Nash Equilibria for Finite Games in Isabelle/HOL

Arthur Freitas Ramos David Barros Hulak
Ruy J. G. B. de Queiroz

May 8, 2026

Abstract

This development formalizes Nash equilibria for finite strategic-form games, following Nash's equilibrium concept [5, 4]. It gives reusable definitions of profiles, unilateral deviations, best responses, dominant strategies, and pure Nash equilibria; proves existence for finite ordinal potential games [3] and games with dominant strategies; and verifies matching pennies as a finite game with no pure Nash equilibrium. It also formalizes mixed-strategy profiles for finite games, support lemmas for equilibrium strategies, Dirac embeddings of pure profiles, and the existence of a mixed Nash equilibrium using Brouwer's fixed point theorem. Worked examples cover the Prisoner's Dilemma, a coordination game, matching pennies, and rock-paper-scissors. AI assistance was used for proof engineering. The final definitions, statements, and proofs are checked by Isabelle.

Contributions and Scope

This entry gives a reusable Isabelle/HOL formalization of pure and mixed Nash equilibria for finite strategic-form games. The pure-game locale supports a finite player set with player-indexed finite strategy sets and is used to formalize unilateral deviations, best responses, dominant strategies, and ordinal potential games. The mixed-game development proves Nash's finite-game existence theorem by defining an excess-payoff map on a compact convex product of simplices and applying Brouwer's fixed point theorem from HOL-Analysis.

The mixed-game locale is intentionally less general than the pure-game locale: players and pure strategies are represented by finite HOL types, so every player uses the same finite pure-strategy type. This choice makes mixed profiles Cartesian vectors indexed by player/strategy pairs, which gives direct access to the compactness, convexity, continuity, and fixed-point infrastructure needed for the Brouwer proof. The entry also includes support lemmas for mixed equilibria, Dirac embeddings of pure profiles, and checked examples of the Prisoner's Dilemma, a coordination game, matching pennies, and rock-paper-scissors.

Related Work

Le Roux, Martin-Dorel, and Smaus formalized a Nash-equilibrium existence theorem in both Coq and Isabelle for finite-outcome games derived from win/lose games [2]. Bagnall, Merten, and Stewart developed an Ssreflect/Coq library for algorithmic game theory, including pure and mixed Nash equilibria, potential games, smooth games, approximate equilibria, and applications to routing and congestion games [1]. The present entry is narrower in mathematical scope but focuses on an AFP-style Isabelle/HOL development for finite strategic-form games, with a Brouwer-based mixed-equilibrium existence proof and small canonical examples.

Contents

1 Pure Nash Equilibria **3**

2 Existence for Finite Potential Games **5**

3 Dominant Strategies as a Degenerate Existence Result **6**

4 Matching Pennies **6**

5 Mixed Nash Equilibria **7**

6 Prisoner's Dilemma **12**

7 Coordination Game **13**

8 Two-Player Profile Sums **14**

9 Matching Pennies as a Mixed Equilibrium **14**

10 Rock-Paper-Scissors **15**

theory *Nash-Equilibrium*
imports *Main*
begin

1 Pure Nash Equilibria

Nash's equilibrium concept says that a strategy profile is stable when no player can improve by changing only her own strategy. This entry formalizes the pure-strategy version for finite strategic-form games with a common strategy type.

The central locale below fixes a finite set of players, a finite nonempty strategy set for each player, and a payoff function. The player and strategy types are finite; this keeps the profile space finite while still allowing player-indexed strategy restrictions.

The development is intended as a reusable finite-game layer: the basic locale gives pure Nash equilibria and best responses, later locales derive existence from ordinal potentials and dominant strategies, and the companion mixed theory proves the finite mixed-equilibrium theorem using HOL-Analysis.

```
locale finite-game =
  fixes players :: 'p::finite set
    and strategies :: 'p ⇒ 's::finite set
    and payoff :: 'p ⇒ ('p ⇒ 's) ⇒ 'u::preorder
  assumes nonempty-strategies: i ∈ players ⟹ strategies i ≠ {}
begin

definition profiles :: ('p ⇒ 's) set where
  profiles = {s. ∀ i∈players. s i ∈ strategies i}

definition deviation :: ('p ⇒ 's) ⇒ 'p ⇒ 's ⇒ ('p ⇒ 's) where
  deviation s i x = s(i := x)

definition Nash-equilibrium :: ('p ⇒ 's) ⇒ bool where
  Nash-equilibrium s ⟷
    s ∈ profiles ∧
    (∀ i∈players. ∀ x∈strategies i.
      payoff i (deviation s i x) ≤ payoff i s)

definition best-response-to :: ('p ⇒ 's) ⇒ 'p ⇒ 's ⇒ bool where
  best-response-to s i x ⟷
    i ∈ players ∧ x ∈ strategies i ∧
    (∀ y∈strategies i. payoff i (deviation s i y) ≤ payoff i (deviation s i x))

definition dominant-strategy :: 'p ⇒ 's ⇒ bool where
  dominant-strategy i x ⟷
    i ∈ players ∧ x ∈ strategies i ∧
    (∀ s∈profiles. ∀ y∈strategies i.
```

payoff i (*deviation s i y*) $\leq$ *payoff i* (*deviation s i x*))

lemma *profiles-iff*:
s $\in$ *profiles* $\longleftrightarrow$ ($\forall$ *i*$\in$*players*. *s i* $\in$ *strategies i*)
⟨*proof*⟩

lemma *profile-strategy*:
assumes *s* $\in$ *profiles i* $\in$ *players*
shows *s i* $\in$ *strategies i*
⟨*proof*⟩

lemma *finite-profiles* [*simp*, *intro*]: *finite profiles*
⟨*proof*⟩

lemma *profiles-nonempty*: *profiles* $\neq$ {}
⟨*proof*⟩

lemma *deviation-apply* [*simp*]:
deviation s i x j = (*if j* = *i then x else s j*)
⟨*proof*⟩

lemma *deviation-self* [*simp*]: *deviation s i* (*s i*) = *s*
⟨*proof*⟩

lemma *deviation-in-profiles*:
assumes *s* $\in$ *profiles i* $\in$ *players x* $\in$ *strategies i*
shows *deviation s i x* $\in$ *profiles*
⟨*proof*⟩

lemma *Nash-equilibriumI*:
assumes *s* $\in$ *profiles*
and $\bigwedge$*i x*. *i* $\in$ *players* $\Longrightarrow$ *x* $\in$ *strategies i* $\Longrightarrow$
payoff i (*deviation s i x*) $\leq$ *payoff i s*
shows *Nash-equilibrium s*
⟨*proof*⟩

lemma *Nash-equilibrium-profile*:
assumes *Nash-equilibrium s*
shows *s* $\in$ *profiles*
⟨*proof*⟩

lemma *Nash-equilibriumD*:
assumes *Nash-equilibrium s i* $\in$ *players x* $\in$ *strategies i*
shows *payoff i* (*deviation s i x*) $\leq$ *payoff i s*
⟨*proof*⟩

lemma *Nash-equilibrium-iff-best-responses*:
assumes *s* $\in$ *profiles*
shows *Nash-equilibrium s* $\longleftrightarrow$ ($\forall$ *i*$\in$*players*. *best-response-to s i* (*s i*))

⟨*proof*⟩

lemma *dominant-strategy-profile-is-Nash*:
  assumes $s \in$ *profiles*
    and *dominant*: $\bigwedge i.\ i \in$ *players* $\Longrightarrow$ *dominant-strategy* $i\ (s\ i)$
  shows *Nash-equilibrium* s
⟨*proof*⟩

end

2 Existence for Finite Potential Games

Pure Nash equilibria need not exist in arbitrary finite games. A standard positive result is that every finite game with an ordinal potential has a pure Nash equilibrium: choose a profile whose potential is maximal. The definition used here is the one needed for the argument: every strict unilateral payoff improvement strictly increases the potential. Exact and ordinal potential games in the sense of Monderer and Shapley satisfy this assumption.

locale *finite-potential-game* =
  finite-game players strategies payoff
  for *players* :: $'p$::*finite set*
    and *strategies* :: $'p \Rightarrow 's$::*finite set*
    and *payoff* :: $'p \Rightarrow ('p \Rightarrow 's) \Rightarrow 'u$::*linorder* +
  fixes *potential* :: $('p \Rightarrow 's) \Rightarrow 'v$::*linorder*
  assumes *potential-increases*:
    $[\![s \in$ *profiles*; $i \in$ *players*; $x \in$ *strategies* i;
      payoff $i\ s <$ *payoff* i (*deviation* $s\ i\ x$)$]\!]$
    $\Longrightarrow$ *potential* $s <$ *potential* (*deviation* $s\ i\ x$)
begin

lemma *maximal-potential-profile*:
  obtains s **where**
    $s \in$ *profiles*
    $\bigwedge t.\ t \in$ *profiles* $\Longrightarrow$ *potential* $t \le$ *potential* s
⟨*proof*⟩

theorem *exists-Nash-equilibrium*:
  $\exists\, s \in$*profiles*. *Nash-equilibrium* s
⟨*proof*⟩

end

3 Dominant Strategies as a Degenerate Existence Result

Dominant strategies give another elementary source of equilibria. The next locale packages the hypothesis that every player has a distinguished dominant strategy and derives existence by constructing the corresponding profile.

locale *finite-dominant-strategy-game* =
finite-game players strategies payoff
for *players* :: $'p$::*finite set*
and *strategies* :: $'p \Rightarrow 's$::*finite set*
and *payoff* :: $'p \Rightarrow ('p \Rightarrow 's) \Rightarrow 'u$::*preorder* +
fixes *dominant* :: $'p \Rightarrow 's$
assumes *dominant*: $i \in$ *players* $\Longrightarrow$ *dominant-strategy i* (*dominant i*)
begin

definition *dominant-profile* :: $'p \Rightarrow 's$ **where**
dominant-profile i = (*if i* $\in$ *players then dominant i else undefined*)

lemma *dominant-profile-in-profiles*:
dominant-profile $\in$ *profiles*
⟨*proof*⟩

theorem *dominant-profile-is-Nash*:
Nash-equilibrium dominant-profile
⟨*proof*⟩

end

4 Matching Pennies

The following two-player zero-sum game shows why an existence theorem for pure Nash equilibria needs additional hypotheses. The row player wants the two coins to match; the column player wants them to differ. After every pure profile, exactly one player can improve by switching sides.

datatype *penny-player* = *Row* | *Column*

datatype *coin-side* = *Heads* | *Tails*

instantiation *penny-player* :: *finite*
begin

instance
⟨*proof*⟩

end

instantiation *coin-side* :: *finite*
begin

instance
⟨*proof*⟩

end

definition *matching-pennies-payoff* :: *penny-player* ⇒ (*penny-player* ⇒ *coin-side*) ⇒ *int* **where**
  matching-pennies-payoff p s =
    (*case p of*
      Row ⇒ *if s Row* = *s Column then 1 else 0*
    | *Column* ⇒ *if s Row* = *s Column then 0 else 1*)

interpretation *matching-pennies*:
  finite-game UNIV λ-. *UNIV matching-pennies-payoff*
  ⟨*proof*⟩

definition *switch-coin* :: *coin-side* ⇒ *coin-side* **where**
  switch-coin x = (*case x of Heads* ⇒ *Tails* | *Tails* ⇒ *Heads*)

lemma *switch-coin-neq* [*simp*]: *switch-coin x* ≠ *x*
  ⟨*proof*⟩

lemma *coin-side-switch*:
  fixes *x* :: *coin-side*
  obtains *y* **where** *y* ≠ *x*
⟨*proof*⟩

lemma *matching-pennies-no-pure-Nash*:
  ¬ *matching-pennies.Nash-equilibrium s*
⟨*proof*⟩

end
theory *Mixed-Nash-Equilibrium*
  imports
    Nash-Equilibrium
    HOL−Analysis.Cartesian-Euclidean-Space
    HOL−Analysis.Brouwer-Fixpoint
begin

5 Mixed Nash Equilibria

This theory develops the mixed-strategy version of Nash equilibrium for finite games whose players and pure strategies are represented by finite HOL types. A mixed profile is a Cartesian vector indexed by player/strategy pairs.

This locale is deliberately more restrictive than the pure-game locale: every player has the same finite pure-strategy type. In return, the profile space is a finite Cartesian product of real coordinates, so compactness, convexity, continuity, and Brouwer's fixed point theorem can be applied directly.

type-synonym *('p, 's) mixed-profile = real ^ ('p × 's)*

locale *finite-type-game =*
  fixes *payoff :: 'p::finite ⇒ ('p ⇒ 's::finite) ⇒ real*
begin

definition *prob :: ('p, 's) mixed-profile ⇒ 'p ⇒ 's ⇒ real* **where**
  prob m i x = m $ (i, x)

definition *mixed-profiles :: ('p, 's) mixed-profile set* **where**
  mixed-profiles =
    {m ∈ cbox 0 1. ∀ i. (∑ x∈UNIV. prob m i x) = 1}

definition *uniform-mixed-profile :: ('p, 's) mixed-profile* **where**
  uniform-mixed-profile = (χ ix. 1 / real CARD('s))

definition *opponent-weight :: 'p ⇒ ('p, 's) mixed-profile ⇒ ('p ⇒ 's) ⇒ real* **where**
  opponent-weight i m s = (∏ j∈UNIV − {i}. prob m j (s j))

definition *pure-deviation-payoff ::*
    'p ⇒ 's ⇒ ('p, 's) mixed-profile ⇒ real **where**
  pure-deviation-payoff i x m =
    *(∑ s∈{s. s i = x}. opponent-weight i m s * payoff i s)*

definition *mixed-payoff :: 'p ⇒ ('p, 's) mixed-profile ⇒ real* **where**
  mixed-payoff i m =
    *(∑ x∈UNIV. prob m i x * pure-deviation-payoff i x m)*

definition *mixed-Nash-equilibrium :: ('p, 's) mixed-profile ⇒ bool* **where**
  mixed-Nash-equilibrium m ⟷
    m ∈ mixed-profiles ∧
    (∀ i x. pure-deviation-payoff i x m ≤ mixed-payoff i m)

definition *excess :: 'p ⇒ 's ⇒ ('p, 's) mixed-profile ⇒ real* **where**
  excess i x m = max 0 (pure-deviation-payoff i x m − mixed-payoff i m)

definition *excess-sum :: 'p ⇒ ('p, 's) mixed-profile ⇒ real* **where**
  excess-sum i m = (∑ x∈UNIV. excess i x m)

definition *nash-map :: ('p, 's) mixed-profile ⇒ ('p, 's) mixed-profile* **where**
  nash-map m = (χ ix. (prob m (fst ix) (snd ix) + excess (fst ix) (snd ix) m) /
    (1 + excess-sum (fst ix) m))

lemma *prob-nash-map*:
  prob (nash-map m) i x = (prob m i x + excess i x m) / (1 + excess-sum i m)
  ⟨*proof*⟩

lemma *mixed-profiles-prob-nonneg*:
assumes *m* ∈ *mixed-profiles*
shows *0* ≤ *prob m i x*
⟨*proof*⟩

lemma *mixed-profiles-prob-le-one*:
assumes *m* ∈ *mixed-profiles*
shows *prob m i x* ≤ *1*
⟨*proof*⟩

lemma *mixed-profiles-sum-prob*:
assumes *m* ∈ *mixed-profiles*
shows (∑ *x*∈*UNIV*. *prob m i x*) = *1*
⟨*proof*⟩

lemma *prob-uniform-mixed-profile* [*simp*]:
prob uniform-mixed-profile i x = *1* / *real CARD*(*'s*)
⟨*proof*⟩

lemma *uniform-mixed-profile-in-mixed-profiles* [*simp*]:
uniform-mixed-profile ∈ *mixed-profiles*
⟨*proof*⟩

lemma *excess-nonneg* [*simp*]: *0* ≤ *excess i x m*
⟨*proof*⟩

lemma *excess-sum-nonneg* [*simp*]: *0* ≤ *excess-sum i m*
⟨*proof*⟩

lemma *denom-pos* [*simp*]: *0* < *1* + *excess-sum i m*
⟨*proof*⟩

lemma *nash-map-in-mixed-profiles*:
assumes *m*: *m* ∈ *mixed-profiles*
shows *nash-map m* ∈ *mixed-profiles*
⟨*proof*⟩

lemma *continuous-prob*:
continuous-on S (λ*m*. *prob m i x*)
⟨*proof*⟩

lemma *continuous-opponent-weight*:
continuous-on S (λ*m*. *opponent-weight i m s*)
⟨*proof*⟩

lemma *continuous-pure-deviation-payoff*:
continuous-on S (λ*m*. *pure-deviation-payoff i x m*)
⟨*proof*⟩

lemma *continuous-mixed-payoff*:
continuous-on S (λm. *mixed-payoff i m*)
⟨*proof*⟩

lemma *continuous-excess*:
continuous-on S (λm. *excess i x m*)
⟨*proof*⟩

lemma *continuous-excess-sum*:
continuous-on S (λm. *excess-sum i m*)
⟨*proof*⟩

lemma *continuous-nash-map*:
continuous-on mixed-profiles nash-map
⟨*proof*⟩

lemma *mixed-profiles-closed*: *closed mixed-profiles*
⟨*proof*⟩

lemma *mixed-profiles-compact*: *compact mixed-profiles*
⟨*proof*⟩

lemma *mixed-profiles-convex*: *convex mixed-profiles*
⟨*proof*⟩

lemma *mixed-profiles-nonempty*: *mixed-profiles* $\neq$ {}
⟨*proof*⟩

lemma *mixed-Nash-equilibrium-profile*:
assumes *mixed-Nash-equilibrium m*
shows *m* $\in$ *mixed-profiles*
⟨*proof*⟩

lemma *mixed-Nash-equilibriumD*:
assumes *mixed-Nash-equilibrium m*
shows *pure-deviation-payoff i x m* $\leq$ *mixed-payoff i m*
⟨*proof*⟩

lemma *mixed-Nash-support-payoff-eq*:
assumes *ne*: *mixed-Nash-equilibrium m* **and** *px*: *prob m i x* $>$ *0*
shows *pure-deviation-payoff i x m* $=$ *mixed-payoff i m*
⟨*proof*⟩

lemma *mixed-Nash-zero-probability-if-less*:
assumes *ne*: *mixed-Nash-equilibrium m*
and *less*: *pure-deviation-payoff i x m* $<$ *mixed-payoff i m*
shows *prob m i x* $=$ *0*
⟨*proof*⟩

definition *dirac-mixed-profile* :: $('p \Rightarrow 's) \Rightarrow ('p, 's)$ *mixed-profile* **where**
dirac-mixed-profile s = (χ *ix. if snd ix = s (fst ix) then 1 else 0*)

lemma *prob-dirac-mixed-profile* [*simp*]:
prob (*dirac-mixed-profile s*) *i x* = (*if x = s i then 1 else 0*)
⟨*proof*⟩

lemma *dirac-mixed-profile-in-mixed-profiles* [*simp*]:
dirac-mixed-profile s ∈ *mixed-profiles*
⟨*proof*⟩

lemma *opponent-weight-dirac-mixed-profile*:
opponent-weight i (*dirac-mixed-profile s*) *t* =
(*if* ∀ *j. j* ≠ *i* ⟶ *t j* = *s j then 1 else 0*)
⟨*proof*⟩

lemma *pure-deviation-payoff-dirac-mixed-profile*:
pure-deviation-payoff i x (*dirac-mixed-profile s*) = *payoff i* (*s*(*i* := *x*))
⟨*proof*⟩

lemma *mixed-payoff-dirac-mixed-profile*:
mixed-payoff i (*dirac-mixed-profile s*) = *payoff i s*
⟨*proof*⟩

lemma *dirac-mixed-Nash-equilibrium*:
assumes ⋀*i x. payoff i* (*s*(*i* := *x*)) ≤ *payoff i s*
shows *mixed-Nash-equilibrium* (*dirac-mixed-profile s*)
⟨*proof*⟩

lemma *fixed-point-imp-excess-zero*:
assumes *m*: *m* ∈ *mixed-profiles* **and** *fp*: *nash-map m* = *m*
shows *excess i x m* = *0*
⟨*proof*⟩

theorem *exists-mixed-Nash-equilibrium*:
∃ *m*∈*mixed-profiles. mixed-Nash-equilibrium m*
⟨*proof*⟩

end

end
theory *Nash-Equilibrium-Examples*
imports *Mixed-Nash-Equilibrium*
begin

lemma *UNIV-penny-player*: (*UNIV* :: *penny-player set*) = {*Row*, *Column*}
⟨*proof*⟩

lemma *UNIV-coin-side* [*simp*]: (*UNIV* :: *coin-side set*) = {*Heads*, *Tails*}
⟨*proof*⟩

lemma *players-except-Row* [*simp*]: (*UNIV* :: *penny-player set*) − {*Row*} = {*Column*}
⟨*proof*⟩

lemma *players-except-Column* [*simp*]: (*UNIV* :: *penny-player set*) − {*Column*} = {*Row*}
⟨*proof*⟩

lemma *player-insert-except-Row* [*simp*]: {*Row*, *Column*} − {*Row*} = {*Column*}
⟨*proof*⟩

lemma *player-insert-except-Column* [*simp*]: {*Row*, *Column*} − {*Column*} = {*Row*}
⟨*proof*⟩

6 Prisoner's Dilemma

The Prisoner's Dilemma gives a small example using the dominant-strategy existence result. Defection is dominant for both players, hence the all-defect profile is a pure Nash equilibrium.

datatype *prisoner* = *Prisoner1* | *Prisoner2*

datatype *prisoner-move* = *Cooperate* | *Defect*

instantiation *prisoner* :: *finite*
begin

instance
⟨*proof*⟩

end

instantiation *prisoner-move* :: *finite*
begin

instance
⟨*proof*⟩

end

fun *other-prisoner* :: *prisoner* ⇒ *prisoner* **where**
other-prisoner Prisoner1 = *Prisoner2*
| *other-prisoner Prisoner2* = *Prisoner1*

definition *prisoners-dilemma-payoff* :: *prisoner* ⇒ (*prisoner* ⇒ *prisoner-move*) ⇒ *int* **where**
prisoners-dilemma-payoff p s =

(case (s p, s (other-prisoner p)) of
(Cooperate, Cooperate) ⇒ 3
| (Defect, Cooperate) ⇒ 5
| (Cooperate, Defect) ⇒ 0
| (Defect, Defect) ⇒ 1)

interpretation *prisoners-dilemma*:
finite-game UNIV λ-. UNIV prisoners-dilemma-payoff
⟨*proof*⟩

interpretation *prisoners-dilemma-dominant*:
finite-dominant-strategy-game UNIV λ-. UNIV prisoners-dilemma-payoff λ-. Defect
⟨*proof*⟩

lemma *prisoners-dilemma-defect-defect-Nash*:
prisoners-dilemma.Nash-equilibrium (λ-. Defect)
⟨*proof*⟩

7 Coordination Game

A two-player coordination game has two pure equilibria. Both players receive payoff one when their choices agree and zero otherwise.

datatype *coordination-choice = Choice-A | Choice-B*

instantiation *coordination-choice* :: *finite*
begin

instance
⟨*proof*⟩

end

definition *coordination-payoff* ::
penny-player ⇒ (penny-player ⇒ coordination-choice) ⇒ int **where**
coordination-payoff p s = (if s Row = s Column then 1 else 0)

interpretation *coordination*:
finite-game UNIV λ-. UNIV coordination-payoff
⟨*proof*⟩

lemma *coordination-A-A-Nash*:
coordination.Nash-equilibrium (λ-. Choice-A)
⟨*proof*⟩

lemma *coordination-B-B-Nash*:
coordination.Nash-equilibrium (λ-. Choice-B)
⟨*proof*⟩

8 Two-Player Profile Sums

definition *two-player-profile* :: *'a* ⇒ *'a* ⇒ *penny-player* ⇒ *'a* **where**
two-player-profile r c p = (*case p of Row* ⇒ *r* | *Column* ⇒ *c*)

lemma *two-player-profile-simps* [*simp*]:
two-player-profile r c Row = *r*
two-player-profile r c Column = *c*
⟨*proof*⟩

lemma *sum-profiles-fixed-Row*:
fixes *F* :: (*penny-player* ⇒ *'a*::*finite*) ⇒ *'b*::*comm-monoid-add*
shows ($\sum$ *s*∈{*s. s Row* = *x*}. *F s*) =
($\sum$ *y*∈*UNIV*. *F* (*two-player-profile x y*))
⟨*proof*⟩

lemma *sum-profiles-fixed-Column*:
fixes *F* :: (*penny-player* ⇒ *'a*::*finite*) ⇒ *'b*::*comm-monoid-add*
shows ($\sum$ *s*∈{*s. s Column* = *x*}. *F s*) =
($\sum$ *y*∈*UNIV*. *F* (*two-player-profile y x*))
⟨*proof*⟩

9 Matching Pennies as a Mixed Equilibrium

definition *matching-pennies-payoff-real* ::
penny-player ⇒ (*penny-player* ⇒ *coin-side*) ⇒ *real* **where**
matching-pennies-payoff-real p s = *real-of-int* (*matching-pennies-payoff p s*)

interpretation *matching-pennies-mixed*:
finite-type-game matching-pennies-payoff-real ⟨*proof*⟩

lemma *matching-pennies-pure-deviation-Row*:
matching-pennies-mixed.pure-deviation-payoff Row x m =
($\sum$ *y*∈*UNIV*.
matching-pennies-mixed.prob m Column y ∗
matching-pennies-payoff-real Row (*two-player-profile x y*))
⟨*proof*⟩

lemma *matching-pennies-pure-deviation-Column*:
matching-pennies-mixed.pure-deviation-payoff Column x m =
($\sum$ *y*∈*UNIV*.
matching-pennies-mixed.prob m Row y ∗
matching-pennies-payoff-real Column (*two-player-profile y x*))
⟨*proof*⟩

lemma *matching-pennies-uniform-pure-deviation-payoff*:
matching-pennies-mixed.pure-deviation-payoff i x
matching-pennies-mixed.uniform-mixed-profile = *1* / *2*
⟨*proof*⟩

lemma *matching-pennies-uniform-mixed-payoff*:
matching-pennies-mixed.mixed-payoff i matching-pennies-mixed.uniform-mixed-profile = *1* / *2*
⟨*proof*⟩

lemma *matching-pennies-uniform-mixed-Nash*:
matching-pennies-mixed.mixed-Nash-equilibrium
matching-pennies-mixed.uniform-mixed-profile
⟨*proof*⟩

10 Rock-Paper-Scissors

datatype *rps* = *Rock* | *Paper* | *Scissors*

instantiation *rps* :: *finite*
begin

instance
⟨*proof*⟩

end

lemma *UNIV-rps* [*simp*]: (*UNIV* :: *rps set*) = {*Rock*, *Paper*, *Scissors*}
⟨*proof*⟩

fun *beats* :: *rps* ⇒ *rps* ⇒ *bool* **where**
beats Rock Scissors = *True*
| *beats Paper Rock* = *True*
| *beats Scissors Paper* = *True*
| *beats* - - = *False*

definition *rps-payoff* :: *penny-player* ⇒ (*penny-player* ⇒ *rps*) ⇒ *real* **where**
rps-payoff p s =
(*if s Row* = *s Column then 0*
else if beats (*s Row*) (*s Column*)
then if p = *Row then 1 else* −*1*
else if p = *Row then* −*1 else 1*)

interpretation *rps-mixed*:
finite-type-game rps-payoff ⟨*proof*⟩

lemma *rps-pure-deviation-Row*:
rps-mixed.pure-deviation-payoff Row x m =
($\sum$ *y*∈*UNIV*. *rps-mixed.prob m Column y* ∗ *rps-payoff Row* (*two-player-profile x y*))
⟨*proof*⟩

lemma *rps-pure-deviation-Column*:

rps-mixed.pure-deviation-payoff Column x m =
($\sum$ *y*∈*UNIV*. *rps-mixed.prob m Row y* ∗ *rps-payoff Column* (*two-player-profile y x*))
⟨*proof*⟩

lemma *rps-uniform-pure-deviation-payoff*:
rps-mixed.pure-deviation-payoff i x rps-mixed.uniform-mixed-profile = *0*
⟨*proof*⟩

lemma *rps-uniform-mixed-payoff*:
rps-mixed.mixed-payoff i rps-mixed.uniform-mixed-profile = *0*
⟨*proof*⟩

lemma *rps-uniform-mixed-Nash*:
rps-mixed.mixed-Nash-equilibrium rps-mixed.uniform-mixed-profile
⟨*proof*⟩

end

References

[1] A. Bagnall, S. Merten, and G. Stewart. A library for algorithmic game theory in ssreflect/coq. *Journal of Formalized Reasoning*, 10(1):67–95, 2017. DOI: `https://doi.org/10.6092/issn.1972-5787/7235`.

[2] S. Le Roux, E. Martin-Dorel, and J.-G. Smaus. An existence theorem of nash equilibrium in coq and isabelle. In *Proceedings Eighth International Symposium on Games, Automata, Logics and Formal Verification*, volume 256 of *Electronic Proceedings in Theoretical Computer Science*, pages 46–60, 2017. DOI: `https://doi.org/10.4204/EPTCS.256.4`.

[3] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996. DOI: `https://doi.org/10.1006/game.1996.0044`.

[4] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951. DOI: `https://doi.org/10.2307/1969529`.

[5] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. DOI: `https://doi.org/10.1073/pnas.36.1.48`.