

FINITE NASH EQUILIBRIA AND DEPENDENT MIXED STRATEGIES IN LEAN

ARTHUR FREITAS RAMOS, DAVID BARROS HULAK, AND RUY J. G. B. DE QUEIROZ

ABSTRACT. We present two complementary Lean 4 developments of finite Nash equilibrium. The pure-game layer uses player-specific legal action sets, distinguishes one-way generalized ordinal potentials from bidirectional ordinal potentials, and proves potential-maximizer certificates, a local-maximum characterization, absence of strict-improvement cycles, and weak acyclicity. The mixed-game layers use either a common finite action type or a dependent finite action type for each player. They prove payoff equalization on equilibrium supports, Dirac payoff identities, continuity of the normalized payoff-excess map, and mixed equilibrium existence. We explain the finite-sum and sum-of-squares arguments and the reindexing bridge to an attributed, proved product-of-simplices Brouwer theorem. An additional legal-subtype interface is separated from the declarations selected by the registry comparator. The account is tied to two immutable source snapshots and dated verification records. It documents established mathematics and its formal interfaces, without claiming a new equilibrium theorem or formalization priority.

1. THE MATHEMATICAL RESULT AND THE TWO ARTIFACTS

Nash’s finite-game existence theorem states that every game with finitely many players, finitely many nonempty action sets, and real payoffs admits a mixed equilibrium [7, 6]. A different, more elementary existence argument applies to finite potential games: maximize a scalar potential and rule out profitable unilateral deviations. The two arguments share the equilibrium concept but use different hypotheses and proof tools.

This article gives a consolidated account of two Lean-native artifacts. The native artifact, `nash-equilibrium-lean`, is inspected at commit `6dd83f3b004c0318e52f4c3cb272d909efca321d` [10]. It contains the pure potential-game interface and a common-action mixed layer. The dependent artifact, `dependent-nash-equilibrium-lean`, is inspected at commit `2f65710af26a24a26660dd1e92d09adf1697a527` [8]. It gives each player its own action type. All source links in this paper refer to these commits.

The formalization contribution is an explicit game API, a careful separation of potential predicates, a finite improvement-path interface, and native mixed-game adapters to fixed-point infrastructure. The mathematics is classical [5]. Earlier formalizations include the Isabelle/HOL AFP development by the present manuscript’s authors [12], and a Ssreflect/Coq library for potential games and best-response dynamics [1]. The reused Lean development of Lyu and Li already addresses Scarf, Brouwer, and Nash [3]; its fixed-point proofs are attributed dependencies here. We do not claim the first formalization of Nash’s theorem, a cross-prover translation, or a machine-checked equivalence with AFP.

The artifacts use Lean 4 and Mathlib [2, 13]. Finite spaces are supplied through `Fintype` instances, with `DecidableEq` on players and on each relevant action type. The mathematical

Date: October 1, 2026.

2020 Mathematics Subject Classification. 91A10, 91A11, 68V20.

Key words and phrases. finite games, Nash equilibrium, ordinal potential, better-response paths, mixed strategies, dependent types, Brouwer fixed point, Lean 4.

Author ORCID identifiers: Arthur Freitas Ramos, 0009-0003-3568-0325; David Barros Hulak, 0009-0002-8056-1774; Ruy J. G. B. de Queiroz, 0000-0003-1482-0977.

Copyright 2026 the authors. Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0).

statements below are readable independently of Lean. Source names identify where the encoded definitions and proofs can be found; the distinction between selected registry declarations and additional repository results is maintained throughout.

2. PURE GAMES AND THE ORDER ON UTILITIES

Let P and A be finite types. For each $i \in P$, let $L_i \subseteq A$ be a nonempty finite legal action set. A profile is a function $s : P \rightarrow A$; it is legal if $s_i \in L_i$ for every i . Write $s[i \leftarrow a]$ for the profile changing only player i to a . Payoffs are functions $u_i : A^P \rightarrow U$, where U is an ordered utility type.

These data form `NashEquilibrium.Game` in `Basic.lean`. The structure stores payoffs even on illegal profiles, but all equilibrium comparisons below start from a legal profile and use legal deviations. The field `nonempty_strategies` supplies a legal action for each player. No nonemptiness assumption on P is made. If P is empty, its unique profile is legal and all playerwise comparisons are vacuous.

Definition 2.1. For a utility preorder, a pure Nash equilibrium is a legal profile s such that

$$(1) \quad \forall i \in P \ \forall a \in L_i, \quad \neg(u_i(s) < u_i(s[i \leftarrow a])).$$

This is exactly `IsNash`. For a general preorder, an incomparable utility is not a strict improvement. Consequently (1) must not be silently replaced by $u_i(s[i \leftarrow a]) \leq u_i(s)$. The repository defines a best response using these weak comparisons and proves `isNash_iff_bestResponse` under a *linear* order on U . It also defines dominant actions and proves that a profile of dominant actions is Nash under a preorder. The selected potential theorems impose linear orders on both utilities and potential values, as required by their formal statements.

3. POTENTIAL CERTIFICATES AND FINITE IMPROVEMENT PATHS

Assume U and V are linearly ordered, and let $\Phi : A^P \rightarrow V$. The repository keeps the following predicates separate.

Definition 3.1. A *generalized ordinal potential* satisfies, for every legal s and $a \in L_i$,

$$(2) \quad u_i(s) < u_i(s[i \leftarrow a]) \implies \Phi(s) < \Phi(s[i \leftarrow a]).$$

An *ordinal potential* satisfies the same statement with $\iff$ in place of $\implies$.

These are `IsGeneralizedOrdinalPotential` and `IsOrdinalPotential`. The bidirectional condition is the usual strict sign agreement. In particular, applying it to a deviation and its reverse rules out a strict potential change along a payoff-neutral deviation. The one-way condition permits such a change.

Theorem 3.2. Assume that utilities and potential values are linearly ordered.

- (1) If Φ is a generalized ordinal potential, there is a legal profile s maximizing Φ over all legal profiles, and this s is Nash.
- (2) If Φ is an ordinal potential, a profile is Nash if and only if it is legal and $\Phi(s[i \leftarrow a]) \leq \Phi(s)$ for all i and $a \in L_i$.
- (3) For a generalized ordinal potential, there is no nonempty finite cycle of strict better responses. From every legal starting profile there is a finite better-response path to a Nash equilibrium, allowing a zero-step path if the starting profile is already Nash.

For the first assertion, the finite set of legal profiles is nonempty: choose one legal action for each player. A maximum of Φ exists in any linear order because the set is finite. A legal strict improvement would produce a legal profile with larger potential by (2), contradicting maximality. No real-valued potential is needed. The proof uses `Finset.exists_max_image`.

The implementation proves this generalized maximizer certificate as `exists_nash_maximizing_generalized_ordinal_potential`. The *selected* registry theorem is its standard-ordinal

specialization, `NashEquilibrium.Palomar.exists_nash_maximizing_ordinal_potential`, which returns both equilibrium and global potential maximality. The selected local-maximum equivalence is `isNash_iff_potential_local_maximum` in the same namespace. Its forward implication uses the reverse direction of the ordinal condition; its backward implication uses (2).

The stronger hypothesis for the equivalence is necessary. Consider one player with two actions, constant payoff, and potentials 0 and 1. Every profile is Nash and every potential is generalized, but the action with potential 0 is not a local potential maximum. The repository's `Examples.lean` contains checked counterexamples illustrating the potential distinction, alongside Prisoner's Dilemma, coordination, matching-pennies, and rock-paper-scissors examples.

A better-response edge $s \rightarrow t$ consists of legal profiles and a legal unilateral deviation strictly improving the deviator's utility. `BetterResponsePath` is inductively generated by one edge and prefixing an edge to a nonempty path. Each edge strictly increases Φ ; induction and transitivity give $\Phi(s) < \Phi(t)$ along every such path. A cycle would imply $\Phi(s) < \Phi(s)$.

For reachability, `IsWeaklyAcyclic` explicitly permits either $t = s$ or a nonempty path from s to a Nash profile t . The proof makes the reversed path relation well founded using finiteness, transitivity, and irreflexivity. Well-founded induction then chooses a strictly profitable deviation whenever the current legal profile is not Nash and concatenates it with the induction hypothesis. The selected `no_betterResponse_cycle_of_generalized_ordinal_potential` and `weaklyAcyclic_of_generalized_ordinal_potential` certify these properties. They specify no scheduling rule, numerical convergence rate, or equilibrium-computation complexity.

4. FINITE MIXED PROFILES AND EXPECTED PAYOFFS

We now use real payoffs. Let A_i be a finite nonempty action type for each player $i \in P$, and put $S = \prod_{i \in P} A_i$. The payoff table is now $u_i : S \rightarrow \mathbb{R}$. A mixed profile is an independent distribution for each player:

$$(3) \quad x_i : A_i \rightarrow \mathbb{R}, \quad x_i(a) \geq 0, \quad \sum_{a \in A_i} x_i(a) = 1.$$

Write $K = \prod_i \Delta(A_i)$ for the feasible set.

The native mixed layer takes a *common* type A and represents profiles as functions $P \rightarrow (A \rightarrow \mathbb{R})$. Unlike the native pure `Game`, its `MixedGame` has no legal-set field: every player may choose every $a \in A$. The dependent layer represents a profile as $(i : P) \rightarrow (A_i \rightarrow \mathbb{R})$, using a family of action types indexed by players. The definitions that follow have the same finite-sum form in both layers.

For a pure profile $s \in S$, define the opponents' weight

$$W_i(x, s) = \prod_{j \in P \setminus \{i\}} x_j(s_j).$$

The source enumerates full pure profiles while fixing the deviator's coordinate:

$$(4) \quad D_i(x, a) = \sum_{s \in S} \mathbf{1}_{\{s_i = a\}} W_i(x, s) u_i(s),$$

$$(5) \quad U_i(x) = \sum_{a \in A_i} x_i(a) D_i(x, a).$$

These are `opponentWeight`, `pureDeviationPayoff`, and `mixedPayoff` in the native `Mixed.lean` and dependent `Basic.lean` modules. The indicator in (4) ensures each opponents' profile is counted once, rather than once per action of player i .

Definition 4.1. A mixed Nash equilibrium is $x \in K$ such that

$$(6) \quad \forall i \in P \ \forall a \in A_i, \quad D_i(x, a) \leq U_i(x).$$

The predicate `IsMixedNash` includes feasibility as well as these inequalities. This finite pure-deviation formulation excludes an improving mixed deviation as well: for any distribution y_i , $\sum_a y_i(a) D_i(x, a) \leq \sum_a y_i(a) U_i(x) = U_i(x)$. This is an elementary explanation of the definition; it is not a separate selected arbitrary-mixed-deviation equivalence theorem.

Theorem 4.2. *If x is a mixed Nash equilibrium and $x_i(a) > 0$, then $D_i(x, a) = U_i(x)$. In particular, if $D_i(x, a) < U_i(x)$, then $x_i(a) = 0$.*

Proof. Each summand $x_i(b)(U_i(x) - D_i(x, b))$ is nonnegative by (3) and (6), and

$$\sum_b x_i(b)(U_i(x) - D_i(x, b)) = U_i(x) \sum_b x_i(b) - U_i(x) = 0.$$

Every summand is therefore zero. For $b = a$, the positive factor $x_i(a)$ can be cancelled. The second assertion follows by contradiction from the first and nonnegativity of probabilities. $\square$

Both selected `mixedNash_support_payoff_eq` theorems implement this argument. They need no additional global nonemptiness assumption on action types: the equilibrium hypothesis already supplies the relevant normalization. Equalization applies only to positive-probability actions; an unused action may tie the equilibrium payoff or have a lower payoff.

For a pure profile s , its Dirac embedding assigns probability one to s_i and zero to every other action. The source proves

$$D_i(\delta_s, a) = u_i(s[i \leftarrow a]), \quad U_i(\delta_s) = u_i(s).$$

Products of Dirac weights eliminate all opponents' profiles except the one agreeing with s , and the finite sum reduces to a single term. Thus the condition of no profitable pure deviation yields a mixed Dirac equilibrium. These identities are additional library lemmas, not separately selected registry declarations. For the common-action mixed game they compare against *every* common action; a restricted pure-game equilibrium is not automatically an equilibrium of that game.

5. THE NORMALIZED PAYOFF-EXCESS MAP

For an arbitrary real coordinate profile x , define

$$(7) \quad e_i(x, a) = \max\{0, D_i(x, a) - U_i(x)\}, \quad E_i(x) = \sum_{a \in A_i} e_i(x, a),$$

and

$$(8) \quad F(x)_i(a) = \frac{x_i(a) + e_i(x, a)}{1 + E_i(x)}.$$

The source names are `excess`, `excessSum`, and `nashMap`. Because $E_i(x) \geq 0$ for every x , the denominator is strictly positive on the entire ambient space, not just on K .

If $x \in K$, the numerator is nonnegative and

$$\sum_a F(x)_i(a) = \frac{\sum_a x_i(a) + \sum_a e_i(x, a)}{1 + E_i(x)} = 1.$$

Hence $F(K) \subseteq K$, as proved by `nashMap_isMixedProfile`. The decisive algebraic step is the following.

Lemma 5.1. *If $x \in K$ and $F(x) = x$, then all excesses are zero, and x is a mixed Nash equilibrium.*

Proof. Fix a player i and abbreviate $T = E_i(x)$. Multiplying a fixed-point coordinate identity by $1 + T$ yields

$$(9) \quad e_i(x, a) = x_i(a)T \quad \text{for every } a \in A_i.$$

If $T = 0$, every excess is zero. Suppose instead $T > 0$. Whenever $x_i(a) > 0$, equation (9) makes the excess positive, so the maximum in (7) is its second argument and

$$D_i(x, a) = U_i(x) + x_i(a)T.$$

For zero-probability actions, multiplying by $x_i(a)$ makes the corresponding identity hold as well. Summing therefore gives

$$U_i(x) = \sum_a x_i(a)D_i(x, a) = U_i(x) \sum_a x_i(a) + T \sum_a x_i(a)^2,$$

so $T \sum_a x_i(a)^2 = 0$. Normalization ensures some $x_i(a) > 0$; thus $\sum_a x_i(a)^2 > 0$, contradicting $T > 0$. It follows that $T = 0$ and each $D_i(x, a) - U_i(x) \leq 0$. Applying this to every player gives (6). $\square$

This sum-of-squares proof follows `fixedPoint_excess_zero`, followed by `mixedNash_of_nashMap_fixedPoint`, in both mixed modules. It does not infer equilibrium merely from nonnegativity of total excess, nor cancel a probability that might be zero. Conversely, an equilibrium has zero excess by definition and hence is fixed by (8); the existence proof uses the forward implication of Lemma 5.1.

6. GEOMETRY AND THE PROVED FIXED-POINT DEPENDENCY

6.1. Finite-dimensional topology. The feasible set K is closed: it is the intersection of coordinate half-spaces $x_i(a) \geq 0$ and normalization hyperplanes $\sum_a x_i(a) = 1$. Every feasible coordinate is at most one, since it is a nonnegative summand of a sum equal to one. Hence

$$K \subseteq \prod_{i \in P} \prod_{a \in A_i} [0, 1].$$

The finite product box is compact, and the closed subset K is compact. Convexity follows directly from (3): a convex combination preserves nonnegativity and normalization. Nonemptiness is witnessed by the uniform distributions $x_i(a) = 1/|A_i|$ when each A_i is nonempty.

These facts are proved as `mixedProfiles_closed`, `mixedProfiles_compact`, `mixedProfiles_s_convex`, and `mixedProfiles_nonempty` in the native `MixedTopology.lean` and dependent `Topology.lean`. Their proofs use finite sums, coordinate evaluations, compact interval products, and closedness. Payoffs are arbitrary lookup values on a finite pure-profile space; no continuity assumption on the payoff table is necessary.

Each $D_i(x, a)$ is a finite sum of finite products of coordinates with fixed real coefficients. The functions D_i , U_i , e_i , and E_i are therefore continuous. Division in (8) is continuous because the denominator never vanishes. The two modules prove `continuous_nashMap` on the full ambient real coordinate space. Their conditional fixed-point reduction accepts a supplied fixed point; it does not postulate that such a point exists.

6.2. The Brouwer bridge.

Theorem 6.1. *Every finite real-payoff game with a nonempty action type for each player has a mixed Nash equilibrium. The common-action version assumes one finite nonempty type available to every player. Neither version requires the player type to be nonempty.*

The existence modules use the theorem `Brouwer_Product` from `Gametheory/Brouwer_product.lean`. Its interface concerns a continuous self-map of

$$\prod_{k \in \text{Fin}(n)} \Delta(\text{Fin}(c_k)), \quad c_k \in \mathbb{N}_{>0}.$$

For this interface the finite player index type is inhabited, so $n > 0$. It returns a fixed point. The imported Scarf-to-Brouwer development is *proved source*, not a new fixed-point axiom introduced by either Nash library. The four included `Gametheory` files are attributed to the

MIT-licensed `math-xmum/Brouwer` development at commit `09941e849a81e520cc0cc53220f10f8e5f4768e0` [4, 3]. The native copies are unchanged from that pin; the dependent copy of `Scarf.lean` narrows its opening imports while retaining the proof body. This paper does not rederive or claim those upstream fixed-point proofs.

For nonempty P , choose a finite equivalence $\pi : P \simeq \text{Fin}(|P|)$. In the common-action bridge, choose $A \simeq \text{Fin}(|A|)$ and set $c_k = |A|$ for every k . In the dependent bridge, set

$$c_k = |A_{\pi^{-1}(k)}|, \quad \eta_k : A_{\pi^{-1}(k)} \simeq \text{Fin}(c_k).$$

Nonempty action types ensure $c_k > 0$. Reindexing the probability coordinates identifies feasible game profiles with this product of standard simplices. The normalized map transports to a continuous self-map of the product using the feasibility and continuity lemmas. The theorem `Brouwer_Product` supplies a fixed point, which is transported back and converted into equilibrium by Lemma 5.1.

The dependent implementation must also transport actions along $\pi^{-1}(\pi(i)) = i$. These casts appear in its playerwise equivalence `emPlayer`; the fixed-point calculation uses heterogeneous equality to reconcile coordinates whose action types have propositionally equal indices. This is a type-correct reindexing issue, not an additional mathematical assumption or the addition of dummy actions.

If P is empty, the existence modules construct the unique empty profile directly; feasibility and fixed-point equality are vacuous. They do not invoke a product theorem requiring an inhabited finite index type in that branch. The native theorem nevertheless retains its stated `NonemptyMove` hypothesis. The dependent assumption $\forall i, A_i \neq \emptyset$ is vacuous for empty P .

The direct existence proofs in `MixedBrouwer.lean` and `Existence.lean` use the simplex-product bridge, continuity, and preservation. The separate compactness and convexity lemmas document the expected geometry; the final proof does not apply an abstract compact-convex fixed-point axiom to them.

7. AN ADDITIONAL RESTRICTED-ACTION INTERFACE

The dependent repository also contains `Restricted.lean`. Given an ambient finite type A , legal finite sets $L_i \subseteq A$, and an ambient payoff table, it defines

$$A_i = \{a : A \mid a \in L_i\}$$

and restricts payoffs by forgetting each subtype membership proof. Nonempty legal sets make these dependent types nonempty, so `exists_mixedNash_restricted` applies Theorem 6.1 to the restricted game. No padding with unavailable or dummy actions is used.

For a dependent pure profile s , let $\bar{s}$ be its ambient image. The file proves that forgetting subtype proofs commutes with unilateral deviation, then establishes

$$(10) \quad \delta_s \text{ is mixed Nash in the restricted game} \iff \forall i \forall a \in L_i, \quad u_i(\bar{s}[i \leftarrow a]) \leq u_i(\bar{s}).$$

This is `restricted_dirac_mixed_nash_iff`. Its proof uses the Dirac payoff identities from the mixed layer of Section 4.

The exact scope matters: equation (10) is a Dirac-profile and pure-deviation correspondence. The file does not construct an equivalence between arbitrary ambient mixed profiles and dependent mixed profiles. These restricted-action results are additional repository results, are not imported by the selected `Solution.lean`, and are not selected by the dependent entry's comparator. Their exposition here must not be read as expanding that entry's verified comparison surface.

8. SELECTED DECLARATIONS AND HISTORICAL VERIFICATION

The native registration is PALOMAR-2026-09-07-000009, version 1 [11]. In the namespace `NashEquilibrium.Palomar`, its six selected theorems are:

- `exists_nash_maximizing_ordinal_potential`;

- `isNash_iff_potential_local_maximum`;
- `no_betterResponse_cycle_of_generalized_ordinal_potential`;
- `weaklyAcyclic_of_generalized_ordinal_potential`;
- `mixedNash_support_payoff_eq`;
- `exists_mixedNash`.

The dependent registration is PALOMAR-2026-09-07-000014, version 1 [9]. Its selected namespace is `DependentNash.Palomar`, and its two theorems are `exists_mixedNash` and `mixedNash_support_payoff_eq`. The wrappers in each `Solution.lean` delegate to implementation theorems; `comparator.json` identifies the statement-side and solution-side declarations and the selected definitions.

The two records report the same Lean toolchain, `leanprover/lean4:v4.33.0`, and Mathlib revision `db584cd6d46c92f209a44c0f1c829460d327499d`. For the native entry, the recorded verification time is September 7, 2026, at 13:45:01 UTC, with workflow run 34128531761. For the dependent entry, it is September 7, 2026, at 20:52:40 UTC, with workflow run 34160743338. Both permit only `propext`, `Quot.sound`, and `Classical.choice` in the selected theorem axiom audit. The records include an external `nanoda` checker revision. These dated records are evidence about the configured declarations; registration is not a claim of journal peer review.

The native `Challenge.lean` contains two deliberate statement-side `sorry` holes, for weak acyclicity and mixed existence; the other selected targets, including support equalization, are proved there directly. The dependent challenge has two deliberate holes. The corresponding proofs are in the implementation and `Solution.lean`; the challenge files are not being presented as complete proof implementations. The inspected implementation modules and solution files contain no `sorry` or `admit` placeholders. Textual inspection alone is not a fresh kernel verification of the complete import closure.

Preparation of this manuscript included inspecting the pinned definitions, proof bodies, comparator configurations, and dependency metadata. The retrieved source files match their pinned Git blob identifiers, and the `Challenge.lean` and `Solution.lean` SHA-256 hashes match both immutable registry records. No new Lean build or rerun of the external checker was performed for this manuscript. Its mechanical verification claims refer to the dated records; its explanatory claims are grounded in source inspection. The dependent record’s warning about restricted-action scope is addressed by the explicit separation in Section 7.

9. AUTHORSHIP DISCLOSURE AND LIMITATIONS

This manuscript was primarily drafted with GPT 6.1 under the authors’ direction. The submitting author reports understanding some parts of the work. This disclosure does not claim a complete independent human reconstruction or line-by-line human validation of every formal proof. Both pinned `formalization.yaml` files separately record GPT-5 Codex assistance with historical proof engineering. The manuscript model is not inferred to be the historical proof model, and model identity is not verification evidence.

The manuscript and its source are licensed CC BY 4.0. Both Lean projects declare BSD-3-Clause; the included fixed-point source retains its MIT notice. The manuscript license does not relicense those artifacts. The two registry entries list Arthur Freitas Ramos as formalization author and responsible maintainer. The present three-author manuscript byline does not alter that attribution or assert undocumented individual contributions.

The results concern finite strategic-form games, pure potential games, and independently mixed real-payoff games under the stated action hypotheses. They do not establish an equilibrium-selection procedure, uniqueness, computable approximation bounds, infinite-game existence, correlated-equilibrium results, or an economic interpretation beyond the

encoded payoff model. The artifact’s value is its explicit separation of hypotheses, source-traceable proof interfaces, and reusable bridge between native game definitions and proved fixed-point infrastructure.

REFERENCES

- [1] Alexander Bagnall, Samuel Merten, and Gordon Stewart. A library for algorithmic game theory in Ssreflect/Coq. *Journal of Formalized Reasoning*, 10(1):67–95, 2017. DOI: 10.6092/issn.1972-5787/7235.
- [2] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635, Cham, 2021. Springer. DOI: 10.1007/978-3-030-79876-5_37.
- [3] Yuwei Lyu and Kai Li. Formalizing Scarf, Brouwer, and Nash in Lean, July 2026. Version 1; arXiv: 2607.05987v1; DOI: 10.48550/arXiv.2607.05987.
- [4] Math_XMUM. Game theory formalization in Lean. GitHub repository, 2026. Commit 09941e849a81e520cc0cc53220f10f8e5f4768e0; MIT License; <https://github.com/math-xmum/Brouwer/tree/09941e849a81e520cc0cc53220f10f8e5f4768e0>.
- [5] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, May 1996. DOI: 10.1006/game.1996.0044.
- [6] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, September 1951. DOI: 10.2307/1969529.
- [7] John F. Nash, Jr. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, January 1950. DOI: 10.1073/pnas.36.1.48.
- [8] Arthur Freitas Ramos. Dependent finite mixed Nash equilibria in Lean. GitHub repository, 2026. Commit 2f65710af26a24a26660dd1e92d09adf1697a527; native sources BSD-3-Clause, vendored fixed-point sources MIT; <https://github.com/Arthur742Ramos/dependent-nash-equilibrium-lean/tree/2f65710af26a24a26660dd1e92d09adf1697a527>.
- [9] Arthur Freitas Ramos. Dependent finite mixed Nash equilibria in Lean. Palomar Registry, September 2026. PALOMAR-2026-09-07-000014, version 1; <https://palomar-registry.org/entry?id=PALOMAR-2026-09-07-000014&version=1>.
- [10] Arthur Freitas Ramos. Native Lean finite Nash equilibria. GitHub repository, 2026. Commit 6dd83f3b004c0318e52f4c3cb272d909efca321d; native sources BSD-3-Clause, vendored fixed-point sources MIT; <https://github.com/Arthur742Ramos/nash-equilibrium-lean/tree/6dd83f3b004c0318e52f4c3cb272d909efca321d>.
- [11] Arthur Freitas Ramos. Native Lean finite Nash equilibria. Palomar Registry, September 2026. PALOMAR-2026-09-07-000009, version 1; <https://palomar-registry.org/entry?id=PALOMAR-2026-09-07-000009&version=1>.
- [12] Arthur Freitas Ramos, David Barros Hulak, and Ruy Jose Guerra Barretto de Queiroz. Nash equilibria for finite games in Isabelle/HOL. *Archive of Formal Proofs*, May 2026. Formal proof development; https://isa-afp.org/entries/Nash_Equilibrium.html.
- [13] The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, CPP 2020, pages 367–381, New York, NY, USA, 2020. Association for Computing Machinery. DOI: 10.1145/3372885.3373824.