array}$$

First let us show that the product of selection functions corresponds to the product of single-valued quantifiers defined in Section 2.

Lemma 5.1. *For any given $\varepsilon: J_R X$ and $\delta: J_R Y$ we have $\overline{\varepsilon \otimes \delta} = \overline{\varepsilon} \otimes \overline{\delta}$.*

Proof. Simply unfolding definitions we have

$$(\overline{\varepsilon \otimes \delta})(q) = q(a, b_a) = \overline{\varepsilon}(\lambda x.q(x, b_x)) = \overline{\varepsilon}(\lambda x.\overline{\delta}(\lambda y.q(x, y))) = (\overline{\varepsilon} \otimes \overline{\delta})(q).$$

□

From this we can conclude that, just like the product of quantifiers (Definition 1) computes the optimal outcome of a single-optimal-outcome game, the product of selection functions (Definition 11) computes an optimal play in such games.

Theorem 5.2 (Optimal plays for 2-round games). *Let a 2-round sequential game $(R, (X_i)_{i=0}^1, (\phi_i)_{i=0}^1, q)$ be given. If the quantifiers ϕ_0, ϕ_1 have selection functions $\varepsilon_0, \varepsilon_1$ then an optimal play in the game can be computed as*

$$x, y = (\varepsilon_0 \otimes \varepsilon_1)(q).$$

Proof. We must show that x and y are optimal moves, i.e that $w_{x,y} = q(x, y)$ equals $w_x = \phi_1(\lambda y.q(x, y))$ which in turn must equal the optimal outcome $w = (\phi_0 \otimes \phi_1)(q)$. First note that $x = \varepsilon_0(\lambda x'.\phi_1(\lambda y'.q(x', y')))$ and $y = \varepsilon_1(\lambda y'.q(x, y'))$. The result then follows as

$$\begin{aligned} (\phi_0 \otimes \phi_1)(q) &= \phi_0(\lambda x'.\phi_1(\lambda y'.q(x', y'))) \\ &= \phi_1(\lambda y'.q(x, y')) \\ &= q(x, y), \end{aligned}$$

since $\phi_0(p) = p(\varepsilon_0(p))$ and $\phi_1(p) = p(\varepsilon_1(p))$. □

In the general case of multi-valued outcome functions, we have:

Theorem 5.3 (Optimal strategies for 2-round games – Multi-valued ϕ_i 's). *Let a 2-round sequential game $(R, (X_i)_{i=0}^1, (\phi_i)_{i=0}^1, q)$ be given, where $\phi_0: S_R X_0$ and $\phi_1: S_R X_1$ are multi-valued quantifiers. If ϕ_0, ϕ_1 have selection functions $\varepsilon_0, \varepsilon_1$ then an optimal strategy in the game can be computed as*

$$\begin{aligned} \text{next}_0() &= \pi_0((\varepsilon_0 \otimes \varepsilon_1)(q)) \\ \text{next}_1(x) &= \varepsilon_1(\lambda y.q_x(y)) \end{aligned}$$

where π_0 denotes the first projection, i.e. $\pi_0(x, y) = x$.

Just as we did with the binary product of quantifiers (see Definition 2) we can iterate the binary product of selection function to obtain an n -ary product as follows:

Definition 12 (n -ary product of selection functions). *Given a sequence of selection functions $\varepsilon_i: J_R X_i$, define their product by simply iterating the binary product as*

$$\bigotimes_{i=k}^n \varepsilon_i = \varepsilon_k \otimes \left(\bigotimes_{i=k+1}^n \varepsilon_i \right)$$

where $(\bigotimes_{i=n}^n \varepsilon_i) = \varepsilon_n$.

The following theorem describes how this product of selection functions can be used to compute optimal strategies.

Theorem 5.4 (Main theorem for finite games of fixed length). *Let an n -round sequential game $(R, (X_i)_{i=0}^{n-1}, (\phi_i)_{i=0}^{n-1}, q)$ be given (in the more general sense of Definition 8). If the multi-valued quantifiers $\phi_i: S_R X$ are attainable, i.e. have associated selection functions ε_i , then an optimal strategy for the game can be computed as*

$$\text{next}_k(x_0, \dots, x_{k-1}) = \pi_0 \left(\left(\bigotimes_{i=k}^{n-1} \varepsilon_i \right) (q_{x_0, \dots, x_{k-1}}) \right)$$

where π_0 denotes the first projection, i.e. $\pi_0(x * \alpha) = x$.

Proof. This is a particular case of the more general Theorem 6.2 which we prove in Section 6. \square

We call the optimal strategy computed from ε the ε -strategy, and its strategic play we call the ε -play. As a consequence of the above theorem we obtain a general way of finding a solution \vec{a} and \vec{p} to the following simple system of equations and membership. We will discuss in the final Section 7 a few examples of how such system of equations appear in practice.

Corollary 5.5. *Let an n -round sequential game $(R, (X_i)_{i=0}^{n-1}, (\phi_i)_{i=0}^{n-1}, q)$ be given. If the multi-valued quantifiers $\phi_i: S_R X$ are attainable then there are functions $p_k: X_k \rightarrow R$ such that*

$$\begin{aligned} a_k &= \varepsilon_k(p_k) \\ q(\vec{a}) &\in \phi_k(p_k), \end{aligned}$$

for $0 \leq k < n$, where \vec{a} is the ε -play.

Proof. This is a particular case of the more general Corollary 6.3 which we prove in Section 6. \square

In particular, note that if ϕ_k are all single-valued then $p_j(a_j) = p_k(a_k)$, for all $0 \leq j, k < n$, since

$$p_j(a_j) = p_j(\varepsilon_j(p_j)) = \phi_j(p_j) = q(\vec{a}) = \phi_k(p_k) = p_k(\varepsilon_k(p_k)) = p_k(a_k).$$

The intuitive reading of Corollary 5.5 is that in any sequential game (with multi-valued outcome functions) it is always possible to turn the *global* outcome function $q: \prod_{i=0}^{n-1} X_i \rightarrow R$ into a family of *local* outcome functions $p_k: X_k \rightarrow R$ so that

- the ε -play \vec{a} can be computed locally from the selection functions ε_k , and
- the outcome of the ε -play is in the intersection of “desired” outcomes for all rounds k .

6. Finite Games of Unbounded Length

The sort of games considered so far terminate after a fixed number of rounds n . We will show next that it is also possible to compute optimal strategies in games with an unbounded (but finite) number of rounds. These are games where the number

of rounds until the outcome is reached depends on how the game develops. Because the number of rounds in the game can be arbitrarily large, we must include in the definition of the game a set of possible moves X_i and an outcome quantifier ϕ_i for all natural numbers $i \in \mathbb{N}$. However, we do not want to consider games that go on indefinitely. One way to ensure that every play in the game is finite is to assume that the value of the outcome function q is determined after a finite (but unbounded) number of rounds, i.e.

$$\forall \alpha \exists n \forall \beta (\forall i \leq n (\alpha(i) \stackrel{X_i}{=} \beta(i)) \rightarrow q(\alpha) \stackrel{R}{=} q(\beta)). \quad (6.1)$$

For any “infinite” play α there exists a point n where the value of the outcome $q(\alpha)$ is fixed, so that any other extension β of the play would lead to the same outcome. This holds, for instance, when the set of outcomes R is discrete[†] and the outcome function q is continuous.

Definition 13 (Finite but unbounded game). *A finite but unbounded sequential game is a tuple $(R, (X_i)_{i=0}^\infty, (\phi_i)_{i=0}^\infty, q)$ where*

1. R is the discrete set of possible outcomes of the game,
2. X_i is the set of possible moves at round $i \in \mathbb{N}$,
3. $\phi_i: S_R X_i$ is the multi-valued quantifier for round $i \in \mathbb{N}$, and
4. $q: \prod_{i=0}^\infty X_i \rightarrow R$ is the continuous outcome function.

A sequence $\alpha \in \prod_{i=0}^\infty X_i$ is called a play of the game. The relevant part of a play α is the finite initial segment $\alpha(0), \dots, \alpha(n) \in \prod_{i=0}^n X_i$ which determines the value of $q(\alpha)$.

As in Section 3, a strategy for an unbounded game $(R, (X_i)_{i=0}^\infty, (\phi_i)_{i=0}^\infty, q)$ is a family of mappings $\text{next}_k: \prod_{i=0}^{k-1} X_i \rightarrow X_k$, for each $k \in \mathbb{N}$. Given a partial play $\vec{a} = a_0 \dots, a_{k-1}$, any strategy defines an *strategic extension* of \vec{a} as

$$\beta^{\vec{a}}(j) = \text{next}_j(a_0, \dots, a_{k-1}, \beta^{\vec{a}}(k), \dots, \beta^{\vec{a}}(j-1))$$

for $j \geq k$.

Definition 14 (Optimal strategy). *A strategy $(\text{next}_k)_{k \in \mathbb{N}}$ is said to be optimal if for all $k \in \mathbb{N}$ and for every partial play $\vec{a} = a_0, \dots, a_{k-1}$ we have*

$$q(\vec{a} * \beta^{\vec{a}}) \in \phi_k(\lambda x_k. q(\vec{a} * x_k * \beta^{\vec{a}, x_k})).$$

Having defined the unbounded game and its generalised notion of optimal strategy, we now show how the infinite iteration of the binary product of selection functions computes optimal strategies in games where the quantifiers ϕ_i are attainable.

Definition 15 (Unbounded product of selection functions). *Given an infinite sequence of selection functions $\varepsilon_i: J_R X_i$ define their unbounded product by simply iterating the binary product as*

$$\bigotimes_{i=k}^\infty \varepsilon_i \stackrel{J_R \prod_{i=k}^\infty X_i}{=} \varepsilon_k \otimes \left(\bigotimes_{i=k+1}^\infty \varepsilon_i \right)$$

[†] E.g. the natural numbers or more generally the types defined in (Escardó 2008, Definition 4.12).

where, for clarity, the type of the final product is shown above the equality sign.

It is perhaps surprising that such product functional is in fact not only well-defined in the model of continuous functionals (Scarpellini 1971) but also computable and part of the standard installation of Haskell (see Section 7(g) for more details). This is in stark contrast with the iterated product of quantifiers which is not well-defined even in the model of continuous functionals.

We will show next that the above product of selection functions permits us to compute optimal strategies for finite but unbounded games with attainable quantifiers. First, let us prove the following useful lemma.

Lemma 6.1. *Let $\vec{x} = x_0, \dots, x_{k-1}$ and*

$$\alpha = \left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right) (q_{\vec{x}}).$$

For all $j \in \mathbb{N}$ the following holds:

$$\alpha = (\alpha)[j] * \left(\bigotimes_{i=j+k}^{\infty} \varepsilon_i \right) (q_{\vec{x}*(\alpha)[j]}).$$

where $(\alpha)[j]$ is the initial segment of α of length j , i.e. $\alpha(0), \dots, \alpha(j-1)$.

Proof. If $j = 0$ this follows by definition. Assume this holds for j we wish to show it for $j + 1$. We have

$$\alpha \stackrel{\text{(IH)}}{=} (\alpha)[j] * \left(\bigotimes_{i=j+k}^{\infty} \varepsilon_i \right) (q_{\vec{x}*(\alpha)[j]}) = (\alpha)[j] * c * \left(\bigotimes_{i=j+k+1}^{\infty} \varepsilon_i \right) (q_{\vec{x}*(\alpha)[j]*c}),$$

where $c = \varepsilon_{j+k} \left(\lambda y. \overline{\left(\bigotimes_{i=j+k+1}^{\infty} \varepsilon_i \right) (q_{\vec{x}*(\alpha)[j]*y})} \right) = \alpha(j)$. Therefore, we have that $\alpha = (\alpha)[j + 1] * \left(\bigotimes_{i=j+k+1}^{\infty} \varepsilon_i \right) (q_{\vec{x}*(\alpha)[j+1]})$. \square

As with the finite games of fixed length (Theorem 5.4) we now show how the unbounded product of selection functions (Definition 15) computes optimal strategies in games of unbounded length.

Theorem 6.2 (Main theorem for finite but unbounded games). *Let a finite but unbounded sequential game $(R, (X_i)_{i=0}^{\infty}, (\phi_i)_{i=0}^{\infty}, q)$ be given. If the quantifiers ϕ_i have selection functions ε_i then an optimal strategy in the game can be computed as*

$$\text{next}_k(\vec{x}) = \pi_0 \left(\left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right) (q_{\vec{x}}) \right),$$

where $\vec{x} = x_0, \dots, x_{k-1}$.

Proof. Fix an arbitrary partial play $\vec{x} = x_0, \dots, x_{k-1}$. Let

$$\alpha = \left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right) (q_{\vec{x}}).$$

We have that

$$\beta^{\vec{x}}(k) = \text{next}_k(\vec{x}) = \alpha(0) = \varepsilon_k \left(\lambda x_k \cdot q_{\vec{x}, x_k} \left(\left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) (q_{\vec{x}, x_k}) \right) \right).$$

By Lemma 6.1 we also have that

$$\pi_n \left(\left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) (q_{\vec{x}, x_k}) \right) = \beta^{\vec{x}, x_k}(k+1+n).$$

This implies that

$$\beta^{\vec{x}}(k) = \text{next}_k(\vec{x}) = \varepsilon_k (\lambda x_k \cdot q(\vec{x}, x_k * \beta^{\vec{x}, x_k})).$$

By the fact that ε_k is a selection function for ϕ_k we get

$$q(\vec{x}, \beta^{\vec{x}}(k) * \beta^{\vec{x}, \beta^{\vec{x}}(k)}) \in \phi_k(\lambda x_k \cdot q(\vec{x}, x_k * \beta^{\vec{x}, x_k})).$$

The result follows since $q(\vec{x}, \beta^{\vec{x}}(k) * \beta^{\vec{x}, \beta^{\vec{x}}(k)}) = q(\vec{x} * \beta^{\vec{x}})$, i.e. starting a strategic play from \vec{x} or an extension of that with the next strategic play $\vec{x}, \beta^{\vec{x}}(k)$ leads to the same final play. \square

As a corollary, we obtain that the following system of equations and membership can be solved via the product of selection functions. This generalises Corollary 5.5.

Corollary 6.3. *Let a finite but unbounded sequential game $(R, (X_i)_{i=0}^{\infty}, (\phi_i)_{i=0}^{\infty}, q)$ be given. If the quantifiers ϕ_k have associated selection functions ε_k then there are functions $p_k: X_k \rightarrow R$ such that*

$$\begin{aligned} \alpha(k) &\stackrel{X_k}{=} \varepsilon_k(p_k) \\ q(\alpha) &\in \phi_k(p_k), \end{aligned}$$

for $k \in \mathbb{N}$, where α is the ε -strategic play from the proof of Theorem 6.2, i. e.

$$\alpha(k) = \varepsilon_k \left(\lambda x_k \cdot q_{(\alpha)[k] * x_k} \left(\left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) (q_{(\alpha)[k] * x_k}) \right) \right).$$

Proof. Let

$$p_k(x_k) = \overline{\left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) (q_{(\alpha)[k] * x_k})}.$$

By definition $\alpha(k) = \text{next}_k(\alpha(0), \dots, \alpha(k-1))$, i.e.

$$\alpha(k) = \varepsilon_k \left(\lambda x_k \cdot q_{(\alpha)[k] * x_k} \left(\left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) (q_{(\alpha)[k] * x_k}) \right) \right) = \varepsilon_k(p_k).$$

Moreover, since next_k is an optimal strategy we have, for all $k \in \mathbb{N}$, that

$$\begin{aligned} q(\alpha) &= q((\alpha)[k] * \beta^{(\alpha)[k]}) \\ &\in \phi_k(\lambda x_k \cdot q((\alpha)[k] * x_k * \beta^{(\alpha)[k], x_k})) \\ &\in \phi_k(\lambda x_k \cdot q((\alpha)[k] * x_k * \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) (q_{(\alpha)[k] * x_k}))) \\ &= \phi_k(p_k). \end{aligned}$$

\square

(a) Variant of Finite Games of Unbounded Length

The generalisation of finite games to unbounded games considered above relies on the set of outcome R being discrete (and q being continuous). What if R is not discrete, e.g. $R = \mathbb{R}$? Another way to ensure that every play in the game is finite is to assume that each outcome $r \in R$ determines the length of the play to be considered, via a fixed mapping $l: R \rightarrow \mathbb{N}$, which we will call the “clock function”.

Definition 16 (Unbounded game with explicit control). *A finite but unbounded sequential game with explicit control is a tuple $(R, (X_i)_{i=0}^\infty, (\phi_i)_{i=0}^\infty, q, l)$ where*

1. R is the set of possible outcomes of the game,
2. X_i is the set of possible moves at round i ,
3. $\phi_i: S_R X_i$ is the multi-valued quantifier for round i ,
4. $q: \prod_{i=0}^\infty X_i \rightarrow R$ is the outcome function, and
5. $l: R \rightarrow \mathbb{N}$ is the clock function[†].

A sequence $\alpha \in \prod_{i=0}^\infty X_i$ is called a play of the game. The relevant part of a play α is the finite initial segment $\alpha_0, \dots, \alpha_n \in \prod_{i=0}^{n-1} X_i$ of length $n = l(q(\alpha))$.

Therefore, only finite initial segments of infinite plays are to be considered, although the non-relevant extension may be crucial to deciding the outcome of the game (a different non-relevant extension might lead to a different outcome). In fact, the length of the relevant part of the infinite play is only determined once we have fixed the infinite play α . As we are only interested in finite games, we assume that for each set of moves X_i there exists a canonical element $c_i \in X_i$, and that $\mathbf{c}^k \in \prod_{i=k}^\infty X_i$ is the infinite sequence of canonical elements from X_k onwards. In this way, after each finite play \vec{x} we check whether the canonical extension of \vec{x} is such that its relevant part is an initial segment of \vec{x} . If so, we have already played past the relevant part of the game and we can stop, since we only require that the first $n = l(q(\alpha))$ moves be optimal. All other (non-relevant) moves might not necessarily be optimal.

Although this sort of unbounded finite game seems less natural than the one introduced in Definition 13, we show in the following section that it does appear naturally in proof theory, from computational interpretations of ineffective principles (see Section 7 (f)).

We now show how to compute plays which have an optimal relevant part. First, we must adapt the notion of optimal strategy to take into account the explicit control function $l: R \rightarrow \mathbb{N}$.

Definition 17 (Optimal strategy for variant of unbounded game). *A strategy $(\text{next}_k)_{k \in \mathbb{N}}$ is said to be optimal if for every partial play $\vec{a} = a_0, \dots, a_{k-1}$ such that $k \leq l(q(\vec{a} * \beta^{\vec{a}}))$ we have*

$$q(\vec{a} * \beta^{\vec{a}}) \in \phi_k(\lambda x_k. q(\vec{a} * x_k * \beta^{\vec{a}, x_k})).$$

[†] For instance, if $R \equiv R' \times \mathbb{N}$ we could take $l(\cdot)$ to be the second projection.

In particular, notice that if next_k is an optimal strategy in the above sense, then for every partial play $\vec{a} = a_0, \dots, a_{k-1}$ we have

$$q(\vec{a} * \beta^{\vec{a}}) \in \phi_j(\lambda x_j. q(\vec{a} * \vec{b} * x_j * \beta^{\vec{a}, \vec{b}, x_j})).$$

for all $k \leq j \leq l(q(\vec{a} * \beta^{\vec{a}}))$, where \vec{b} is the strategic play of length $j - k$, starting from \vec{a} . This is obtained by a simple iteration, always taking \vec{a} to be longer and longer along the strategic play, until the condition $k \leq l(q(\vec{a} * \beta^{\vec{a}}))$ no longer holds. As in Section 5 and 6, we now define a slightly different iterated product of selection functions that computes such optimal strategies for unbounded games with explicit control.

Definition 18 (Unbounded product of selection functions with explicit control). *Let $l(\cdot): R \rightarrow \mathbb{N}$ be fixed. Given an infinite sequence of selection functions $\varepsilon_i: J_R X_i$ and an outcome function $q: \prod_{i=0}^{\infty} X_i \rightarrow R$, we define the unbounded product of $(\varepsilon_i)_{i \in \mathbb{N}}$ by simply iterating the binary product of selection functions as*

$$\left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right) (q) \prod_{i=k}^{\infty} X_i \begin{cases} \mathbf{c}^k & \text{if } l(q(\mathbf{c}^k)) < k \\ (\varepsilon_k \otimes \left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right)) (q) & \text{otherwise.} \end{cases}$$

The following theorem shows how (as in Theorems 5.4 and 6.2) the product of selection functions defined above computes optimal strategies for unbounded games with explicit control.

Theorem 6.4 (Main theorem for unbounded games with explicit control). *Let an unbounded sequential game with explicit control $(R, (X_i)_{i=0}^{\infty}, (\phi_i)_{i=0}^{\infty}, q, l)$ be given. If the quantifiers ϕ_i have associated selection functions ε_i then an optimal strategy in the game can be computed as*

$$\text{next}_k(\vec{x}) = \pi_0 \left(\left(\bigotimes_{i=k}^{\infty} \varepsilon_i \right) (q_{\vec{x}}) \right),$$

where $\vec{x} = x_0, \dots, x_{k-1}$.

Proof. Let $\vec{x} = x_0, \dots, x_{k-1}$ and assume $l(q(\vec{x} * \beta^{\vec{x}})) \geq k$. First, note that this implies $l(q(\vec{x} * \mathbf{c})) \geq k$, as otherwise, by the definition of the strategy next_k we would have that $\beta^{\vec{x}} = \mathbf{c}$, and get a contradiction. Hence, by the definition of the strategic move at round k we have

$$\beta^{\vec{x}}(k) = \text{next}_k(\vec{x}) = \varepsilon_k \left(\lambda x_k. q_{\vec{x}, x_k} \left(\left(\bigotimes_{i=k+1}^{\infty} \varepsilon_i \right) (q_{\vec{x}, x_k}) \right) \right).$$

By a variant of Lemma 6.1 to the explicitly controlled product, each component of the product of selection functions above corresponds to a strategic move from the initial partial play \vec{x}, x_k , i.e.

$$\beta^{\vec{x}}(k) = \varepsilon_k (\lambda x_k. q(\vec{x}, x_k * \beta^{\vec{x}, x_k})).$$

By the fact that ε_k is a selection function for ϕ_k we get

$$q(\vec{x}, \beta^{\vec{x}}(k) * \beta^{\vec{x}, \beta^{\vec{x}}(k)}) \in \phi_k(\lambda x_k. q(\vec{x}, x_k * \beta^{\vec{x}, x_k})).$$

The result follows since $q(\vec{x}, \beta^{\vec{x}}(k) * \beta^{\vec{x}, \beta^{\vec{x}}(k)}) = q(\vec{x} * \beta^{\vec{x}})$. \square

As a corollary, we obtain that the following set of equations can be solved via the product of selection functions. The following is another possible generalisation of Corollary 5.5. Since we no longer require R to be discrete, but assume a clock function instead, Corollary 6.5 seems to be orthogonal to Corollary 6.3.

Corollary 6.5. *Let an unbounded game $(R, (X_i)_{i=0}^\infty, (\phi_i)_{i=0}^\infty, q, l)$ with explicit control be given. If the quantifiers ϕ_i have associated selection functions ε_i then there are functions $p_i: X_i \rightarrow R$ such that*

$$\begin{aligned} \alpha(i) &\stackrel{X_i}{=} \varepsilon_i(p_i) \\ q(\alpha) &\in \phi_i(p_i), \end{aligned}$$

for $0 \leq i \leq l(q(\alpha))$, where α is the ε -strategic play arising from the optimal strategy described in Theorem 6.4.

Proof. Let

$$p_k(x_k) = \left(\bigotimes_{i=k+1}^\infty \varepsilon_i \right) (q_{\alpha(0), \dots, \alpha(k-1), x_k}).$$

By definition $\alpha(k) = \text{next}_k(\alpha(0), \dots, \alpha(k-1))$, i.e.

$$\alpha(k) = \varepsilon_k \left(\lambda x_k. q_{\vec{x}, x_k} \left(\left(\bigotimes_{i=k+1}^\infty \varepsilon_i \right) (q_{\vec{x}, x_k}) \right) \right) = \varepsilon_k(p_k).$$

Moreover, since next_k is an optimal strategy we have

$$\begin{aligned} q(\alpha) &= q((\alpha)[k] * \beta^{(\alpha)[k]}) \\ &\in \phi_k(\lambda x_k. q((\alpha)[k] * x_k * \beta^{(\alpha)[k], x_k})) \\ &\in \phi_k(\lambda x_k. q((\alpha)[k] * x_k * (\bigotimes_{i=k+1}^\infty \varepsilon_i) (q_{(\alpha)[k], x_k}))) \\ &= \phi_k(p_k), \end{aligned}$$

for all $k \leq l(q(\alpha))$. □

7. Selection Functions Everywhere

In this section we show how these natural notions of finite (Definitions 3 and 8) and unbounded (Definitions 13 and 16) sequential games, and the corresponding computation of optimal strategies, has appeared in several different areas. We start each subsection by defining a particular instance of the game, fixing the the set of outcomes R , the sets of moves X_i , the (multi-valued) quantifiers ϕ_i and the outcome function $q: \prod_{i=0}^{n-1} X_i \rightarrow R$. After each instantiation a discussion will follow showing how the derived optimal strategy, in case ϕ_i are attainable, corresponds to a known concept in a particular field of research. In some cases we will assume the reader is familiar with particular notions of each field, as it would be difficult to give all background necessary for each particular application discussed.

(a) Algorithms – Backtracking

Consider first the following simple instance of the finite game with single-valued quantifiers (Definition 3).

- **Outcomes and moves.** Let the set of outcomes and the set of moves at each round be the booleans $\mathbb{B} = \{\mathbf{tt}, \mathbf{ff}\}$, i.e. $R = X_i = \mathbb{B}$ for all $0 \leq i < n$.
- **Quantifiers.** The quantifiers $\phi_i: (X_i \rightarrow \mathbb{B}) \rightarrow \mathbb{B}$ are given by the existential quantifier

$$\phi_i(p^{X_i \rightarrow \mathbb{B}}) \stackrel{\mathbb{B}}{=} \exists x \in X_i p(x).$$

- **Outcome function.** For the outcome function take a boolean formula on n variables, i.e. $q: \mathbb{B}^n \rightarrow \mathbb{B}$.

First, note that the optimal outcome

$$w = \left(\bigotimes_{i=0}^{n-1} \phi_i \right) (q) = \exists x_1 \in X_1 \dots \exists x_{n-1} \in X_{n-1} q(x_0, \dots, x_{n-1})$$

calculates whether the formula q is satisfiable or not, whereas an optimal play \vec{a} returns a satisfying assignment which makes q true in case the formula is satisfiable. Note also that

$$\varepsilon_i(p^{X_i \rightarrow \mathbb{B}}) \stackrel{X_i}{=} \begin{cases} \mathbf{tt} & \text{if } p(\mathbf{tt}) = \mathbf{tt} \\ \mathbf{ff} & \text{otherwise} \end{cases}$$

are selection functions for $\phi_i: (X_i \rightarrow \mathbb{B}) \rightarrow \mathbb{B}$, i.e. given any $p: X_i \rightarrow \mathbb{B}$ we have

$$p(\varepsilon_i(p)) = \phi_i(p).$$

By Theorem 5.4 the product of selection functions computes an optimal play, i.e. a satisfying assignment in case the formula is satisfiable. Observing how the product of selection functions operates in this particular case, we note that the computation of the satisfying assignment is done *backtracking*. Each individual selection function ε_i performs an exhaustive search on the space \mathbb{B} in order to find a witness for $p: X_i \rightarrow \mathbb{B}$, when put together, the product of selection function is then able to perform an exhaustive search on the product space \mathbb{B}^n .

We note, however, that a search only happens when the formula q queries a particular variable x_i . Therefore, if the value of q is already decided given a few particular values the backtracking terminates that search branch and tries a different branch. In other words, in this particular case the products of selection functions performs backtracking with “automatic” pruning.

(b) Fixed point theory – Bekič’s lemma

Consider the following instance of the finite game with multi-valued quantifiers (Definition 8).

- **Outcomes and moves.** The sets of moves X_i are fixed sets each endowed with a fixed point operator

$$\text{fix}_i: (X_i \rightarrow X_i) \rightarrow X_i.$$

We are assuming now to be working in some particular model or “domain” where such fixed point operators exist. The set of outcomes will be the product space $R \equiv \prod_{i=0}^{n-1} X_i$.

- **Quantifiers.** The multi-valued quantifiers $\phi_i: (X_i \rightarrow R) \rightarrow 2^R$ are given as

$$\phi_i(p^{X_i \rightarrow R}) = \{r \in R : r_i = \text{fix}_i(\pi_i \circ p)\}$$

where $\pi_i: R \rightarrow X_i$ is the standard i -th projection mapping.

- **Outcome function.** The outcome function $q: \prod_{i=0}^{n-1} X_i \rightarrow R$ ($\equiv \prod_{i=0}^{n-1} X_i \rightarrow \prod_{i=0}^{n-1} X_i$) is a given fixed self-map on the product space.

First we note that $\varepsilon_i(p^{X_i \rightarrow R}) = \text{fix}_i(\pi_i \circ p)$ are selection functions for the multi-valued quantifiers $\phi_i: (X_i \rightarrow R) \rightarrow 2^R$, i.e. given any $p: X_i \rightarrow R$ we have

$$p(\text{fix}_i(\pi_i \circ p)) \in \phi_i(p).$$

By Theorem 5.4 we can compute an optimal strategy in this game as

$$\text{next}_k(x_0, \dots, x_{k-1}) = \pi_0 \left(\left(\bigotimes_{i=k}^{n-1} \varepsilon_i \right) (q_{x_0, \dots, x_{k-1}}) \right).$$

This optimal strategy induces an strategic play \vec{a} which, by Corollary 5.5, satisfies

$$\begin{aligned} a_k &= \varepsilon_k(p_k) \\ q(\vec{a}) &\in \phi_k(p_k), \end{aligned}$$

for the given family of functions

$$p_k(x_k) = \overline{\left(\bigotimes_{i=k+1}^{n-1} \varepsilon_i \right)} (q_{a_0, \dots, a_{k-1}, x_k}).$$

Unwinding the definitions of ϕ_i and ε_i we in fact have

$$\begin{aligned} a_k &= \text{fix}_k(p_k \circ \pi_k) \\ (q(\vec{a}))_k &= \text{fix}_k(p_k \circ \pi_k). \end{aligned}$$

Hence, $a_k = (q(\vec{a}))_k$, for all $0 \leq k < n$, which is equivalent to say that $q(\vec{a}) = \vec{a}$. I.e. the tuple \vec{a} is a fixed point for the mapping $q: \prod_{i=0}^{n-1} X_i \rightarrow \prod_{i=0}^{n-1} X_i$.

What we have shown is that in this particular instance of the game, the optimal strategy induces an strategic play which computes a fixed point for a mapping on the product space given a family of fixed point operators for each of the individual spaces X_i . Moreover, the whole construction shows that if each space X_i has a fixed point operator, then the product space $\prod_{i=0}^{n-1} X_i$ must also have a fixed point operator. That is the essential content of Bekič’s lemma (Bekič 1984), a celebrated result in fixed point theory, and frequently used in domain theory.

(c) Game theory – Nash equilibrium

Consider now another instance of the finite game with multi-valued quantifiers (Definition 8) as follows:

- **Outcomes and moves.** Fix sets X_i as the sets of moves, which for simplicity we assume are finite. For the set of outcomes take n -tuples of reals, i.e. $R \equiv \mathbb{R}^n$.
- **Quantifiers.** The multi-valued quantifiers $\phi_i: (X_i \rightarrow R) \rightarrow 2^R$ are given as

$$\phi_i(p^{X_i \rightarrow R}) = \{r \in R : r_i = \max_{x_i \in X_i} \pi_i(p(x_i))\}$$

where $\pi_i: R \rightarrow X_i$ is the standard i -th projection mapping. I.e. ϕ_i returns all the elements in the co-domain of p which have maximal value at coordinate i .

- **Outcome function.** The outcome function $q: \prod_{i=0}^{n-1} X_i \rightarrow \mathbb{R}^n$ is a given mapping from plays to n -tuples of real numbers.

First, since all quantifiers are different, we can think of this as an n -player game. Moreover, the outcome function can be thought of as calculating the payoff each of the n players gets at the end of the game. The quantifiers says that all players are trying to maximise their own payoffs, without paying attention to what impact this might have on other players. Note also that

$$\varepsilon_i(p^{X_i \rightarrow \mathbb{R}^n}) \stackrel{X_i}{=} \operatorname{argmax}(\pi_i \circ p)$$

are selection functions for the multi-valued quantifiers $\phi_i: (X_i \rightarrow \mathbb{R}^n) \rightarrow 2^{\mathbb{R}^n}$, i.e. given any $p: X_i \rightarrow \mathbb{R}^n$ we have

$$p(\operatorname{argmax}(\pi_i \circ p)) \in \phi_i(p).$$

By Theorem 5.4 we can compute an optimal strategy in this game as

$$\operatorname{next}_k(x_0, \dots, x_{k-1}) \stackrel{X_k}{=} \pi_0 \left(\left(\bigotimes_{i=k}^{n-1} \varepsilon_i \right) (q_{x_0, \dots, x_{k-1}}) \right).$$

A strategy being optimal means

$$q(\vec{a}, b_k^{\vec{a}}, \dots, b_{n-1}^{\vec{a}}) \in \phi_k(\lambda x_k. q(\vec{a}, x_k, b_{k+1}^{\vec{a}, x_k}, \dots, b_{n-1}^{\vec{a}, x_k})),$$

which, unwinding the definition of ϕ_k gives

$$\pi_k(q(\vec{a}, b_k^{\vec{a}}, \dots, b_{n-1}^{\vec{a}})) = \max_{x_k \in X_k} \pi_k(q(\vec{a}, x_k, b_{k+1}^{\vec{a}, x_k}, \dots, b_{n-1}^{\vec{a}, x_k})).$$

This is saying that the move $b_k^{\vec{a}}$ suggested by the strategy leads to an outcome with maximal payoff for player k . It follows that the strategy next_k is in Nash equilibrium, as no player has any incentive to unilaterally change their strategy for round k . In this example, the construction of optimal strategies via products of selection functions corresponds to *backward induction*, a technique used in Game Theory (Nisan et al. 2007) to compute Nash equilibria in sequential games.

(d) *Higher-type computability – Searchable sets*

Consider now the following instance of the unbounded game (Definition 13), which generalises the finite game of Subsection (a).

- **Outcomes and moves.** Let R be the domain of boolean values \mathbb{B} and each X_i be subsets S_i of a fixed domain A .
- **Quantifiers.** Let the quantifiers $\phi_i: (S_i \rightarrow \mathbb{B}) \rightarrow \mathbb{B}$ be the existential quantifiers over the subsets $S_i \subseteq A$.
- **Outcome function.** For the outcome function we will take an arbitrary predicate on the product space of the S_i 's, i.e. $q: \prod_{i=0}^{n-1} S_i \rightarrow \mathbb{B}$.

First, we note that the assumption that ϕ_i are attainable correspond in this case to the sets S_i being *searchable*, as defined in (Escardó 2008).

It turns out that any searchable set (of total elements) is topologically compact, and, mimicking the Tychonoff theorem from topology, it was shown in (Escardó 2008) that searchable sets are closed under countable products. For that one must essentially show that the game above defined has an optimal strategy, and that the optimal play witnesses q whenever q is a non-empty predicate.

In fact, the construction used in (Escardó 2008) is a particular case of the iterated product of selection function (Definition 15), and it was precisely this observation that led us to look at this construction in more details.

(e) *Proof theory – Arithmetic*

In the next examples we show how the finite product of selection functions gives a computational interpretation to *arithmetic*, whereas the unbounded product interprets full classical *analysis*. Consider the principle of bounded collection for existential formulas, i.e.

$$\forall b \exists n \leq t \forall m \leq b A(n, m) \rightarrow \exists n \leq t \forall m A(n, m).$$

where $A(n, m)$ is a Σ_1^0 -formula. It is well-known that such principle lies in between Σ_2^0 -induction and Σ_1^0 -induction, see (Parsons 1970, Kohlenbach 2008). For simplicity, let us consider the case when $A(n, m) = \exists k A_0(n, m, k)$ where $A_0(n, m, k)$ is quantifier-free, i.e.

$$\forall b \exists n \leq t \forall m \leq b \exists k A_0(n, m, k) \rightarrow \exists n \leq t \forall m \exists k A_0(n, m, k).$$

The negative translation (assuming Markov principle) of this principle is equivalent to

$$\forall b \exists n \leq t \forall m \leq b \exists k A_0(n, m, k) \rightarrow \neg \neg \exists n \leq t \forall m \exists k A_0(n, m, k).$$

Its dialectica interpretation, see (Avigad & Feferman 1998), is

$$\exists f, g \forall b (fb \leq t \wedge \forall m \leq b A_0(fb, m, gbm)) \rightarrow \forall \varepsilon \exists n \leq t \exists p A_0(n, \varepsilon_n p, p(\varepsilon_n p)).$$

In other words, given f, g and ε we must produce n, b and p such that

$$\forall b (fb \leq t \wedge \forall m \leq b A_0(fb, m, gbm)) \rightarrow (n \leq t \wedge A_0(n, \varepsilon_n p, p(\varepsilon_n p))).$$

This boils down to producing n, b, p given f, g, ε_n such that for some $m \leq b$ we have

$$\begin{aligned} n &= fb \\ m &= \varepsilon_n p \\ gbm &= p(\varepsilon_n p) \end{aligned} \tag{7.1}$$

assuming f is bounded by t . We show that this can be solved using the optimal strategy of the following finite game. Let f, g and $(\varepsilon_n)_{n \leq t}$ be given.

- **Outcomes and moves.** The sets of moves X_i and the set of outcomes R is taken to be the natural numbers \mathbb{N} .
- **Quantifiers.** The quantifiers $\phi_i: (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ are defined from the given selection functions

$$\phi_i(p) = \bar{\varepsilon}_i p = p(\varepsilon_i(p))$$

so that ϕ_i is attainable by definition.

- **Outcome function.** The outcome function $q: \mathbb{N}^{t+1} \rightarrow \mathbb{N}$ is taken to be

$$q(\vec{x}) = g(\max \vec{x})(x_{f(\max \vec{x})}).$$

Given that the quantifiers are attainable, we can compute the optimal strategy of this game, and its respective strategic play

$$\vec{a} = \left(\bigotimes_{i=0}^t \varepsilon_i \right) (q).$$

By Corollary 5.5 we have that there are functions $p_n: \mathbb{N} \rightarrow \mathbb{N}$ such that

$$\begin{aligned} a_n &= \varepsilon_n(p_n) \\ q(\vec{a}) &= \phi_n(p_n) = p_n(\varepsilon_n(p_n)), \end{aligned}$$

for $0 \leq n \leq t$. Take $b = \max \vec{a}$ and $n = fb$ (because f is bounded by t we have $n \leq t$) and $p = p_n$ and $m = a_n$. We show that this solves the set of equations (7.1). By definition $n = fb$ and $m = a_n = \varepsilon_n p_n = \varepsilon_n p$. By the definition of q we also have

$$p_n(\varepsilon_n(p_n)) = q(\vec{a}) = g(\max \vec{a})(a_{f(\max \vec{a})}) = gba_{fb} = gba_n = gbm.$$

This is essentially the solution given in (Oliva 2006) for the dialectica interpretation of the infinite pigeon-hole principle, but here it is given an intuitive game-theoretic explanation.

(f) Proof theory – Analysis

The product of selection functions also appears in proof theory, in the form of *bar recursion* (Spector 1962). More precisely, in order to extend Gödel's consistency proof (Gödel 1958) from arithmetic to analysis, Spector arrived at the following

system of equations: Given a family of (selection) functions $\varepsilon_i: J_Y X$ and functions $r: X^\omega \rightarrow Y$ and $\omega: X^\omega \rightarrow \mathbb{N}$ find $p: X \rightarrow Y$ and $\alpha: X^\omega$ such that

$$\begin{aligned} i & \stackrel{\mathbb{N}}{=} \omega(\alpha) \\ \alpha(i) & \stackrel{X}{=} \varepsilon_i(p) \\ p(\alpha(i)) & \stackrel{Y}{=} r(\alpha). \end{aligned}$$

We show now that this system of equations can be solved from the optimal strategy of the following unbounded game with explicit control (Definition 16):

- **Outcomes and moves.** Let $R = Y \times \mathbb{N}$ and $X_i = X$.
- **Quantifiers.** The quantifiers $\phi_i: (X \rightarrow R) \rightarrow 2^R$ are defined from the ε_i 's as

$$\phi_i(p^{X \rightarrow Y \times \mathbb{N}}) \stackrel{Y \times \mathbb{N}}{=} \{p(\varepsilon_i(\pi_0 \circ p))\}.$$

- **Outcome function.** The outcome function $q: \prod_{i=0}^{n-1} X_i \rightarrow R$ is

$$q(\alpha) = (r(\alpha), \omega(\alpha))$$

where $r: X^\omega \rightarrow Y$ and $\omega: X^\omega \rightarrow \mathbb{N}$ are as given in Spector's equations.

- **Clock function.** Let $l: R \rightarrow \mathbb{N}$ be the second projection i.e. $l(\alpha, n) = n$.

By Corollary 6.5 we have that for the strategic play $\alpha: X^\omega$ there are $p_i: X \rightarrow Y \times \mathbb{N}$ such that

$$\begin{aligned} \alpha(i) & \stackrel{X}{=} \varepsilon_i(\pi_0 \circ p_i) \\ q(\alpha) & \in \phi_i(p_i), \end{aligned}$$

for $0 \leq i \leq l(q(\alpha))$, where α is the ε -strategic play. Unwinding the definitions of ϕ, q and l , and taking $p_i^*(x) = \pi_0(p_i(x))$ we have

$$\begin{aligned} \alpha(i) & \stackrel{X}{=} \varepsilon_i(p_i^*) \\ r(\alpha) & \stackrel{Y}{=} p_i^*(\varepsilon_i(p_i^*)), \end{aligned}$$

for $0 \leq i \leq \omega(\alpha)$, which solves Spector's equations taking $i = \omega(\alpha)$ and $p = p_i^*$. In other words, the notion required to extend Gödel's consistency proof from arithmetic to analysis was the computation of optimal strategies for unbounded games with explicit control.

We have shown that in fact Spector's bar recursion is primitive recursive equivalent to the product of selection functions from Definition 18, and that a different form of bar recursion (Berardi, Bezem & Coquand 1998, Berger & Oliva 2005, Berger & Oliva 2006) is primitive recursive equivalent to the product of selection functions from Definition 15. The second form of bar recursion arose from a game computational interpretation of classical logic (Coquand 1995), and it would be interesting to investigate in further details how our notion of games corresponds to the one used to interpret classical logic. See also (Aczel 2001) for more connections between strong monads and classical logic.

(g) *Functional programming*

We have also shown in (Escardó & Oliva 2010c) that the construction J_R over any cartesian closed category gives rise to a strong monad, with the monad morphism

$$\overline{(\cdot)}: J_R \rightarrow K_R$$

into the well-known continuation monad K_R (Griffin 1990). This monad morphism assigns the quantifier $\bar{\varepsilon} \in K_R A$ defined by Equation (5.1) to a given selection function $\varepsilon \in J_R A$. Moreover, the case $n = 2$ of the product of selection functions turns out to be simply the canonical map that makes any strong monad into a monoidal monad.

Monads are widely used in programming language semantics as a way to interpret side-effects (Moggi 1991) and in functional programming, particularly in the language Haskell, more generally as a way of structuring programs. As it turns out – see (Escardó & Oliva 2010d) – the ability to iterate this binary product into an infinite product is already built into the Haskell language, once the selection monad is defined. All constructions described here can be easily implemented in functional programming, as done in (Escardó & Oliva 2010d). In fact, experimental results show that the calculation of optimal strategies using the iterated product of selection functions is computationally very efficient with respect to both time and space.

8. Games in Normal and Extensive Form

The games discussed in this paper are presented in what is known as *normal form* in the literature (von Neumann & Morgenstern 1944). This means that one is given an outcome function q that associates outcomes to plays (sequences of moves). For games presented in *extensive form* one organises the set of all possible plays as a decision tree, as also discussed by (von Neumann & Morgenstern 1944), who show that the two presentations are mathematically equivalent. For each node of the tree there is a set of edges labelled by the moves that can follow that node, leading to subtrees. Each leaf of the tree is labelled by an outcome. Given a sequence of moves, one follows the tree starting from the root until a leaf is obtained, from which one reads off the outcome of the play. The presentation of games in extensive form allows one to see that the n -ary product of selection functions (Definition 12) corresponds to backward induction, as discussed in Section 7 (c).

For games of fixed, finite length n , all leaves are at level n of the tree. For a game of unbounded, finite length, leaves can occur at any level of the tree, but it is required that all paths eventually lead to a leaf. Trees which enjoy this property are called *well founded*. The well-foundedness condition for games in extensive form is equivalent to our continuity condition (6.1) for games in normal form.

Note that in the finite case, the optimal play $(\bigotimes_{i < n} \varepsilon_i)(q)$ of the game q can be seen as defined by induction on n , but this does not make sense for the infinite product of selection functions. In both the finite and infinite case, for q continuous, the outcome $(\bigotimes_i \varepsilon_i)(q)$ can be seen as defined by structural induction on the well-founded tree that defines q in extensive form (see (Escardó & Oliva 2010c) for details). Thus, curiously, in the finite case, the same construction of $(\bigotimes_{i < n} \varepsilon_i)(q)$

can be seen at the same time as a definition by induction on n and as a definition by induction on the extensive form of q .

The above discussion shows that bar recursion, mentioned in Section 7 (f), amounts to recursion on well-founded trees, where the trees are indirectly presented as continuous functions $q: \prod_n X_n \rightarrow R$.

The authors would like to thank the two anonymous referees for their detailed comments and suggestions which greatly improved the presentation of the paper. The second author also gratefully acknowledges support of The Royal Society under grant 516002.K501/RH/kk.

References

- Aczel, P. (2001), ‘The Russell-Prawitz modality’, *Math. Structures Comput. Sci.* **11**(4), 541–554. Modalities in type theory (Trento, 1999).
- Avigad, J. & Feferman, S. (1998), Gödel’s functional (“Dialectica”) interpretation, in S. R. Buss, ed., ‘Handbook of proof theory’, Vol. 137 of *Studies in Logic and the Foundations of Mathematics*, North Holland, Amsterdam, pp. 337–405.
- Bekič, H. (1984), *Programming languages and their definition*, Vol. 177 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin. Selected papers edited by C. B. Jones.
- Berardi, S., Bezem, M. & Coquand, T. (1998), ‘On the computational content of the axiom of choice’, *The Journal of Symbolic Logic* **63**(2), 600–622.
- Berger, U. & Oliva, P. (2005), ‘Modified bar recursion and classical dependent choice’, *Lecture Notes in Logic* **20**, 89–107.
- Berger, U. & Oliva, P. (2006), ‘Modified bar recursion’, *Mathematical Structures in Computer Science* **16**, 163–183.
- Bezem, M. (1985), ‘Strongly majorizable functionals of finite type: a model for bar recursion containing discontinuous functionals’, *The Journal of Symbolic Logic* **50**, 652–660.
- Coquand, T. (1995), ‘A semantics of evidence for classical arithmetic’, *Journal of Symbolic Logic* **60**, 325–337.
- Escardó, M. H. (2008), ‘Exhaustible sets in higher-type computation’, *Logical Methods in Computer Science* **4**(3), paper 4.
- Escardó, M. H. & Oliva, P. (2010a), Computational interpretations of analysis via products of selection functions, in F. Ferreira, B. Lowe, E. Mayordomo & L. M. Gomes, eds, ‘Computability in Europe 2010, LNCS’, Springer, pp. 141–150.
- Escardó, M. H. & Oliva, P. (2010b), The Peirce translation and the double negation shift, in F. Ferreira, B. Löwe, E. Mayordomo & L. M. Gomes, eds, ‘Programs, Proofs, Processes - CiE 2010, LNCS 6158’, Springer, pp. 151–161.
- Escardó, M. H. & Oliva, P. (2010c), ‘Selection functions, bar recursion, and backward induction’, *Mathematical Structures in Computer Science* **20**(2), 127–168.
- Escardó, M. H. & Oliva, P. (2010d), ‘What sequential games, the Tychonoff theorem and the double-negation shift have in common’, *To appear: MSFP 2010 (ACM SIGPLAN Mathematically Structured Functional Programming)*.
- Gödel, K. (1958), ‘Über eine bisher nicht benützte Erweiterung des finiten Standpunktes’, *Dialectica* **12**, 280–287.
- Griffin, T. G. (1990), A formulae-as-types notion of control, in ‘17th Annual ACM Symp. on Principles of Programming Languages, POPL’90, San Francisco, CA, USA’, pp. 17–19.
- Kohlenbach, U. (2008), *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*, Monographs in Mathematics, Springer.
- Moggi, E. (1991), ‘Notions of computation and monads’, *Inf. Comput.* **1**, 55–92.
- Mostowski, A. (1957), ‘On a generalization of quantifiers’, *Fundamenta Mathematicae* **44**, 12–36.

- Nisan, N. et al. (2007), *Algorithmic Game Theory*, Cambridge University Press.
- Normann, D. (1999), The continuous functionals, in E. R. Griffor, ed., 'Handbook of Computability Theory', North Holland, Amsterdam, chapter 8, pp. 251–275.
- Oliva, P. (2006), Understanding and using Spector's bar recursive interpretation of classical analysis, in A. Beckmann, U. Berger, B. Löwe & J. V. Tucker, eds, 'Proceedings of CiE'2006, LNCS 3988', Springer, pp. 423–234.
- Parsons, C. (1970), On a number theoretic choice schema and its relation to induction, in 'Intuitionism and Proof Theory', North-Holland, pp. 459–473.
- Scarpellini, B. (1971), 'A model for bar recursion of higher types', *Compositio Mathematica* **23**, 123–153.
- Spector, C. (1962), Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics, in F. D. E. Dekker, ed., 'Recursive Function Theory: Proc. Symposia in Pure Mathematics', Vol. 5, American Mathematical Society, Providence, Rhode Island, pp. 1–27.
- von Neumann, J. & Morgenstern, O. (1944), *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, New Jersey.