Selection Equilibria of Higher-Order Games

Jules Hedges[1], Paulo Oliva[2]([✉]), Evguenia Shprits[3], Viktor Winschel[4], and Philipp Zahn[5]

[1] Department of Computer Science, University of Oxford, Oxford, UK
[2] Department of Electronic Engineering and Computer Science,
Queen Mary University of London, London, UK
p.oliva@qmul.ac.uk
[3] Department of Economics, University of Mannheim, Mannheim, Germany
[4] Department of Management, Technology and Economics,
ETH Zürich, Zürich, Switzerland
[5] Department of Economics, University of St. Gallen, St. Gallen, Switzerland

Abstract. In applied game theory the modelling of each player's intentions and motivations is a key aspect. In classical game theory these are encoded in the payoff functions. In previous work [2,4] a novel way of modelling games was introduced where players and their goals are more naturally described by a special class of higher-order functions called *quantifiers*. We refer to these as higher-order games. Such games can be directly and naturally implemented in strongly typed functional programming languages such as Haskell [3]. In this paper we introduce a new solution concept for such higher-order games, which we call *selection equilibrium*. The original notion proposed in [4] is now called *quantifier equilibrium*. We show that for a special class of games these two notions coincide, but that in general, the notion of selection equilibrium seems to be the right notion to consider, as illustrated through variants of coordination games where agents are modelled via fixed-point operators. This paper is accompanied by a Haskell implementation of all the definitions and examples.

1 Introduction

In this paper we introduce a representation of simultaneous move games that formally summarises the goals of agents via *quantifiers* and *selection functions*. Both quantifiers and selection functions are examples of higher-order functions (also called functionals or operators) and originate in a game-theoretic approach to proof theory [2,4].

As shown in [2,4], the standard Nash equilibrium concept can be seamlessly generalised to this higher-order representation of games. The original work on these higher-order games used a notion of equilibrium which we will now call *quantifier equilibrium*. In this paper we introduce an alternative notion, which we call *selection equilibrium*. We prove that quantifier and selection equilibria coincide in the case of the classical max and arg max operators, but that, generally, this equivalence does not hold: For other quantifiers and selection functions

the two different equilibrium concepts yield different sets of equilibria. We give a sufficient condition for the two notions to coincide based on the notion of *closedness* of selection functions. We prove that in general, the selection equilibrium is an equilibrium refinement of the quantifier equilibrium, and present evidence that for games based on non-closed selection functions, the selection equilibrium is the appropriate solution concept.

A Haskell implementation of the theory and examples contained in this paper is available online.[1] We chose not to discuss the actual code here in detail due to lack of space. This code, however, was crucial for the development of the theory here presented. The ability to implement not just the various players, but also the outcome functions and the equilibrium checkers, enabled us to quickly test several different examples of games, with different notions of equilibrium. Careful testing of a variety of situations ultimately led us to the conclusion that the new notion of equilibrium for higher-order games is preferable in general. Although in this paper we could have used any other strongly typed functional language, in the case of sequential games [2–4] the use of Haskell monads is essential. In higher-order *sequential* games one can make use of the fact that the type of selection functions forms a monad, and backwards induction can be simply implemented as the "sequencing" of monads.

2   Players, Quantifiers and Selection Functions

A *higher order function* (or *functional*) is a function whose domain is itself a set of functions. Given sets X and Y we denote by $X \to Y$ the set of all functions with domain X and codomain Y. There are familiar examples of higher-order functions, such as the max operator, which has type $\max\colon (X \to \mathbb{R}) \to \mathbb{R}$ returning the maximum value of a given real-valued function $p\colon X \to \mathbb{R}$. One will normally write $\max p$ as $\max_{x \in X} p(x)$. A corresponding operator is $\arg\max$ which returns all the points where the maximum of a function $p\colon X \to \mathbb{R}$ is attained, i.e. $\arg\max\colon (X \to \mathbb{R}) \to \mathcal{P}(X)$ using $\mathcal{P}(X)$ for the power-set[2] of X. Note that as opposed to max, $\arg\max$ is naturally a multi-valued function, even when the maximal value is unique.

Of a slightly different nature is the fixed point operator $\mathrm{fix}\colon (X \to X) \to \mathcal{P}(X)$ which calculates all the fixed points of a given self-mapping $p\colon X \to X$, or the anti-fixed-point operator which calculates all points that are not fixed points.

In this section we define two particular classes of higher-order functions: *quantifiers* and *selection functions*. We first establish that these functions provide means to represent agents' goals in an abstract and general way. In particular, these notions usefully generalise utility maximisation and preference relations.

---

[1] http://www.eecs.qmul.ac.uk/~pbo/papers/hog-padl-2017.hs.
[2] As long as our games are finite we can easily replace powersets with lists, which we do in the accompanying implementation.

2.1 Game Context

To define players' goals we first need a structure that represents the strategic situation on which these goals are based. To this end we introduce the concept of a *game context* which summarises information of the strategic situation from the perspective of a single player.

Definition 1 (Game context). *For a player $\mathcal{A}$ choosing a move from a set X, having in sight a final outcome in a set R, we call any function $p\colon X \to R$ a possible* game context *for the player $\mathcal{A}$.*

Consider the following *voting contest* which we will use as a running example throughout this paper: three judges are voting simultaneously for one of two contestants $X = \{A, B\}$. The winner is decided by the majority rule $\mathrm{maj}\colon X \times X \times X \to X$. In a setting where judges 1 and 3 have fixed their choices, say $x_1 = A$ and $x_3 = B$, this gives rise to a game context for the second judge, namely

$$x_2 \quad \mapsto \quad \mathrm{maj}(A, x_2, B)$$

which is in fact the identity function since $\mathrm{maj}(A, x_2, B) = x_2$. If, on the other hand, judges 1 and 3 had fixed their choices as $x_1 = x_3 = A$, the game context for player 2 would be the constant function $x_2 \mapsto A$, since his vote does not influence the outcome.

One can think of the game context $p\colon X \to R$ as an *abstraction* of the actual game context that is determined by knowing the rules of the game, and how each opponent played. Notice that in the example above the game context which maps A to B, and B to A, never arises. It would arise, however, if one replaced the majority rule by the minority rule.

It might seem like we are losing too much information by adopting such an abstraction. We hope that the examples given here will illustrate that this level of abstraction is sufficient for modelling players' individual motivations and goals. And precisely because it is abstract and it captures the strategic context of a player as if it was a single decision problem, it allows for a description of the players' intrinsic motivations, irrespective of how many players are around, or which particular game is being played. This is key for obtaining a modular description of games as well as a modular Haskell implementation.

2.2 Quantifiers and Selection Functions

Suppose now that $\mathcal{A}$ makes a decision $x \in X$ in a game context $p\colon X \to R$. First of all, it is important to realise that the only achievable outcomes in the context $p\colon X \to R$ are the elements in the image of p, i.e. $\mathrm{Im}(p) \subseteq R$. Out of these achievable outcomes the player should consider some outcomes to be *good* (or *acceptable*). Since the good outcomes must in particular be achievable, it is clear that the set of good outcomes can only be defined in relation to the given context. That dependence, however, can go further than just looking at $\mathrm{Im}(p)$. For instance, an element $r \in R$ might be the maximal attainable value of the

mapping $p_1\colon X \to R$, but could be unachievable, sub-optimal or even the worst outcome in a different context $p_2\colon X \to R$.

Definition 2 (Quantifiers, [2,4]). *Let $\mathcal{P}(R)$ denote the power-set of the set of outcomes R. We call* quantifiers[3] *any higher-order function of type*

$$\varphi\colon (X \to R) \to \mathcal{P}(R)$$

from contexts $p\colon X \to R$ to non-empty sets of outcomes $\varphi(p) \subseteq R$.

The approach of [2,4] is to model players $\mathcal{A}$ as quantifiers $\varphi_{\mathcal{A}}\colon (X \to R) \to \mathcal{P}(R)$ We think of $\varphi_{\mathcal{A}}(p) \subseteq R$ as the set of outcomes the player $\mathcal{A}$ considers preferable in a given game context $p\colon X \to R$. It is crucial to recognise that this is a *qualitative* description of a player, in the sense that an outcome is either preferable or it is not, with no numerical measure attached.

It could be, however, that the notion of being a "good outcome" indeed comes from a numeric measure. In fact, the classical example of a quantifier is *utility maximisation*, with the outcome set $R = \mathbb{R}^n$ consisting of n-tuples of real-valued payoffs. If we denote by $\pi_i\colon \mathbb{R}^n \to \mathbb{R}$ the i-projection, then the utility of the ith player is $\pi_i(r)$. Hence, given a game context $p\colon X \to \mathbb{R}^n$, the good outcomes for the ith player are precisely those for which the ith coordinate, i.e. his utility, is maximal. This quantifier is given by

$$i\text{-max}(p) = \{r \in \text{Im}(p) \mid r_i \geq (\pi_i \circ p)(x') \text{ for all } x' \in X\}$$

where $\text{Im}(p)$ denotes the image of the function $p\colon X \to \mathbb{R}^n$, and $\pi_i \circ p$ denotes the composition of p with the i-th projection.

For a very different example of a quantifier, when the set of moves is equal to the set of outcomes $R = X$ there is a quantifier whose good moves are precisely the fixpoints of the context. This quantifier models a player whose aim is to make a choice that is equal to the resulting outcome. If the context has no fixpoint, then the player will be equally satisfied with any outcome. Therefore such a quantifier can be defined as

$$\text{fix}(p) = \begin{cases} \{x \in X \mid p(x) = x\} & \text{if } p(x) = x \text{ for some } x \in X \\ X & \text{otherwise.} \end{cases}$$

Just as a quantifier tells us which outcomes a player considers good in each given context, one can also consider the higher-order function that determines which *moves* a player considers good in any given context.

---

[3] The terminology comes from the observation that the usual existential $\exists$ and universal $\forall$ quantifiers of logic can be seen as operations of type $(X \to \mathbb{B}) \to \mathbb{B}$, where $\mathbb{B}$ is the type of booleans. Mostowski [12] also called arbitrary functionals of type $(X \to \mathbb{B}) \to \mathbb{B}$ *generalised quantifiers*. We are choosing to generalise this further by replacing the booleans $\mathbb{B}$ with an arbitrary type R, and allowing for the operation to be multi-valued.

Definition 3 (Selection functions). *A selection function is any function of the form*[4]

$$\varepsilon \colon (X \to R) \to \mathcal{P}(X).$$

Similarly to quantifiers, the canonical example of a selection function is maximising one of the coordinates in $\mathbb{R}^n$, defined by

$$i\text{-}\arg\max(p) = \{x \in X \mid (\pi_i \circ p)(x) \geq (\pi_i \circ p)(x') \text{ for all } x' \in X\}.$$

Even in one-dimensional $\mathbb{R}^1$ the $\arg\max$ selection function is naturally multi-valued: a function may attain its maximum value at several different points.[5]

2.3   Relating Quantifiers and Selection Functions

It is clear that quantifiers and selection functions are closely related. One important relation between them is that of *attainment*. Intuitively this means that the outcome of a good move should be a good outcome.

Definition 4. *Given a quantifier $\varphi \colon (X \to R) \to \mathcal{P}(R)$ and a selection function $\varepsilon \colon (X \to R) \to \mathcal{P}(X)$, we say that ε attains φ iff for all contexts $p \colon X \to R$ it is the case that*

$$x \in \varepsilon(p) \implies p(x) \in \varphi(p).$$

One can check that the attainability relation holds between the quantifier i-max and the selection function i-$\arg\max$. Any point where the maximum value is attained will evaluate to the maximum value of the function. More interestingly, the fixpoint quantifier is also a selection function, and it attains itself since

$$x \in \mathrm{fix}(p) \implies p(x) \in \mathrm{fix}(p).$$

Let us briefly reflect on the game theoretic meaning of attainability. Suppose we have a quantifier φ which describes the outcomes that a player considers to be good. The quantifier might be *unrealistic* in the sense that it has no attainable good outcome. For example, a player may consider it a good outcome if he received a million dollars, but in his current context there may just not be a move available which will lead to this outcome. The attainable quantifiers $\varphi \colon (X \to R) \to \mathcal{P}(R)$ describe realistic players, i.e. for any game context $p \colon X \to R$ there is always a move $x \colon X$ which leads to a good outcome $p(x) \in \varphi(p)$.

Given any selection function $\varepsilon \colon (X \to R) \to \mathcal{P}(X)$, we can form the smallest quantifier which it attains as follows.

---

[4] Where selection functions have been considered previously [2,4] the focus was on single-valued ones. However, as multi-valued selection functions are extremely important in our examples we have adapted the definitions accordingly.

[5] In the following we will assume that quantifiers and selection functions are non-empty. That is, agents will always have a preferred outcome, respectively move, in all situations they have to make a decision. See [9] for a discussion.

Definition 5. *Given a selection function* $\varepsilon\colon (X \to R) \to \mathcal{P}(X)$, *define the quantifier* $\overline{\varepsilon}\colon (X \to R) \to \mathcal{P}(R)$ *as*

$$\overline{\varepsilon}(p) = \{p(x) \mid x \in \varepsilon(p)\}.$$

It is easy to check that ε attains $\overline{\varepsilon}$. Conversely, given any quantifier we can define a corresponding selection function as follows.

Definition 6. *Given a quantifier* $\varphi\colon (X \to R) \to \mathcal{P}(R)$, *define the selection function* $\overline{\varphi}\colon (X \to R) \to \mathcal{P}(X)$ *as*

$$\overline{\varphi}(p) = \{x \mid p(x) \in \varphi(p)\}.$$

Again, it is easy to check that the selection function $\overline{\varphi}$ attains the quantifier φ. We use the same overline notation, as it will be clear from the setting whether we are applying it to a quantifier or a selection function.

3   Higher-Order Games

Quantifiers and selection functions as introduced in the previous section can be used to model games. In this section we define higher-order games, illustrate the definition using the voting contest as a running example, and, lastly, discuss two equilibrium concepts.

Definition 7 (Higher-Order Games). *An n-players game* $\mathcal{G}$, *with a set* R *of outcomes and sets* X_i *of strategies for the* i^{th} *player, consists of an* $(n+1)$-*tuple* $\mathcal{G} = (\varepsilon_1, \ldots, \varepsilon_n, q)$ *where*

– *for each player* $1 \le i \le n$, $\varepsilon_i\colon (X_i \to R) \to \mathcal{P}(X_i)$ *is a selection function describing the i-th player's preferred moves in each game context.*
– $q\colon \prod_{i=1}^{n} X_i \to R$ *is the outcome function, i.e., a mapping from the strategy profile to the final outcome.*

(Note that a strategy profile for a game is simply a tuple $\mathbf{x}\colon \prod_{i=1}^{n} X_i$, consisting of a choice of strategy for each player.)

Intuitively, we think of the outcome function q as representing the 'situation', or the rules of the game, while we think of the selection functions as describing the players. Thus we can imagine the same player in different situations, and different players in the same situation. This allows us to decompose a modelling problem into a global and a local part: modelling the situation (i.e. q) and modelling the individual players (i.e. $\varepsilon_1, \ldots, \varepsilon_n$).

Remark 1 (Classical Game [13]). The ordinary definition of a normal form game of n-players with standard payoff functions is a particular case of Definition 7 when

– for each player i the set of strategies is X_i,

Table 1. Voting contest with classical players; Nash equilibria in bold.

	J_2: A	J_2: B			J_2: A	J_2: B	
J_1: A	**1,1,0**	1,1,0	J_3: A	J_1: A	**1,1,0**	0,0,1	J_3: B
J_1: B	1,1,0	0,0,1		J_1: B	0,0,1	**0,0,1**	

– the set of outcomes R is $\mathbb{R}^n$, modelling the vector of payoffs obtained by each player,
– the selection function of player i is i-arg max: $(X_i \rightarrow \mathbb{R}^n) \rightarrow \mathcal{P}(X_i)$, i.e. arg max with respect to the ith coordinate, representing the idea that each player is solely interested in maximising their own payoff,
– the ith component of the outcome function $q\colon \prod_{i=1}^{n} X_i \rightarrow \mathbb{R}^n$ can be viewed as the payoff function $q_i\colon \prod_{j=1}^{n} X_j \rightarrow \mathbb{R}$ of the ith player.

Remark 2. For an implementation in a simply-typed language (as opposed to dependently typed) such as Haskell, it is convenient either to fix the number of players and store the data in tuples, or to take the sets X_i to be equal and store the data in homogeneous lists. (See also [1].) In the accompanying implementation we opt for the latter, because in our running example the X_i are equal.

3.1 Example: Voting Contest

Reconsider the voting contest outlined in Sect. 2.1: There are three players, the judges $J = \{J_1, J_2, J_3\}$, who each vote for one of two contestants A or B. The winner is determined by the simple majority rule. We analyse two instances of this game with different motivations of players while keeping the overall structure of the game fixed.

Classical Players. Suppose the judges rank the contestants according to a preference ordering. Say judges 1 and 2 prefer A and judge 3 prefers B. Table 1 depicts a payoff matrix which encodes this situation, including the rules for choosing a winner (majority) and the goals of each individual player. The two separate tables show the cases when judge 3 has played either A (left table) or B (right table). Within each table, we also have the four possibilities for the voting of judge 2 (columns) and judge 1 (rows). A numeric value such as 1, 1, 0 says that in that particular play judges 1 and 2 got payoff 1, but judge 3 got payoff 0.

How is such a game modelled following Definition 7? The set of strategies in this case is the same as the set of possible outcomes, i.e. $X_i = R = \{A, B\}$. The outcome function $q\colon X_1 \times X_2 \times X_3 \rightarrow R$ is the majority function maj: $X \times X \times X \rightarrow X$, e.g. $\mathrm{maj}(A, B, B) = B$. It remains for us to find suitable selection functions representing the goals of the three players. Consider two order relations on X, call it $B \prec' A$ and $A \prec'' B$. The judges wish to maximise the final outcome

Table 2. Voting contest with Keynesian players; Nash equilibria in bold.

	J_2: A	J_2: B	
J_1: A	**1,1,1**	1,0,1	J_3: A
J_1: B	**1,1,1**	0,1,0	

	J_2: A	J_2: B	
J_1: A	1,1,0	**0,1,1**	J_3: B
J_1: B	0,0,1	**0,1,1**	

with respect to their preferred ordering. Hence the three selection functions are

$$\varepsilon_1 = \varepsilon_2 = \prec'\text{-arg max}$$
$$\varepsilon_3 = \prec''\text{-arg max}.$$

Therefore, the game is described by the tuple of higher-order functionals

$$\mathcal{G} = (\prec'\text{-arg max}, \prec'\text{-arg max}, \prec''\text{-arg max}, \text{maj}).$$

Keynesian Players. Now, consider the case where the first judge J_1 still ranks the candidates according to a preference ordering $B \prec A$. The second and third judges, however, have no preference relations over the candidates per se, but want to vote for the winning candidate. They are *Keynesian*[6] players.

It is possible to model such a game via standard payoff matrices, and Table 2 presents such an encoding. If there is a majority for a candidate and player J_2 or J_3 votes for the majority candidate they will get a certain payoff, say 1. If they vote for another candidate, their payoff is lower, say 0. Note, however, that in the process of attaching payoffs to strategies, one has to compute the outcome of the votes and then check for the second and the third player whether their vote is in line with the outcome.

Let us now contrast this with the higher-order modelling of games. First note that from the game $\mathcal{G}$ of the previous example, only the "motivation" of players 2 and 3 have changed. Accordingly, we will only need to adjust their selection functions so as to capture their new goal which is to vote for the winner of the contest. Such a goal is exactly captured by equipping J_2 and J_3 with the *fixpoint selection function* fix: $(X \to X) \to \mathcal{P}(X)$, defined in Sect. 2.2. Note that it is neither necessary to change the structure of the game nor to manually compute anything. The new game with the two Keynesian judges is directly described by the tuple

$$\mathcal{G}_K = (\prec\text{-arg max}, \text{fix}, \text{fix}, \text{maj}).$$

One can say that in the higher-order modelling of games we have equipped the individual players themselves with the problem solving ability that we used to compute the payoff matrices such that they represent the motivations of the Keynesian players.

---

[6] The economist John Maynard Keynes [11] remarked that investors in financial markets can be described as not being interested in the outcome per se but that they want to behave in line with the majority (in order to "buy low and sell high"). This behaviour can be elegantly captured as fixed point goals.

3.2   Quantifier Equilibrium

Let us now discuss two different notions of equilibria for higher-order games. Consider a game with n players, and a strategy profile $\mathbf{x} \in \prod_{i=1}^{n} X_i$. Given an outcome function $q\colon \prod_{i=1}^{n} X_i \to R$, the game outcome resulting from this choice of strategy profile is $q(\mathbf{x})$. We can describe the *game context* in which player i unilaterally changes his strategy as

$$\mathcal{U}_i^q(\mathbf{x})(x_i') = q(\mathbf{x}[i \mapsto x_i'])$$

where $\mathbf{x}[i \mapsto x_i']$ is the tuple obtained from $\mathbf{x}$ by replacing the ith entry of the tuple $\mathbf{x}$ with x_i'. Note that indeed $\mathcal{U}_i^q(\mathbf{x})$ has type $X_i \to R$, the appropriate type of a game context for player i.

   We call the n functions $\mathcal{U}_i^q$ $(1 \leq i \leq n)$ the *unilateral maps* of the game. They were introduced in [6] in which it is shown that the proof of Nash's theorem amounts to showing that the unilateral maps have certain topological (continuity and closure) properties. The concept of a context was introduced later in [7], so now we can say that $\mathcal{U}_i^q(\mathbf{x})\colon X_i \to R$ is the game context in which the ith player can unilaterally change his strategy, therefore we call it a unilateral context.

   Using this notation we can abstract the classical definition of Nash equilibrium to our framework.

Definition 8 (Quantifier equilibrium).   *Given a game* $\mathcal{G} = (\varepsilon_1, \ldots, \varepsilon_n, q)$, *we say that a strategy profile* $\mathbf{x} \in \prod_{i=1}^{n} X_i$ *is in* quantifier equilibrium *if*

$$q(\mathbf{x}) \in \overline{\varepsilon_i}(\mathcal{U}_i^q(\mathbf{x}))$$

for all players $1 \leq i \leq n$.

As with the usual notion of Nash equilibrium, we are also saying that a strategy profile is in quantifier equilibrium if no player has a motivation to unilaterally change their strategy. This is expressed formally by saying that preferred outcomes, specified by the selection function when applied to the unilateral context, contain the *outcome* obtained by sticking with the current strategy.

   For illustration, we now compute a quantifier equilibrium for the voting contest game with classical players

$$\mathcal{G} = (\prec'\text{-arg max}, \prec'\text{-arg max}, \prec''\text{-arg max}, \mathrm{maj})$$

as described in Sect. 3.1 in the notation of quantifiers and unilateral contexts. We look at two possible strategy profiles: BBB and BBA. We claim that BBB is a quantifier equilibrium. Note that BBB has outcome $\mathrm{maj}(BBB) = B$. Let us verify this for player 1. The unilateral context of player 1 is

$$\mathcal{U}_1^{\mathrm{maj}}(BBB)(x) = \mathrm{maj}(xBB) = B,$$

meaning that in the given context the outcome is B no matter what player 1 chooses to play. The maximisation quantifier applied to such a unilateral context gives

$$\overline{\varepsilon_1}(\mathcal{U}_1^{\mathrm{maj}}(BBB)) = \succeq_1\text{-max}(\mathcal{U}_1^{\mathrm{maj}}(BBB)) = \{B\},$$

meaning that, in the given context, player 1's preferred outcome is B. Hence, we can conclude by $\mathrm{maj}(BBB) = B \in \{B\} = \overline{\varepsilon_1}(\mathcal{U}_1^{\mathrm{maj}}(BBB))$ that B is a quantifier equilibrium strategy for player 1. This condition holds for each player and allows us to conclude that BBB is a quantifier equilibrium.

On the other hand, we show that BBA is not in quantifier equilibrium. We have that

$$\mathrm{maj}(BBA) = B \notin \{A\} = \overline{\varepsilon_1}(\mathcal{U}_1^{\mathrm{maj}}(BBA)).$$

since $\mathcal{U}_1^{\mathrm{maj}}(BBA)(x) = \mathrm{maj}(xBA) = x$. In other words, the strategy profile BBA gives rise to a game context $\mathcal{U}_1^{\mathrm{maj}}(BBA)$ where player 1 has an incentive to change his strategy to A, so that the new outcome $\mathrm{maj}(ABA) = A$ is better than the previous outcome B.

This game has three quantifier equilibria: $\{AAA, AAB, BBB\}$. They are exactly the same as the Nash equilibria in the normal form representation (cf. Table 1). We will discuss this coincidence in more detail in Sect. 4.2.

3.3   Selection Equilibrium

The definition of quantifier equilibrium is based on quantifiers. However, we can also use selection functions directly to define an equilibrium condition.

Definition 9 (Selection equilibrium). *Given a game* $\mathcal{G} = (\varepsilon_1, \ldots, \varepsilon_n, q)$, *we say that a strategy profile* $\mathbf{x} \in \prod_{i=1}^{n} X_i$ *is in* selection equilibrium *if*

$$x_i \in \varepsilon_i(\mathcal{U}_i^q(\mathbf{x}))$$

for all players $1 \le i \le n$, *where* x_i *is the* i^{th} *component of the tuple* $\mathbf{x}$.

As in the previous subsection, let us illustrate the concept above using the voting contest with classical players from Sect. 3.1. The set of selection equilibria is $\{AAA, AAB, BBB\}$, the same as the set of quantifier equilibria.

Consider BBB and the rationale for player 1. As seen above, his unilateral context is

$$\mathcal{U}_1^{\mathrm{maj}}(BBB)(x) = \mathrm{maj}(xBB) = B.$$

Hence, given this game context his selection function calculates

$$\varepsilon_1(\mathcal{U}_1^{\mathrm{maj}}(BBB)) = \{B\}$$

As before, given that he is not pivotal, an improvement by switching votes is not possible. The same condition holds analogously for the other players.

Let us now investigate the strategy profile BBA. The unilateral context is

$$\mathcal{U}_1^{\mathrm{maj}}(BBA)(x) = \mathrm{maj}(xBA) = x.$$

Given this context, the selection function tells us that player 1 would switch to A:

$$\varepsilon_1(\mathcal{U}_1^{\mathrm{maj}}(BBA)) = \{A\}.$$

Hence, BBA is not a selection equilibrium.

4 Relationship Between Equilibrium Concepts

In this section we show that selection equilibrium is a strict refinement of quantifier equilibrium. Moreover, we show that for a special class of selection functions, which we call *closed selection functions*, the two notions coincide. The obvious question then arises: which concept is more reasonable when games involve non-closed selection functions? We will argue by example that in such cases selection equilibrium is the adequate concept.

4.1 Closed Selection Functions

Selection functions such as *i-arg max(p)*, which one obtains from utility functions as discussed in Sect. 2.2, are examples of what we call *closed selection functions.*

Definition 10 (Closedness). *A selection function* $\varepsilon\colon (X \to R) \to \mathcal{P}(X)$ *is said to be* closed *if whenever* $x \in \varepsilon(p)$ *and* $p(x) = p(x')$ *then* $x' \in \varepsilon(p)$.

Intuitively, a closed selection function is one which chooses optimal moves only based on the outcomes they generate. Two moves that lead to the same outcome are therefore indistinguishable, they are either both good or bad. It is easy to see that the selection function arg max is closed. Agents modelled via closed selection functions do not put any preferences on moves that lead to identical outcomes.

An example of a non-closed selection function is the fixpoint operator

$$\mathrm{fix}\colon (X \to X) \to \mathcal{P}(X).$$

defined in Sect. 2.2. To see that fix is non-closed, we might have two points $x \neq x'$ which both map to x (i.e. $p(x) = p(x') = x$) so that x is a fixed point but x' is not.

One can consider translating quantifiers into selection functions and back into quantifiers, or conversely.

Proposition 1. *For all* $p\colon X \to R$ *we have*

(i) $\overline{\overline{\varphi}}(p) = \varphi(p)$ *if* φ *is an attainable quantifier of type* $(X \to R) \to \mathcal{P}(R)$
(ii) $\varepsilon(p) \subseteq \overline{\overline{\varepsilon}}(p)$ *for any selection functions of type* $(X \to R) \to \mathcal{P}(X)$.

Proof. These are easy to derive. Let us briefly outline $\varepsilon(p) \subseteq \overline{\overline{\varepsilon}}(p)$. Suppose $x \in \varepsilon(p)$ is a good move in the game context $p\colon X \to R$. By Definition 5 we have that $p(x) \in \overline{\varepsilon}(p)$. Finally, by Definition 6 we have that $x \in \overline{\overline{\varepsilon}}(p)$.   □

The proposition above shows that on attainable quantifiers the double-overline operation calculates the same quantifier we started with. On general selection functions, however, the mapping $\varepsilon \mapsto \overline{\overline{\varepsilon}}$ can be viewed as a *closure* operator.[7] Intuitively, the new selection function $\overline{\overline{\varepsilon}}$ will have the same good *outcomes* as the original one, but it might consider many more *moves* to be good as well, as it does not distinguish moves which both lead to equally good outcomes.

---

[7] Note that we might have a strict inclusion $\varepsilon(p) \subset \overline{\overline{\varepsilon}}(p)$ in case we have $x_1 \neq x_2$, with $x_1 \in \varepsilon(p)$ and $x_2 \notin \varepsilon(p)$ but $p(x_1) = p(x_2)$.

Proposition 2. *A selection function ε is* closed *if and only if $\varepsilon = \overline{\overline{\varepsilon}}$.*

Proof. Assume first that ε is closed, i.e.

(i) $x \in \varepsilon(p)$ and $p(x) = p(x')$ then $x' \in \varepsilon(p)$.

By Proposition 1 is it enough to show that if $x' \in \overline{\overline{\varepsilon}}(p)$ then $x' \in \varepsilon(p)$. Assuming $x' \in \overline{\overline{\varepsilon}}(p)$, and by Definition 6 we have

(ii) $p(x') \in \overline{\varepsilon}(p)$.

By Definition 5, (ii) says that $p(x') = p(x)$ for some $x \in \varepsilon(p)$. By (i) it follows that $x \in \varepsilon(p)$.
Conversely, assume that $\varepsilon = \overline{\overline{\varepsilon}}$ and that $x \in \varepsilon(p)$ and $p(x) = p(x')$. We wish to show that $x' \in \varepsilon(p)$. Since $x \in \varepsilon(p)$ then $p(x) \in \overline{\varepsilon}(p)$. But since $p(x) = p(x')$ we have that $p(x') \in \overline{\varepsilon}(p)$. Hence, $x' \in \overline{\overline{\varepsilon}}(p)$. But since $\varepsilon = \overline{\overline{\varepsilon}}$ it follows that $x' \in \varepsilon(p)$. $\square$

4.2 Selection Refines Quantifier Equilibrium

The following theorem shows that selection equilibrium is a refinement of quantifier equilibrium.

Theorem 1. *Every selection equilibrium is a quantifier equilibrium.*

Proof. Recall that by definition, for every context p we have $x \in \varepsilon_i(p) \implies p(x) \in \overline{\varepsilon_i}(p)$, since $\overline{\varepsilon_i}(p) = \{p(x) \mid x \in \varepsilon_i(p)\}$. Assuming that $\mathbf{x}$ is a selection equilibrium we have $x_i \in \varepsilon_i(\mathcal{U}_i^q(\mathbf{x}))$ Therefore $\mathcal{U}_i^q(\mathbf{x})(x_i) \in \overline{\varepsilon_i}(\mathcal{U}_i^q(\mathbf{x}))$. It remains to note that $\mathcal{U}_i^q(\mathbf{x})(x_i) = q(\mathbf{x})$, because $\mathbf{x}[i \mapsto x_i] = \mathbf{x}$. $\square$

However, for closed selection functions the two notions coincide:

Theorem 2. *If $\varepsilon_i = \overline{\overline{\varepsilon_i}}$, for $1 \leq i \leq n$, then the two equilibrium concepts coincide.*

Proof. Given the previous theorem, it remains to show that under the assumption $\varepsilon_i = \overline{\overline{\varepsilon_i}}$ any strategy profile $\mathbf{x}$ in quantifier equilibrium is also in selection equilibrium. Fix i and suppose $\mathbf{x}$ is such that $q(\mathbf{x}) \in \overline{\varepsilon_i}(\mathcal{U}_i^q(\mathbf{x}))$. Since $\mathcal{U}_i^q(\mathbf{x})(x_i) = q(\mathbf{x})$, we have $\mathcal{U}_i^q(\mathbf{x})(x_i) \in \overline{\varepsilon_i}(\mathcal{U}_i^q(\mathbf{x}))$. By the definition of $\overline{\overline{\varepsilon_i}}$ it follows that $x_i \in \overline{\overline{\varepsilon_i}}(\mathcal{U}_i^q(\mathbf{x}))$. Therefore, since $\varepsilon_i = \overline{\overline{\varepsilon_i}}$, we obtain $x_i \in \varepsilon_i(\mathcal{U}_i^q(\mathbf{x}))$. $\square$

The theorem above explains why in the voting contest with classical preferences the strategy profiles that were quantifier equilibrium were the same as those in selection equilibrium. This example can be modelled with closed selection functions. Moreover, since arg max can be easily shown to be closed, in the classical modelling of games via maximising players, our two notions of equilibrium also coincide. The following theorem shows that they both indeed also coincide with the standard notion of Nash equilibrium.

Theorem 3. *In a classical game (see Remark 1) the standard definition of Nash equilibrium and the equilibrium notions of Definitions 8 and 9 are equivalent.*

Proof. Suppose the set of outcomes R is $\mathbb{R}^n$ and that the selection functions ε_i are i-arg max, i.e. maximising with respect to i^{th} coordinate. Unfolding Definition 9 and that of a unilateral context $\mathcal{U}_i^q(\mathbf{x})$, we see that a tuple $\mathbf{x}$ is an equilibrium strategy profile if for all $1 \leq i \leq n$

$$x_i \in i\text{-arg}\max_{x \in X_i} q(\mathbf{x}[i \mapsto x]).$$

But x_i is a point on which the function $p(x) = q(\mathbf{x}[i \mapsto x])$ attains its maximum precisely when $p(x_i) \in \max_{x \in X_i} p(x)$. Hence

$$q(\mathbf{x}) = q(\mathbf{x}[i \mapsto x_i]) = p(x_i) = \max_{x \in X_i} p(x) = \max_{x \in X_i} q(\mathbf{x}[i \mapsto x])$$

which is the standard definition of a Nash equilibrium: for each player i, the outcome obtained by not changing the strategy, i.e. $q(\mathbf{x})$, is the best possible amongst the outcomes when any other available strategy is considered, i.e. $\max_{x \in X_i} q(\mathbf{x}[i \mapsto x])$. □

Theorem 3 above shows that in the case of classical games the usual concept of a Nash equilibrium coincides with both the quantifier equilibrium and the selection equilibrium. On the other hand, for general games, Theorem 1 proves that every selection equilibrium is a quantifier equilibrium.

$$\text{selection equilibria} \subsetneq \text{quantifier equilibria}$$

In the following section we give examples showing that the inclusion above is strict, i.e. that there are games where selection equilibrium is a strict refinement of quantifier equilibrium. By Theorem 2 these examples necessarily make use of players modelled by non-closed selection functions.

4.3   Illustrating the Two Solution Concepts

In Sect. 3.1 we have discussed the representation of the *voting contest with Keynesian players* game both in normal form as well as in higher-order functions. Here, we will turn to analysing the equilibria of its higher-order representation

$$\mathcal{G}_{\mathrm{K}} = (\prec\text{-arg}\max, \mathrm{fix}, \mathrm{fix}, \mathrm{maj}).$$

We begin with quantifier equilibria (see Table 3). These include the strategy profiles where judges (players) J_2 and J_3 are both coordinated but also profiles where either J_2 or J_3 is in the minority. Readers are encouraged to download the Haskell implementation, and interactively verify entries of this table.

We illustrate the rationale for the strategy profile AAB of the Keynesian player 3. The outcome of AAB is $\mathrm{maj}(AAB) = A$. The unilateral context of player 3 is

$$\mathcal{U}_3^{\mathrm{maj}}(AAB)(x) = \mathrm{maj}(AAx) = A$$

Table 3. Players: max, fix, fix

Strategy	Outcome	Quantifier eq.	Defects	Selection eq.	Defects
AAA	A	✓		✓	
AAB	A	✓		-	J_3
ABA	A	✓		-	J_2
ABB	B	✓		✓	
BAA	A	✓		✓	
BAB	B	-	J_1	-	$J_1,\ J_2$
BBA	B	-	J_1	-	$J_1,\ J_3$
BBB	B	✓		✓	

meaning that the outcome is (still) A if player 3 unilaterally changes from B to A. The fixed point quantifier applied to this context gives

$$\overline{\varepsilon_3}(\mathcal{U}_3^{\mathrm{maj}}(AAB)) = \mathrm{fix}(\mathcal{U}_3^{\mathrm{maj}}(AAB)) = \{A\}$$

meaning that A is the outcome resulting from an optimal choice. Hence, we can conclude by

$$\mathrm{maj}(AAB) = A \in \{A\} = \overline{\varepsilon_3}(\mathcal{U}_3^{\mathrm{maj}}(AAB))$$

that player 3 is happy with his choice of move B according to the quantifier equilibrium notion. This already demonstrates the problem with the *quantifier equilibrium* notion, since the Keynesian player 3 has voted for B but A is the winner, so he should not be happy at all!

Now, let us turn to the selection equilibria. Table 3 also contains the selection equilibria and it shows that they are a strict subset of the quantifier equilibria. Consider again the strategy profile AAB, focusing on the third player. In the case of the selection equilibrium we have

$$B \notin \{A\} = \mathrm{fix}(\mathcal{U}_3^{\mathrm{maj}}(AAB)) = \varepsilon_3(\mathcal{U}_3^{\mathrm{maj}}(AAB))$$

meaning that player 3 is not happy with his current choice of strategy B with respect to the strategy profile AAB.

Remark 3. Given Theorem 2 it follows immediately that $\mathrm{fix} \colon (X \to X) \to \mathcal{P}(X)$ is not a closed selection function. Indeed, it is easy to calculate that

$$\overline{\overline{\mathrm{fix}}}(p) = \{x \mid p(x) = p(y),\ \text{for some } y \text{ such that } y = p(y)\},$$

i.e. $\overline{\overline{\mathrm{fix}}}(p)$ is the inverse image of $\mathrm{fix}(p)$, so it contains not only all fixed points of p but also points that map through p to a fixed point.

The selection equilibria are precisely those in which J_2 and J_3 are coordinated, and J_1 is not pivotal in any of these. For illustration, consider the strategy

AAA, which is a selection equilibrium of this game. Suppose the moves of J_1 and J_2 are fixed, but J_3 may unilaterally change strategy. The unilateral context is

$$\mathcal{U}_3^{\mathrm{maj}}(AAA)(x) = \mathrm{maj}(AAx) = A$$

Thus the unilateral context is a constant function, and its set of fixpoints is

$$\mathrm{fix}(\mathcal{U}_3^{\mathrm{maj}}(AAA)) = \{A\}.$$

This tells us that J_3 has no incentive to unilaterally change to the strategy B, because he will no longer be voting for the winner.

On the other hand, for the strategy ABB the two Keynesian players are indifferent, because if either of them unilaterally changes to A then A will become the majority and they will still be voting for the winner. This is still a selection equilibrium (as we would expect) because the unilateral context is the identity function, and in particular B is a fixpoint.

As a last point, let us compare the selection and quantifier equilibria of Table 3 with the Nash equilibria in the normal form game. The payoff matrix in Table 2 also depicts Nash equilibria payoffs as marked in bold. Note that the latter are the same as the selection equilibria. Thus, in general selection equilibrium appears to be the adequate solution concept.

Coordination. As a last point, consider a game where all players want to vote for the winner of the contest. Table 4 represents the payoffs of this game; Nash equilibria are in bold. Clearly the only two equilibria are when all judges vote unanimously for a given contestant. Judges J_1, J_2 and J_3 want to vote for the winner, so the selection functions are all given by the fixpoint operator.

Table 4. Nash equilibria of coordination game

	J_2: A	J_2: B	
J_1: A	**1,1,1**	1,0,1	J_3: A
J_1: B	0,1,1	1,1,0	

	J_2: A	J_2: B	
J_1: A	1,1,0	0,1,1	J_3: B
J_1: B	1,0,1	**1,1,1**	

The selection equilibria of the higher-order representation of this game are exactly the coordinated strategies. This game is a good example of why quantifier equilibria are not suitable for modelling games with non-closed selection functions: every strategy is a quantifier equilibrium of this game, but the selection equilibrium captures the intuition perfectly that the equilibria should be the strategy profiles that are maximally coordinated, namely AAA and BBB.

5   Conclusion

In this paper, we introduced a representation of strategic games based on *quantifiers* and *selection functions* as well as a new equilibrium concept, and showed by

example that the selection equilibrium is the appropriate concept as it works well even when players are described by non-closed selection functions. We focused on simultaneous move games. Yet, the theory as well as the implementation naturally extend to sequential games. Moreover, multi-valued selection functions as formulated here have sparked new research avenues, for instance on so called "open games" [5, 8, 10], a compositional approach to game theory.

References

1. Botta, N., Ionescu, C., Brady, E.: Sequential decision problems, dependently typed solutions. In: Proceedings of PLMMS 2013 (2013)
2. Escardó, M., Oliva, P.: Selection functions, bar recursion and backward induction. Math. Struct. Comput. Sci. **20**(2), 127–168 (2010)
3. Escardó, M., Oliva, P.: What sequential games, the Tychonoff theorem and the double-negation shift have in common. In: Proceedings of the Third ACM SIGPLAN Workshop on Mathematically Structured Functional Programming (MSFP 2010), pp. 21–32 (2010)
4. Escardó, M., Oliva, P.: Sequential games and optimal strategies. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. **467**(2130), 1519–1545 (2011)
5. Ghani, N., Hedges, J.: A compositional approach to economic game theory. arXiv:1603.04641 (2016)
6. Hedges, J.: A generalization of Nash's theorem with higher-order functionals. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. **469**(2154) (2013). http://rspa.royalsocietypublishing.org/content/469/2154/20130041
7. Hedges, J.: Monad transformers for backtracking search. In: Proceedings of the 5th Workshop on Mathematically Structured Functional Programming, pp. 31–50. Open Publishing Association (2014)
8. Hedges, J.: Towards compositional game theory. Ph.D. thesis, Queen Mary University of London (2016)
9. Hedges, J., Oliva, P., Sprits, E., Winschel, V., Zahn, P.: Higher-order decision theory. arXiv preprint cs.GT, arXiv:1506.01003 (2015)
10. Hedges, J., Sprits, E., Winschel, V., Zahn, P.: Compositionality and string diagrams for game theory. arXiv:1604.06061 (2015)
11. Keynes, J.M.: General Theory of Employment, Interest and Money. Macmillan, London (1936)
12. Mostowski, A.: On a generalization of quantifiers. Fundamenta Mathematicae **44**, 12–36 (1957)
13. Osborne, M., Rubinstein, A.: Course in Game Theory. MIT Press, Cambridge (1994)