An Existence Theorem of Nash Equilibrium in Coq and Isabelle*

Stéphane Le Roux
Université Libre de Bruxelles
stephane.le.roux@ulb.ac.be

Érik Martin-Dorel
IRIT, Université de Toulouse
erik.martin-dorel@irit.fr

Jan-Georg Smaus
IRIT, Université de Toulouse
jan-georg.smaus@irit.fr

Nash equilibrium (NE) is a central concept in game theory. Here we prove formally a published theorem on existence of an NE in two proof assistants, Coq and Isabelle: starting from a game with finitely many outcomes, one may derive a game by rewriting each of these outcomes with either of two basic outcomes, namely that Player 1 wins or that Player 2 wins. If all ways of deriving such a *win/lose game* lead to a game where one player has a winning strategy, the original game also has a Nash equilibrium.

This article makes three other contributions: first, while the original proof invoked linear extension of strict partial orders, here we avoid it by generalizing the relevant definition. Second, we notice that the theorem also implies the existence of a secure equilibrium, a stronger version of NE that was introduced for model checking. Third, we also notice that the constructive proof of the theorem computes secure equilibria for non-zero-sum priority games (generalizing parity games) in quasi-polynomial time.

1 Introduction and motivations

The four-color theorem was the first major theorem proved with the assistance of a computer in 1976. In [2] the computer was merely checking thousands of cases via a dedicated program, thus completing an otherwise paper-and-pencil proof. In [9] the computer checked the whole formalized proof via a general-purpose and widely used software named Coq. Since then, other challenging (or just interesting) theorems have been likewise formalized and checked.

In 2006 Coq was used by [19], which was generalized by [13], to formalize and check a result from game theory: Kuhn's existence of a Nash equilibrium (NE) in finite games in extensive form. Coq was also used to deal with some infinite games in extensive form [15], or with random Boolean games [17]. Isabelle/HOL, another proof software, was used recently [6] to formalize and check a result in game theory for logic and computer science: the positional determinacy of parity games.

This article formalizes a game-theoretic result, essentially [14, Lemma 2.4], both in Coq and Isabelle. The result is as follows: starting from a game with finitely many outcomes, one may derive a game by rewriting each of these outcomes with either of two basic outcomes, namely that Player 1 wins or that Player 2 wins. If all ways of deriving such a *win/lose game* lead to a game where one player has a winning strategy, the original game also has an NE. We chose to prove this result for several reasons:

- It lies at the boundary of traditional game theory and game theory for logic and computer science. Indeed, it extends determinacy, typically a logic and computer science concern, into existence of

*This work was partly supported by the FAGames project of LabEx CIMI. S. Le Roux was also partly supported by the ERC inVEST (279499) project.

P. Bouyer, A. Orlandini & P. San Pietro (Eds.): 8th Symposium on Games, Automata, Logics and Formal Verification (GandALF'17)
EPTCS 256, 2017, pp. 46–60, doi:10.4204/EPTCS.256.4

NE, typically a game-theoretic concern. As examples, [14, Lemma 2.4] generalizes Borel determinacy [16], finite-memory determinacy of Muller games [10], and positional determinacy of parity games [7]. In this article, we further notice that the theorem also implies the existence of a secure equilibrium, a stronger version of NE that was introduced in [5] for model checking.

- The proof is constructive and the corresponding algorithm (building an NE provided that we can solve determinacy) has linear time complexity in the number of outcomes. Since a recent breakthrough [4] shows that parity games can be solved in quasi-polynomial time, we note in this article that secure equilibria for non-zero-sum priority games (generalizing parity games) can also be computed in quasi-polytime.
- The result features games in normal form, a very general class of games that includes finite and infinite games in extensive form discussed in [19], [13], [15] and [16], and Muller and parity games discussed in [10], [7], [6], and [4].
- The result's statement and proof involve and combine several basic concepts in game theory, so the ability to handle them properly could constitute a basis for a usable game-theory library.
- The result is a slight weakening of [14, Lemma 2.4], i.e. the interesting part of the lemma. The full lemma is only technically needed in [14], since it is the base case of the big proof of [14, Theorem 2.7]. The big proof goes by induction on the order types of the inverses of the players' preferences, where the order types are assumed to be countable ordinals. So, this article essentially proves the base case of the induction and leaves the inductive case for future work.
- Variants of the base case were proved independently in [12] and [11]. The fact that the idea behind the theorem emerged in different communities suggests that it is broadly interesting.

A significant contribution of this article is the modification of the proof structure of [14, Lemma 2.4] to simplify its formalization: In [14, Lemma 2.4], the preferences are extended linearly in the beginning of the proof. Then the new linear preferences are lifted to subsets of outcomes, where the definition of the lift hinges upon the linearity assumption. This helps find an NE for the new preferences, which is also an NE for the original ones. While it is convenient to invoke linear extension in the paper-and-pencil proof, it is costly to formalize. It was already formalized in Coq in [13] (and improved in [1] in terms of algorithmic complexity), but we prefer to avoid relying too much on external libraries. So we generalize the lift such that the input may be an arbitrary partial order instead of necessarily a linear order.

Organization of the paper: Section 2 gives background definitions in game theory; Section 3 generalizes the lift of the preference; Section 4 gives a new paper-and-pencil proof of [14, Lemma 2.4] without invoking linear extension; Section 5 discusses secure equilibria and their computation; Sections 6 and 7 describe the formal proofs in Coq and Isabelle/HOL, respectively. Section 8 gives concluding remarks. The formal developments are available at `https://www.irit.fr/~Erik.Martin-Dorel/equi-thm/`.

2 Background definitions

Game forms (introduced in [8]) are the central concept of our article. They can be instantiated into games by providing preferences for the players. Then, the Nash equilibria are defined for games. The win/lose games and their winning strategies are an important special case. We recall all this below.

Definition 1 A *game form* is a tuple $\langle A, (S_a)_{a\in A}, O, v\rangle$ such that

- A is a nonempty set (of players, or agents),
- $\prod_{a\in A} S_a$ is a nonempty Cartesian product (whose elements are the strategy profiles and where S_a represents the strategies available to player a),
- O is a nonempty set (of possible outcomes),
- $v : \prod_{a\in A} S_a \to O$ is the outcome function that values the strategy profiles.

A game form endowed with a binary relation $\prec_a$ over O for each player a (modeling her preference) is called a *game in normal form*. In the remainder we just write "game" for short.

Definition 2 (Nash equilibrium) Let $\langle A, (S_a)_{a\in A}, O, v, (\prec_a)_{a\in A}\rangle$ be a game. A strategy profile s in $S := \prod_{a\in A} S_a$ is a *Nash equilibrium* if it makes every player a stable, i.e., $v(s) \not\prec_a v(s')$ for all $s' \in S$ that differ from s at most in the a-component:

$$NE(s) \quad := \quad \forall a \in A, \forall s' \in S, \quad \big(\forall b \in A \setminus \{a\}, s_b = s'_b\big) \;\Rightarrow\; v(s) \not\prec_a v(s')$$

Four games are shown in Figure 1, with Players 1 and 2 who have two strategies each. Here, the outcomes are in $\mathbb{R}^2$ and are called *payoff pairs*. Player 1 (2) prefers payoff pairs with greater first (second) component. In the first game, if Player 1 picks the strategy 1_t and Player 2 picks 2_l, the strategy profile $(1_t, 2_l)$ then yields payoff 1 for Player 1 and 0 for Player 2. The payoff pairs that correspond to NEs are written in bold. E.g., the second game has no NE. Note that the usual definition of NE uses $\geq$ to compare real numbers, but in our general setting, using $\not\prec_a$ instead expresses exactly the intended concept of NE.

	2_l	2_r
1_t	$1, 0$	$\mathbf{5, 0}$
1_b	$\mathbf{2, 4}$	$5, 3$

	2_l	2_r
1_t	$0, 1$	$1, 0$
1_b	$1, 0$	$0, 1$

	2_l	2_r
1_t	$\mathbf{2, 1}$	$0, 0$
1_b	$0, 0$	$\mathbf{1, 2}$

	2_l	2_r
1_t	$0, 1$	$0, 1$
1_b	$\mathbf{1, 0}$	$\mathbf{1, 0}$

Figure 1: Four two-player games with two strategies each

Definition 3
- A *win/lose game* is a game where $A = \{1,2\}$ and $O = \{(1,0),(0,1)\}$ and the preferences are defined by $(0,1) \prec_1 (1,0)$ and $(1,0) \prec_2 (0,1)$.
- A *winning strategy* for Player 1 is a strategy $s_1 \in S_1$ such that $v(s_1, s_2) = (1,0)$ for all $s_2 \in S_2$. A winning strategy for Player 2 is a strategy $s_2 \in S_2$ such that $v(s_1, s_2) = (0,1)$ for all $s_1 \in S_1$.
- A win/lose game such that one player has a winning strategy is said to be *determined*. For $i \in \{1,2\}$, if Player i is the winning player and the winning strategy is in some $R_i \subseteq S_i$, the game is said to be *determined via* R_i.

The second game of Figure 1 is a non-determined win/lose game. The fourth game is also win/lose, and 1_b is a winning strategy for Player 1, so the game is determined.

The notion of winning strategy is relevant for win/lose games only, but the following remark clarifies why the transfer from winning strategy to multi-outcome Nash equilibrium is a process of generalization.

Remark 4 A win/lose game has a winning strategy iff it has a Nash equilibrium. ◁

Remark 4 is formalized as the Coq lemma `determined_iff_NE` in Section 6.3, and its right-to-left implication is also formalized as the Isabelle theorem *`someone_wins`* in Section 7.3.

From a two-player game form one may derive both games and win/lose games:

Definition 5 Let $G = \langle \{1,2\}, S_1, S_2, O, v \rangle$ be a two-player game form.

1. For all $\prec_1, \prec_2 \subseteq O^2$ the game $\langle \{1,2\}, S_1, S_2, O, v, \{\prec_1, \prec_2\} \rangle$ is said to be *derived* from G.
2. Let $wl : O \to \{(1,0),(0,1)\}$. The win/lose game $\langle S_1, S_2, wl \circ v \rangle$ is also said to be *derived* from G.
3. Let $R_1 \subseteq S_1$ and $R_2 \subseteq S_2$. If all win/lose games derived from a game form are determined (via R_1 or R_2, resp., depending on who wins), the game form is also said to be *determined* (*via* R_1 *and* R_2).
4. Let $P \subseteq O$, and let $s_1 \in S_1$ be such that $v(s_1, S_2) := \{v(s_1, s_2) \mid s_2 \in S_2\} \subseteq P$. The strategy s_1 is said to *enforce* P. (And likewise $s_2 \in S_2$ may enforce subsets of outcomes.)

The concept of *determined game form* is used to state our theorem. The leftmost game form in Figure 2 is not determined, e.g., because instantiating X with $(0,1)$ and Y with $(1,0)$ yields a non-determined game, namely the second game in Figure 1. The second game form in Figure 2 is not determined either. The third game form in Figure 2 is determined: if Y is mapped to $(1,0)$ then 1_b is winning for Player 1; if X and Z are mapped to $(1,0)$ then 1_t is winning for Player 1; else either 2_l or 2_r is winning for Player 2. The last game form in Figure 2 is also determined: if Y is mapped to $(1,0)$ then 1_b is winning for Player 1; else 2_r is winning for Player 2. Note that the winner of a game obtained by instantiating a determined game form may depend on the instance.

	2_l	2_r
1_t	X	Y
1_b	Y	X

	2_l	2_m	2_r
1_t	X	Y	Z
1_b	Y	Z	X

	2_l	2_r
1_t	X	Z
1_b	Y	Y

	2_l	2_m	2_r
1_t	X	Z	Y
1_b	Y	Y	Y

Figure 2: Four two-player game forms

Note that since S_1 and S_2 are nonempty by definition, no player can enforce the empty set.

Also note that the subsets R_i from Definition 5 represent strategies of special interest. For instance, Muller games are determined *via* strategies that can be described by finite automata [10], and parity games are determined *via* strategies that are called *positional* [7].

Finally note that, given a two-player game form G, Player a can enforce an outcome subset P in G iff $P \in E_G(\{a\})$ where E_G is the *effectivity function* of G. These functions were introduced in [18] and are now widely used in cooperative game theory and social choice. Here we use only a special case of them.

Lemma 6 connects Definitions 5.3 and 5.4. It is key in the original and in the formalized proofs.

Lemma 6 A game form is determined (via R_1 and R_2) iff for each subset of the outcomes, either the subset can be enforced by Player 1 (via R_1), or its complement can be enforced by Player 2 (via R_2).

Note that in the lemma one can interchange 1 and 2 and obtain an equivalent statement, but one cannot replace "by Player ..." with the phrase "by one of the players"! See the second game form in Figure 2: for every subset $P \subseteq \{X,Y,Z\}$, either P or $\{X,Y,Z\} \setminus P$ can be enforced by one of the players, but $\{X,Y\}$, e.g., cannot be enforced by Player 1 and its complement cannot be enforced by Player 2.

3 Lifting the preference without prior linear extension

We now present the extension of the lift of the preference, which we mentioned in the end of Section 1. In [14] the lift required that the preference be a linear order; below we extend the definition to arbitrary partial orders. In the whole section, $\prec$ is a strict partial order over a set O, i.e., it is a binary relation that is transitive and irreflexive. (Irreflexivity means that it satisfies $\forall x \in O, \neg(x \prec x)$.)

Definition 7
- For all $x \in O$, let $u(x) := \{z \in O \mid x \prec z\}$, the *strict upper set* of x.
- For all $Y \subseteq O$, let $u(Y) := \cup_{y \in Y} u(y)$, the *strict upper set* of Y.
- For $A, B \subseteq O$, let $A \prec^{\mathscr{P}} B$ (*lift* of $\prec$) iff $\exists A' \subseteq A \setminus B,\ A' \neq \emptyset \wedge A \setminus (A' \cup u(A')) = B \setminus (A' \cup u(A'))\big)$.

Let us give some intuition behind Definition 7. First, $\prec^{\mathscr{P}}$ is irreflexive since there is no nonempty A' in $A \setminus A$. Second, if $B \subsetneq A \subseteq O$, picking $A' := A \setminus B$ shows that $A \prec^{\mathscr{P}} B$. In particular, $\emptyset$ is the only $\prec^{\mathscr{P}}$-maximal set and O the only $\prec^{\mathscr{P}}$-minimal set. More generally, note that for all sets A, B, C we have $A \setminus C = B \setminus C$ iff $A \Delta B \subseteq C$, where $A \Delta B := (A \setminus B) \cup (B \setminus A)$ is the symmetric difference of A and B. Therefore $A \prec^{\mathscr{P}} B$ iff there is a nonempty $A' \subseteq A \setminus B$ such that $A \Delta B \subseteq A' \cup u(A')$. (This emphasizes that whether $A \prec^{\mathscr{P}} B$ only depends on how $\prec$ behaves on $A \Delta B$, the points where A and B disagree.)

Let us further assume that O is finite in the remainder of the article, so $A \prec^{\mathscr{P}} B$ iff the minimal elements of $A \Delta B$ are all in $A \setminus B$. It is now easy to see that for all $A, B \subseteq O$, if $A \prec^{\mathscr{P}} B$, the A' witnessing it are exactly the subsets of $A \setminus B$ containing all the minima of $A \setminus B$ (and thus of $A \Delta B$).

Note that although the lift of a preference coincides with the preference on singleton outcomes, it is difficult to interpret the full lift game-theoretically. This remark holds even for linear preferences, as shown by considering the usual order $<$ on the natural numbers. Then $\{1,2,3,4,5\} <^{\mathscr{P}} \{1\} <^{\mathscr{P}} \{2,3,4,5\} <^{\mathscr{P}} \{2,3\} <^{\mathscr{P}} \{2,4,5\} <^{\mathscr{P}} \{2\} <^{\mathscr{P}} \{3,5\} <^{\mathscr{P}} \{3\} <^{\mathscr{P}} \{4\} <^{\mathscr{P}} \emptyset$.

If O is finite, the characterization of $\prec^{\mathscr{P}}$ using the minima of the symmetric difference is more intuitive than Definition 7 itself, yet it is unclear whether it is easier to handle with a proof assistant.

Lemma 8 proves that $\prec^{\mathscr{P}}$ is a strict partial order, just like $\prec$. Note that dropping the non-emptiness condition (of A') in Definition 7 would yield a partial order, i.e., the reflexive closure of the actual $\prec^{\mathscr{P}}$.

Lemma 8 If O is finite, $\prec^{\mathscr{P}}$ is a strict partial order.

PROOF It is irreflexive as discussed above; the main difficulty is to prove transitivity. Let $A \prec^{\mathscr{P}} B$ be witnessed by $A' \neq \emptyset$ and $B \prec^{\mathscr{P}} C$ be witnessed by $B' \neq \emptyset$. Since $A' \subseteq O$, $B' \subseteq O$, and O is finite, $A' \cup B'$ is finite too. Let A'' be the minimal elements of $A' \cup B'$. We argue below that $A \prec^{\mathscr{P}} C$ is witnessed by A''.

Let $x \in A''$, and let us first prove that $x \notin C$. First case, $x \in B'$, so $x \notin C$ by definition. Second case, $x \in A'$, so $x \in A$ but $x \notin B$ and in particular $x \notin B'$. For all $y \prec x$ we have $y \notin B'$ by definition of A'', so $x \notin u(B')$, so $x \notin B' \cup u(B')$. By definition $B \setminus (B' \cup u(B')) = C \setminus (B' \cup u(B'))$, so $x \notin C$ since $x \notin B$. Let us now prove that $x \in A$. First case, $x \in A'$, so $x \in A$ by definition. Second case, $x \in B'$, so $x \in B$, and $x \notin A'$. Moreover $y \notin A'$ for all $y \prec x$ by definition of A'', so $x \notin A' \cup u(A')$. By definition $A \setminus (A' \cup u(A')) = B \setminus (A' \cup u(A'))$, so $x \in A$ since $x \in B$.

Let us finally prove that $A \setminus (A'' \cup u(A'')) = C \setminus (A'' \cup u(A''))$. Let $x \notin A'' \cup u(A'')$, so $x \notin A' \cup u(A')$ and $x \notin B' \cup u(B')$ since $A' \cup u(A') \cup B' \cup u(B') \subseteq A'' \cup u(A'')$. (Equality holds but is not needed here.) Therefore $x \in A$ iff $x \in B$ (since $x \notin A' \cup u(A')$) and $x \in B$ iff $x \in C$ (since $x \notin B' \cup u(B')$). □

Note that if $\prec$ and set membership are decidable in polynomial time in the cardinality of O, so is deciding $A \prec^{\mathscr{P}} B$: decide whether $A \neq B$ and whether for all elements in $B \setminus A$ there is a smaller element in $A \setminus B$.

Beside being convenient for our proofs, Definition 7 might be useful outside of game theory. It provides a canonical way to lift a finite order of elements to a finite order of subsets. We can already note that it can be generalized to finite multisets: for all multisets $f, g : O \to \mathbb{N}$, let us define $f \prec^{\mathscr{P}} g$ iff $\{x \in O \mid g(x) < f(x)\} \prec^{\mathscr{P}} \{x \in O \mid f(x) < g(x)\}$.

4 Paper-and-pencil proof of [14, Lemma 2.4] using Lemma 8

Let us now state the theorem and provide the alternative paper-and-pencil proof that is easier to formalize. We recall that the original proof invokes linear extension of the preferences, whereas our new proof uses

only Lemma 8.

Theorem 9 (Finitary equilibrium transfer) Let $\langle\{1,2\},S_1,S_2,O,v,\{\prec_1,\prec_2\}\rangle$ be a two-player game with finite O, let $R_1 \subseteq S_1$ and $R_2 \subseteq S_2$ be subsets of the strategy sets, and let us assume the following:

1. the underlying game form is determined via R_1 and R_2;
2. both preferences $\prec_1$ and $\prec_2$ are strict partial orders.

Then the game $\langle\{1,2\},S_1,S_2,O,v,\{\prec_1,\prec_2\}\rangle$ has a Nash equilibrium in $R_1 \times R_2$.

PROOF We number the paragraphs of the proof to facilitate comparison with the formal proofs.

1. Let $\prec_1^{\mathscr{P}}$ be the lift of $\prec_1$ along Definition 7. It is a strict partial order by Lemma 8. Let M be a $\prec_1^{\mathscr{P}}$-maximal subset of O that Player 1 can enforce via R_1 and let $s_1 \in R_1$ be a strategy enforcing M.
2. M is finite, as a subset of O, and nonempty, since no player can enforce the empty set. Since $\prec_2$ is a strict partial order, let m be $\prec_2$-maximal in M, and let $M' := (M \setminus \{m\}) \cup u(m)$ (where $u(m)$ denotes $\{z \in M' \mid m \prec_1 z\}$, the strict upper set with respect to Player 1's preference). Note that for all sets A,B,C such that $B \subseteq C$ we have $(A \setminus B) \setminus C = (A \cup B) \setminus C = A \setminus C$. So $M \setminus (\{m\} \cup u(m)) = M' \setminus (\{m\} \cup u(m))$. Since $m \in M \setminus M'$, it witnesses that $M \prec_1^{\mathscr{P}} M'$.
3. By maximality in the definition of M, Player 1 cannot enforce M'. So Player 2 can enforce $O \setminus M'$ by determinacy assumption and Lemma 6. Let $s_2 \in R_2$ be a strategy enforcing $O \setminus M'$, so that $v(s_1,s_2) \in M \cap (O \setminus M') = \{m\}$.
4. First, the strategy profile (s_1,s_2) makes Player 2 stable, since m is $\prec_2$-maximal among M, which is enforced by s_1. Second, let $o \in O$ be such that $m \prec_1 o$, i.e. $o \in u(m)$, so $o \in M'$. This shows that m is $\prec_1$-maximal among $O \setminus M'$, which is enforced by s_2. So the profile (s_1,s_2) also makes Player 1 stable. Therefore $(s_1,s_2) \in R_1 \times R_2$ is a Nash equilibrium. □

Note that although the roles of the players are symmetric in the original question of existence of Nash equilibria, the proof of Theorem 9 breaks this symmetry: One player (Player 1 in the proof) first selects a strategy that somehow maximizes her guarantee in a "lexicographic-like" way, and then her opponent (Player 2 in the proof) maximizes his outcome among the available ones, while locking Player 1 into her strategy. (The symmetry need not be broken if the preferences are antagonistic, though, i.e. if one is the symmetric of the other: both players may independently pick strategies as Player 1 does in the proof.)

5 Existence and computation of secure equilibria

The secure equilibria were introduced [5] in connection with model checking, for two-player games with outcomes in $\{0,1\}^2$. They were generalized into quantitative secure equilibria [3] for two-player games with outcomes in $\mathbb{R}^2$. The (quantitative) secure equilibria of a game are the NEs of another game obtained by changing the usual preference of each player into malevolent preference: instead of just trying to maximize her own payoff, she tries primarily to do so and, in case of ties, to minimize the opponent's payoff. Since these new preferences are strict linear orders, by Theorem 9 we know the following:

Corollary 10 If a game form with finitely many outcomes is determined via strategies of some sort, every derived game with real-valued payoffs (and the usual preferences) has a secure equilibrium using strategies of the same sort.

Computationally, the hard part of the proof of Theorem 9 is to find M from Paragraph 1. Potentially there are indeed exponentially many subsets to check, but it is shown in [14] (after Lemma 2.4) that it suffices to check linearly many of them, by dichotomy. I.e., to compute a Nash or secure equilibrium it suffices to decide $|O|$ times the winner of a derived win/lose game, and to compute twice a winning strategy.

Let us now apply the above complexity remark to parity games. A recent breakthrough [4] shows that deciding the winner and computing a winning strategy can both be done in quasi-polynomial time in parity games. To exploit this, let us (similarly to [14, Section 3.2]) define a priority game by the arena of a parity game and a function from its priorities (in $\mathbb{N}$) to $\mathbb{R}^2$: after an infinite play, the payoff for the first (second) player is the first (second) component of the pair associated with the least priority occurring infinitely often.

Corollary 11 Consider priority games with n vertices and m priorities.

1. Positional secure equilibria can be computed in $O(n^{\log(m)+7})$.
2. For fixed parameter m, there is an FPT-algorithm with runtime $O(n^6)+h(m)$.

PROOF In [4] the numbers are $O(n^{\log(m)+6})$ and $O(n^5)+g(m)$. By the above complexity remark, they have to be multiplied by m, which cannot be greater than n since each vertex carries one priority. □

6 Coq formal setup for Theorem 9

We use the Coq proof assistant along with the SSReflect proof language and the MathComp library,[1] especially the following theories: `fintype` (finite types with decidable equality), `finfun` (functions over finite domains), `finset` (finite sets), and `bigop` (iterated operators). We also use the `RelationClasses` theory from Coq's standard library, to facilitate the reasoning on relations that are not in the scope of MathComp's framework. As of now, the size of the development is about 1300 lines of Coq code.

6.1 Main definitions

We chose to start the formalization by defining games and Nash equilibrium in the most general way. This general setting was not compulsory for mechanizing the proofs we focus on, but it allows one to get a wider game-theory library as a basis for further developments.

First, we define strategies as follows, relying on the dependently-typed theory of Coq:

```
Section Generic.
Variables (Agt : Type) (Strat : Agt → Type).
Definition strategy := ∀a : Agt, Strat a.
```

`Agt` and `Strat` represents the space (a term more appropriate than "set" in Coq) of agents and strategies, respectively, and `strategy` represents the (dependently-typed) space of strategy profiles (mapping each agent to its space of strategies).

We then define game forms as follows:

```
Variable Outc : Type.
Record game_form := GameForm
{ preform :> strategy → Outc ;
  eq_strategy : ∀strat' strat, (∀x, strat x = strat' x) →
    preform strat = preform strat' }.
```

[1] `https://math-comp.github.io/math-comp/`

So a game form is simply defined as a function mapping a strategy profile to an outcome. Note that the `eq_strategy` extensionality property is required in this Coq setting since `strategy` is a function type and equality is not extensional in Coq. (To demonstrate that this property can be instantiated in practice, we also give another definition of 2-player game forms as functions of type $\texttt{Strat}_1 * \texttt{Strat}_2 \to \texttt{Outc}$, and provide a function `game_form_of_alt2` that converts any such function into a `game_form` record.)

We then define games as a game form endowed with a preference relation — "$\texttt{prefs a o}_1\ \texttt{o}_2$" means that agent `a` prefers $\texttt{o}_2$ over $\texttt{o}_1$.[2]

```
Record game := Game
{ form :> game_form ;
  prefs : Agt → Outc → Outc → bool }.
```

We then define what it means that a given strategy profile is a Nash equilibrium (in the Coq code this notion is split into several definitions, but for the sake of readability we give below a syntactically equivalent definition, in one go):

```
Definition is_NE (g : game) (strat : strategy) : Prop :=
  ∀a : Agt, ∀strat' : strategy, (∀b : Agt, a ≠ b → strat b = strat' b) →
  ¬ prefs g a (g strat) (g strat').
```

This means that each agent is stable (he or she has no incentive to change strategy assuming other agents keep their strategy). We then introduce a Σ-type gathering a profile strategy and a proof that it is an NE:

```
Definition ex_NE (g : game) : Type := {strat : strategy | is_NE g strat}.
```

Another useful notion is : "Player a can enforce a set S of outcomes (using some strategy s_a)":

```
Definition can_enforce_by
  (v : game_form) (a : Agt) (S : Outc → bool) (sa : Strat a) : Prop :=
  ∀strat : strategy, strat a = sa → v strat ∈ S.
Definition can_enforce (v : game_form) (a : Agt) (S : Outc → bool) : Type :=
  {sa : Strat a | can_enforce_by v a S sa}.
End Generic.
```

Next, we define the following enumerated types of players and win/lose outcomes as well as the natural preference relations over these outcomes:

```
Inductive player := player1 | player2. Inductive winlose_outc := win1 | win2.

Definition game_form_2 := game_form player. Definition game_2 := game player.

Definition winlose_prefs (a : player) (o1 o2 : winlose_outc) : bool :=
  match a, o1, o2 with
  | player1, win2, win1
  | player2, win1, win2 ⇒ true
  | _, _, _ ⇒ false
  end.
```

We then formalize the notion of winning strategy for a given player (using dependent types) as well as the notion of determinacy, following Definition 3:

```
Definition preferred_outc (a : player) : winlose_outc :=
  if a is player1 then win1 else win2.
```

[2]Note that no property is assumed by this definition: the preference relation is an arbitrary binary relation.

```
Definition win_strat
  (Strat : player → Type) (v : game_form_2 winlose_outc Strat)
  (a : player) (sa : Strat a) : Prop :=
  ∀strat : strategy Strat, strat a = sa → v strat = preferred_outc a.

Definition determined Strat (v : game_form_2 winlose_outc Strat) : Type :=
  {a : player & {sa : Strat a | win_strat v a sa}}.
```

Next, we formalize Definition 5 regarding the derived win/lose game from a two-player game form, and the notion of determined form:

```
Program Definition derivedWLGame (Outc : Type) (Strat : player → Type)
  (wl : Outc → winlose_outc) (v : game_form_2 Outc Strat)
: game_2 winlose_outc Strat := Game (GameForm (wl \o v) _) winlose_prefs.

Definition determined_form Outc Strat (v : game_form_2 Outc Strat) : Type :=
  ∀wl : Outc → winlose_outc, determined (derivedWLGame wl v).
```

In previous definitions, it should be noted that the strategy spaces of all players (declared by variable `Strat : Agt → Type`) are arbitrary types. As a result, formalizing the constraints $R_1 \subseteq S_1$ and $R_2 \subseteq S_2$ involved in Theorem 9 cannot be done using MathComp's inclusion of finite sets. Instead, we introduce another variable `(Strat_R : Agt → Type)` corresponding to $(R_a)_{a \in \texttt{Agt}}$ and formalize the inclusion as `(∀a : Agt, Strat_R a → Strat a)`.

We then extend the definitions presented up to now with the condition "*via* $\prod_{a \in \texttt{Agt}} R_a$". In particular, the two predicates below (whose body is omitted for conciseness) are straightforwardly defined from predicates `is_NE` and `determined_form`:

```
Definition is_NE_via (Agt Outc : Type) (Strat_R Strat : Agt → Type) :
  (∀a : Agt, Strat_R a → Strat a) → game Outc Strat → strategy Strat_R → Prop.

Definition determined_form_via (Outc : Type) (Strat_R Strat : player → Type) :
  (∀a : player, Strat_R a → Strat a) → game_form_2 Outc Strat → Type.
```

6.2 Results and proofs

The main result of the Coq formalization is given by the following theorem, which has been formally verified without relying on any axiom:

```
Theorem finite_equilibrium_transfer :
  ∀(Strat : player → Type) (_ : strategy player Strat)
    (Outc : finType) (g : game_2 Outc Strat)
    (Strat_R : player → Type) (incl : ∀a : player, Strat_R a → Strat a),
  StrictOrder (prefs g player1) →
  StrictOrder (prefs g player2) →
  determined_form_via incl (form g) →
  ex_NE_via incl g.
```

The first hypothesis of this Coq theorem (`strategy player Strat`) formalizes the requirement that the space of strategy profiles $\prod_{a \in \texttt{player}} S_a$ is nonempty. This formalized theorem corresponds precisely to Theorem 9. This theorem relies on a formal proof of Lemma 6, which consists of the following two lemmas:

```
Lemma determined_form_enforce_outc :
  ∀(Outc : Type) (Strat : player → Type) (v : game_form_2 Outc Strat),
  determined_form v ⇔ (∀S : Outc → bool,
                         can_enforce v player1 S
                       + can_enforce v player2 (predC S)).

Lemma determined_form_via_enforce_outc :
 ∀(Outc : Type) (Strat_R Strat : player → Type)
  (incl : ∀a : player, Strat_R a → Strat a)
  (v : game_form_2 Outc Strat),
  determined_form_via incl v ⇔ (∀S : Outc → bool,
                                  can_enforce_via incl v player1 S +
                                  can_enforce_via incl v player2 (predC S)).
```

Here, `(predC S)` denotes the complement of set `S`, and `+` denotes the disjoint union (analogous to the $\vee$ connector, but for `Type` arguments).

6.3 Confidence lemmas and Remark 4

We have also proven several results that were not needed for proving the main theorem, but that are helpful to give more intuition or increase the confidence one can have in the formalized definitions. For example, regarding `winlose_prefs` and `preferred_outc`, we have proven:

```
Lemma winlose_prefs_strict : ∀a : player, StrictOrder (winlose_prefs a).

Lemma preferred_outc_correct :
  ∀(a : player) (o : winlose_outc), ¬ winlose_prefs a (preferred_outc a) o.
```

Then, we have proven the following lemma that is a formal version of Remark 4:

```
Lemma determined_iff_NE :
  ∀(Strat : player → Type) (v : game_form_2 winlose_outc Strat),
  determined v * strategy Strat ⇔ ex_NE (derivedWLGame id v).
```

Here, symbols $\Leftrightarrow$ and $*$ are connectors taking `Type` arguments. They correspond respectively to the usual connectors $\leftrightarrow$ and $\wedge$ (taking `Prop` arguments). The left-hand-side of this equivalence has two parts: the fact that the win/lose game is determined, and the fact that the space of strategy profiles ($\prod_{a \in \texttt{player}} S_a$) is nonempty.

7 Isabelle formal setup for Theorem 9

We use standard Isabelle/HOL in ISAR proof style without any special libraries. The current proof code has approximately 1100 lines.

7.1 Main definitions

Before we define games, Nash equilibrium etc., we need to look at some technicalities concerning the preference order. Concerning the lifting of the preference order (Definition 7) we have the following Isabelle definitions for u, u on sets, and $\prec^{\mathscr{P}}$, respectively:

```
definition
  ucone :: "('a ⇒ 'a ⇒ bool) ⇒ 'a ⇒ ('a set)"
  where "ucone les x = {z. les x z}"

definition
  uCone :: "('a ⇒ 'a ⇒ bool) ⇒ ('a set) ⇒ ('a set)"
  where "uCone les Y = ⋃ ((ucone les) ` Y)"

definition
  lessP :: "('a ⇒ 'a ⇒ bool) ⇒ ('a set) ⇒ ('a set) ⇒ bool"
  where "lessP les A B = (finite A ∧
     (∃A'⊆A-B. A'≠{} ∧ A-(A'∪(uCone les A')) = B-(A'∪(uCone les A'))))"
```

Observe that compared to Definition 7, there is no explicit set of outcomes O, but the type parameter `'a` is used for the type of the outcomes. However, it is stated that $A \prec^{\mathscr{P}} B$ presupposes that A is finite. We conjecture that one could weaken the requirements even further by only requiring that the witness set A' is finite – this would be a topic for future work.

We proved in Isabelle that $\prec^{\mathscr{P}}$ is a strict partial order (Lemma 8, see also Subsec. 7.2), without the explicit hypothesis that O is finite (which would be difficult to formulate given that we have no explicit O in Isabelle). Our somewhat weaker implicit hypothesis that A is finite (see paragraph above) suffices.

In the proof of Theorem 9 we do not construct all kinds of pairs $A \prec^{\mathscr{P}} B$. Instead we only construct a pair by removing a single element from a set A and by replacing it with all the preferred ones w.r.t. a given order $\prec$. This construction is formalized in Isabelle as follows:

```
definition
  replaceWithPreferred :: "('a ⇒ 'a ⇒ bool) ⇒ 'a ⇒ 'a set ⇒ 'a set ⇒ 'a set"
  where "replaceWithPreferred les a A U = (A ∪ {a'∈U. les a a'})-{a}"
```

Now let us consider game forms and games (Definition 1). In the current Isabelle formalization, we restrict ourselves to two players, which is sufficient to formalize Theorem 9. This is in contrast to the Coq formalization of Section 6. Also, the dependent types of Coq make generic definitions (for an arbitrary number of players) easier. In the Isabelle formalization, there is nothing to define about *strategies* and the *outcomes*: they are simply type parameters. A *game form* is then given by an outcome function that maps a pair of strategies to an outcome:

```
type_synonym ('O,'S1,'S2) game_form = "('S1 * 'S2) ⇒ 'O"
```

and a *game* is obtained by adding one preference relation for each of the two players:

```
type_synonym ('O,'S1,'S2) game =
   "('O ⇒ 'O ⇒ bool) * ('O ⇒ 'O ⇒ bool) * (('O,'S1,'S2) game_form)"
```

Then there are functions *pref1*, *pref*2, and *gf* that extract each of the three components of a game in the obvious way. These are used in the following definition of a Nash equilibrium (Definition 2), i.e., a function taking a game and two strategies (one per player) and telling whether they constitute an NE:

```
definition
  isNash :: "(('O,'S1,'S2) game) ⇒ 'S1 ⇒ 'S2 ⇒ bool"
  where "isNash g s1 s2 =
     ((∀s1'. ¬(pref1 g) ((form g) (s1,s2)) ((form g) (s1',s2))) ∧
      (∀s2'. ¬(pref2 g) ((form g) (s1,s2)) ((form g) (s1,s2'))))"
```

We now give the definition of a determined (via ...) game (Definition 3):

```
definition
  determined :: "((bool,'S1,'S2) game) ⇒ ('S1 set) ⇒ ('S2 set) ⇒ bool"
  where "determined g R1 R2 = ((∃s1∈R1. ∀s2. (form g) (s1,s2) = True) ∨
                               (∃s2∈R2. ∀s1. (form g) (s1,s2) = False))"
```

We now give the definition of the *derived win/lose game* (Definition 5):

```
definition
  derivedWLGame :: "(('O,'S1,'S2) game_form) ⇒ ('O set) ⇒ ((bool,'S1,'S2) game)"
  where "derivedWLGame gf Ou =
     ((λ ou p. p∧¬ou), (λ ou p. ou∧¬p), (λou. ou∈Ou)∘gf)"
```

The function takes as input a game with outcomes of type `'O` and a set O[3] of outcomes, those for which the first player wins. The derived win/lose game is the game with outcomes of Boolean type[4], where the preference relations say that Player 1 says $\mathit{false} \prec \mathit{true}$ (expressed by the λ-term $\lambda o\ p.p \wedge \neg o$ and vice versa for Player 2), and the outcome function is the original outcome function composed with a function that says "*true*" for all values in O.

We now give the definition of a determined (via ...) game form (Definition 5):

```
definition
  determinedForm :: "(('O,'S1,'S2) game_form) ⇒ ('S1 set) ⇒ ('S2 set) ⇒ bool"
  where "determinedForm gf R1 R2 = (∀ Ou. determined (derivedWLGame gf Ou) R1 R2)"
```

In the paper-and-pencil version, "Player 1 can enforce P" (Definition 5) is defined as an overapproximation, i.e., Player 1 might even be able to enforce a subset of P. We also have the Isabelle versions of this notion, but it turned out to be more useful in the formal development to define an *exact* notion, i.e., exactly the outcomes that may occur using a given strategy. We give the definition for Player 1, but there is an analogous definition for Player 2:

```
definition
  enforceSet1 ::  "(('O,'S1,'S2) game_form) ⇒ 'S1 ⇒ ('O set)"
  where "enforceSet1 f s1 = {ou. ∃s2. f(s1,s2) = ou}"
```

7.2 Results and proofs

In the presentation of the proof path we choose a top-down approach, i.e., we present the main result (Theorem 9) and give some of the lemmas needed to show it.

```
theorem equilibrium_transfer_finite :
  assumes finiteO : "finite (range (form g))"
      and trans1 : "⋀a b c. (pref1 g) a b ⟹ (pref1 g) b c ⟹ (pref1 g) a c"
      and irref1 : "⋀a. ¬(pref1 g) a a"
      and trans2 : "⋀a b c. (pref2 g) a b ⟹ (pref2 g) b c ⟹ (pref2 g) a c"
      and irref2 : "⋀a. ¬(pref2 g) a a"
      and det : "determinedForm (form g) R1 R2"
  shows "∃s1∈R1. ∃s2∈R2. isNash g s1 s2"
```

[3] In Isabelle the letter "o" has a reserved meaning which is why we used "ou" instead.

[4] In the Definition 5, we assumed that the outcome of a win/lose game is $(1,0)$ (Player 1 gets 1, Player 2 gets 0) or $(0,1)$, which may be intuitive, but it is simpler to say the the outcome is *true* (first player wins) or *false*.

The Isabelle version, like the Coq version, corresponds precisely to Theorem 9.

Unlike the Coq version, the Isabelle version does not have an explicit assumption that the sets of strategies are nonempty. As a matter of fact, they are formalized as types in Isabelle/HOL (namely, the type parameters `'S1`, `'S2` in the definition of a `game`), which are necessarily nonempty.

The proof has 153 lines of ISAR code but uses various lemmas. We now sketch how the paragraphs of the paper-and-pencil proof of Theorem 9 translate into Isabelle.

Paragraph 1: We need 22 lines of code to exhibit M, show its finiteness, and exhibit s_1. This relies on two lemmas adding up to Lemma 8:

```
lemma lift_irreflexive :
  shows "¬(lessP les A A)"
```

```
lemma lift_transitive :
  assumes les_irr : "⋀x. ¬ (les x x)"
      and les_trans : "⋀x y z. (les x y) ⟹ (les y z) ⟹ (les x z)"
      and AB : "lessP les A B"
      and BC : "lessP les B C"
  shows "lessP les A C"
```

The first has 7 lines of proof, but we found the second one surprisingly hard with 160 lines of proof code. One can observe a striking discrepancy between the shortness of the paper-and-pencil proofs and the Isabelle proofs; probably the paper-and-pencil proofs hide too many details, while the Isabelle proofs are more complicated than necessary.

Paragraph 2: we need 8 lines to exhibit m and construct M'. The proof of $M \prec_1^{\mathscr{P}} M'$ relies on a lemma stating `"lessP les A (replaceWithPreferred les a A U)"` under the conditions $a \in A$ and finiteness of A. This simple lemma has nonetheless a proof of 17 lines.

Paragraph 3: The fact that Player 1 cannot enforce M' has a proof of 27 lines. To show that Player 2 can therefore enforce the complement, we use a formalisation of one direction of Lemma 6:

```
lemma determined_form_via_impl_enforce_outc :
  assumes det : "determinedForm gf R1 R2"
  shows "(∀ Ou. (∃s1∈R1. enforceSet1 gf s1 ⊆ Ou)∨
                (∃s2∈R2. enforceSet2 gf s2 ⊆ (range gf) - Ou))"
```

which has a proof of 35 lines. We then need 11 lines to prove that Player 2 can actually enforce the complement. Another 13 lines are needed for the proof of $v(s_1,s_2) \in M \cap (O \backslash M') = \{m\}$.

Paragraph 4: we show that Player 1 has no incentive to deviate (32 lines) and likewise for Player 2 (12 lines). This implies that we have found a Nash equilibrium.

7.3 Remark 4

The right-to-left implication of Remark 4 is formalized as follows:

```
lemma someone_wins :
  assumes isNashWL : "isNash ((derivedWLGame gf Ou)::((bool,'S1,'S2) game)) s1 s2"
     (is "isNash ?wlG s1 s2")
  shows "(∀s2'. (form ?wlG) (s1,s2') = True) ∨ (∀s1'. (form ?wlG) (s1',s2) = False)"
```

It has 35 lines of proof. The other direction, albeit not needed for our main result, would also be interesting to prove and we plan to do it as future work.

8 Conclusion and future work

We have formally proven an existence theorem of Nash equilibrium in both Coq and Isabelle proof assistants. This theorem is applicable to every two-player game with finitely many outcomes and strict partial order preferences, provided that all derived win/lose games from the original game are determined. Thus, the theorem proves a transfer from winning strategies to multi-outcome Nash equilibrium, where the latter notion is a faithful generalization of the former.

Also, this dual formalization effort gave us the opportunity to sketch a comparison between Coq and Isabelle via a case study.

Regarding the underlying logic, both proof assistants rely on a higher-order logic but that of Coq is more expressive especially thanks to the support of dependent types, which allowed us to formalize the basic notions of game theory in a more general way, e.g., for an arbitrary number of players with arbitrary strategy spaces. We suspect that generalizing the Isabelle formalization to an arbitrary number of players would be notationally very heavy.

Regarding the proof languages, we relied on an SSReflect proof style with a systematic use of forward-chaining in longer proofs, in order to put forth the structure of the Coq proofs. On the other hand, the Isabelle ISAR style allows for human-readable *declarative* proof code, but we hope to increase the degree of automation somewhat, not too much, to avoid distraction by too much detail. Also, our formalization work made us aware of the fact that we tend to write too terse paper-and-pencil proofs, and suggested some improvements in the paper-and-pencil formulation.

Various generalizations of the result could be proven in both proof assistants, and some work remains to be done to benefit from the mutual insemination between theory (paper-and-pencil proofs) and practice (proof assistants) and between the two proof assistants. We see four natural, independent directions to extend this article:

- Proving formally the existence of secure equilibria.
- Since our theorem transforms determinacy into existence of NE, we plan to feed it with, e.g., the positional determinacy of parity games, which has already been formalized in Isabelle [6], as mentioned in the introduction. (Technical issues may arise at the interface, though.)
- One challenging goal would be to prove the full [14, Theorem 2.7]. As mentioned in the introduction, our result is a slight weakening of [14, Lemma 2.4], which is the base case of the proof by transfinite induction leading to [14, Theorem 2.7]. Formalizing this proof would require us to choose a convenient representation of the countable ordinals with an associated induction proof principle. This sounds more tractable in Coq than in Isabelle.
- Our result is weaker than [14, Lemma 2.4] for a second reason that we have not mentioned yet. This second reason is indeed orthogonal to the transfinite induction: we consider preferences that are strict partial orders instead of just acyclic binary relations. Extending our result accordingly can be done by defining a transitive closure operator, proving that it maps acyclic binary relations onto strict partial orders, and proving that an NE for bigger preferences is also an NE for smaller preferences. This sounds doable both in Coq and in Isabelle.

References

[1] Jade Alglave & Assia Mahboubi (2011): *A Generic Formalised Framework for Reasoning About Weak Memory Models*. Available at `https://hal.inria.fr/inria-00604656`. Working paper or preprint.

[2] K. Appel & W. Haken (1976): *Special announcement*. Discrete Mathematics 16(2), pp. 179–180, doi:10.1016/0012-365X(76)90147-3.

[3] Thomas Brihaye, Véronique Bruyère & Julie De Pril (2010): *Equilibria in Quantitative Reachability Games*. In Farid Ablayev & Ernst W. Mayr, editors: Computer Science – Theory and Applications: 5th International Computer Science Symposium in Russia, CSR 2010, Kazan, Russia, June 16-20, 2010. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 72–83, doi:10.1007/978-3-642-13182-0_7.

[4] Cristian S Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li & Frank Stephan (2017): *Deciding Parity Games in Quasi-Polynomial Time*. Accepted at STOC 2017.

[5] Krishnendu Chatterjee, Thomas A. Henzinger & Marcin Jurdziński (2006): *Games with secure equilibria*. Theoretical Computer Science 365(1), pp. 67 – 82, doi:10.1016/j.tcs.2006.07.032.

[6] Christoph Dittmann (2016): *Positional Determinacy of Parity Games*. Available at `www.isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf`.

[7] E.A. Emerson & C.S. Jutla (1991): *Tree automata, mu-calculus and determinacy*. In: Proceedings 32nd Annual Symposium of Foundations of Computer Science, IEEE Comput. Soc. Press, pp. 368–377, doi:10.1109/sfcs.1991.185392.

[8] Allan Gibbard (1973): *Manipulation of Voting Schemes: A General Result*. Econometrica 41(4), pp. 587–601, doi:10.2307/1914083.

[9] Georges Gonthier (2008): *Formal Proof—The Four-Color Theorem*. Notices of the American Mathematical Society 55(11), pp. 1382–1393. Available at `http://www.ams.org/notices/200811/tx081101382p.pdf`.

[10] Y. Gurevich & L. Harrington (1982): *Trees, automata, and games*. In: STOC'82, ACM Press, pp. 60–65, doi:10.1145/800070.802177.

[11] Vladimir Gurvich (1975): *Solution of positional games in pure strategies*. USSR Comput. Math. and Math. Phys. 15(2), pp. 358–371, doi:10.1016/0041-5553(75)90042-7. Originally written in Russian.

[12] Vladimir Gurvich (1989): *Equilibrium in pure strategies*. Soviet Math. Dokl 38(3), pp. 597–602.

[13] Stéphane Le Roux (2009): *Acyclic Preferences and Existence of Sequential Nash Equilibria: A Formal and Constructive Equivalence*. In Stefan Berghofer, Tobias Nipkow, Christian Urban & Makarius Wenzel, editors: Theorem Proving in Higher Order Logics: 22nd International Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 293–309, doi:10.1007/978-3-642-03359-9_21.

[14] Stéphane Le Roux (2014): *From winning strategy to Nash equilibrium*. Math. Log. Q. 60(4-5), pp. 354–371, doi:10.1002/malq.201300034.

[15] Pierre Lescanne & Matthieu Perrinel (2012): *"Backward" coinduction, Nash equilibrium and the rationality of escalation*. Acta Informatica 49(3), pp. 117–137, doi:10.1007/s00236-012-0153-3.

[16] Donald A. Martin (1975): *Borel Determinacy*. The Annals of Mathematics 102(2), pp. 363–371, doi:10.2307/1971035.

[17] Erik Martin-Dorel & Sergei Soloviev (2017): *A Formal Study of Boolean Games with Random Formulas as Pay Functions*. Available at `https://www.irit.fr/publis/ACADIE/IRIT-RR-2017-01-FR.pdf`. Research Report.

[18] H Moulin & B Peleg (1982): *Cores of effectivity functions and implementation theory*. Journal of Mathematical Economics 10(1), pp. 115 – 145, doi:10.1016/0304-4068(82)90009-X.

[19] René Vestergaard (2006): *A constructive approach to sequential Nash equilibria*. Information Processing Letters 97(2), pp. 46 – 51, doi:10.1016/j.ipl.2005.09.010.