Raisonnements dans les jeux séquentiels infinis

Pierre Lescanne (ENS de Lyon)

12 novembre 2008

Three words: sequential, infinite, reasoning

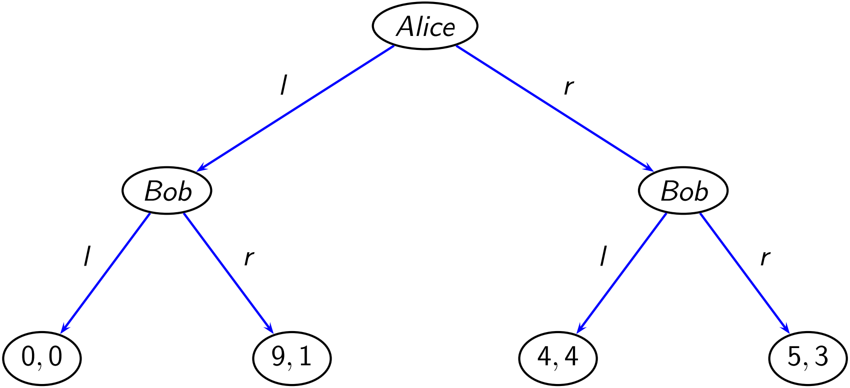
Mechanizing the reasoning

Finite sequential games in COQ

Illogic conflict of escalation revisited

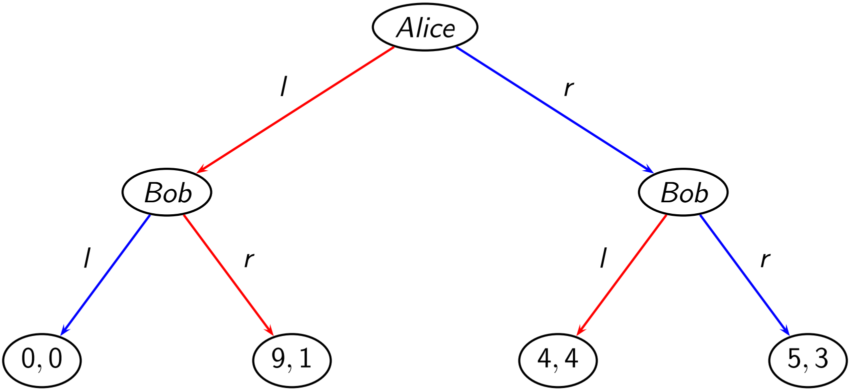
What is a sequential game?

A sequential game is described by a labeled tree



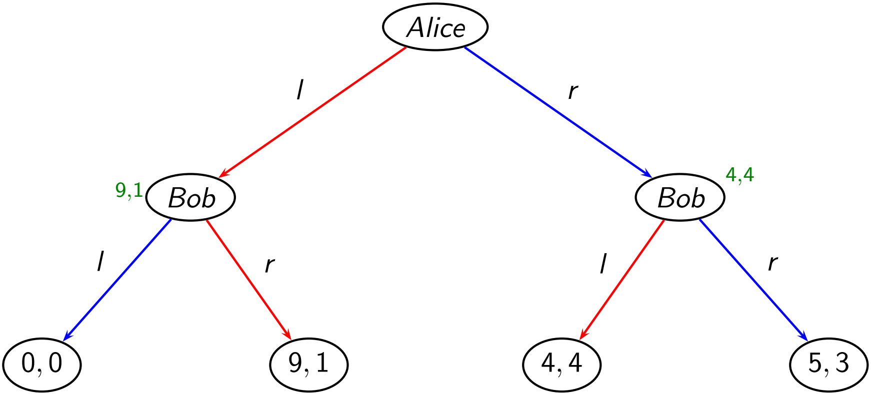
What is a sequential game?

A Nash equilibrium is a situation where if an agent changes alone his action he will get a utility which is not better.



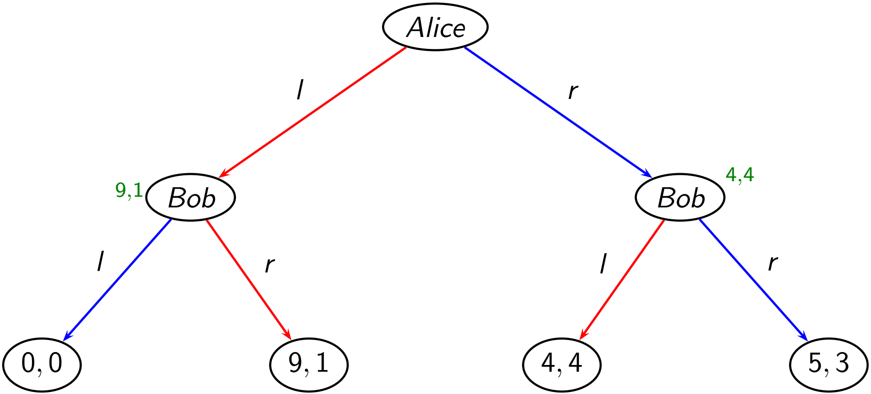
What is a sequential game?

A Nash equilibrium is a situation where if an agent changes alone his action he will get a utility which is not better.



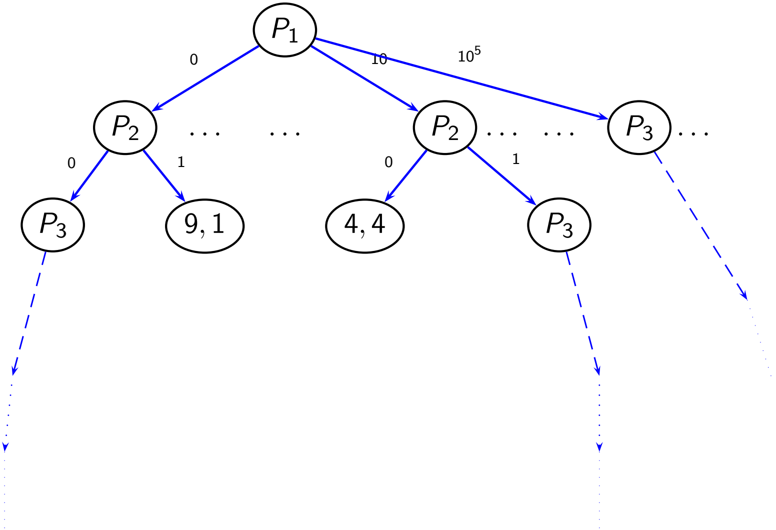
What is a sequential game?

A Nash equilibrium is a situation where if an agent changes alone his action he will get a utility which is not better.



This method for computing a Nash equilibrium is called backward induction.

A sequential game can be infinite



What is an infinite sequential game?

What does "infinite" mean?

- The length of the game?
- The number of players?
- The number of choices of actions a player can perform at each decision node?

Definition [of extensive game] allows terminal histories to be infinitely long. Thus we can use the model of an extensive game to study situations in which the participants do not consider any particular fixed horizon when making decisions. If the length of the longest terminal history is in fact finite, we say that the game has a **finite horizon***.*

Even a game with a finite horizon may have infinitely many terminal histories, because some player has infinitely many actions after some history. If a game has a finite horizon and finitely many terminal histories we say it is **finite***.*

Martin Osborne,
An Introduction to Game Theory
Oxford U. Press, (2004), p. 137

Definition [of extensive game] allows terminal histories to be infinitely long. Thus we can use the model of an extensive game to study situations in which the participants do not consider any particular fixed horizon when making decisions. If the length of the longest terminal history is in fact finite, we say that the game has a **finite horizon***.*

Even a game with a finite horizon may have infinitely many terminal histories, because some player has infinitely many actions after some history. If a game has a finite horizon and finitely many terminal histories we say it is **finite***.*

Martin Osborne,
An Introduction to Game Theory
Oxford U. Press, (2004), p. 137

A typical example: the illogic escalation

In 1971, Martin Shubik described an infinite game, he calls

The Dollar Auction Game,

in which players bid forever.

A typical example: the illogic escalation

In 1971, Martin Shubik described an infinite game, he calls

The Dollar Auction Game,

in which players bid forever.

- Does this game has an infinite history?

A typical example: the illogic escalation

In 1971, Martin Shubik described an infinite game, he calls
The Dollar Auction Game,
in which players bid forever.

- Does this game has an infinite history?
- If not, this contradicts König lemma.

A typical example: the illogic escalation

In 1971, Martin Shubik described an infinite game, he calls
The Dollar Auction Game,
in which players bid forever.

- Does this game has an infinite history?
- If not, this contradicts König lemma.

We should have a way to discriminate among infinite paths to keep those that are actual histories.

How do agents reason?

The question is:

"How do agents reason in infinite sequential games?"

How do agents reason?

The question is:

"How do agents reason in infinite sequential games?"

I consider agents as idealized entities with a full reasoning power and I wonder what a full reasoning could be.

My solution is to formalize the reasoning in a proof assistant which pushes me to analyze the reasoning in all its details:

- to highlight reasoning done unconsciously by humans.
- to examine the reasonings involved in infinite games,
  maybe explaining "paradoxes".

How do agents reason?

The question is:

"How do agents reason in infinite sequential games?"

I consider agents as idealized entities with a full reasoning power and I wonder what a full reasoning could be.

My solution is to formalize the reasoning in a proof assistant which pushes me to analyze the reasoning in all its details:

- to highlight reasoning done unconsciously by humans.
- to examine the reasonings involved in infinite games,
  maybe explaining "paradoxes".

  Like the infinite escalation.

How do agents reason?

The question is:

"How do agents reason in infinite sequential games?"

I consider agents as idealized entities with a full reasoning power and I wonder what a full reasoning could be.

My solution is to formalize the reasoning in a proof assistant which pushes me to analyze the reasoning in all its details:

- to highlight reasoning done unconsciously by humans.
- to examine the reasonings involved in infinite games,
  maybe explaining "paradoxes".
  Like the infinite escalation.

For instance, from my experiment I can claim that

- agents do not use the middle excluded $p \vee \neg p$ and
- agents use only constructive reasoning.

Three words: sequential, infinite, reasoning

Mechanizing the reasoning

Finite sequential games in Coq

Illogic conflict of escalation revisited

Mechanical proof a revolution

We are living a revolution in mathematics.
Proofs are now mechanically checked by computer.

Mechanical proof a revolution

We are living a revolution in mathematics.
Proofs are now mechanically checked by computer.

- Fundamental theorem of Algebra (2000)
- Prime number theorem (2005)
- Jordan curve theorem (2005)
- Four color theorem (2005)
- Feit-Thompson theorem (under development)
- Kepler conjecture (under development)

Mechanical proof a revolution

We are living a revolution in mathematics.
Proofs are now mechanically checked by computer.

- Fundamental theorem of Algebra (2000)
- Prime number theorem (2005)
- Jordan curve theorem (2005)
- Four color theorem (2005)
- Feit-Thompson theorem (under development)
- Kepler conjecture (under development)

Thomas Hales (who proved Kepler conjecture) says that such a collection of proofs would be akin to ''the sequencing of the mathematical genome''.

More

For those who want to know more read

Gilles Dowek, Les métamorphoses du calcul, Le Pommier (2007).
Grand Prix de philosophie de l'Académie française.

or
Proof by computer: Harnessing the power of computers to verify mathematical proofs
on `http://www.physorg.com/news145200777.html`

Why a mechanical proof ?

Knuth quoting George Forsythe (the founder of the CS department at Stanford) said:

> *People have said you don't understand something until you've taught it in a class.*
>
> *The truth is you don't really understand something until you've taught it to a computer, until you've been able to program it.*

Why a mechanical proof ?

Knuth quoting George Forsythe (the founder of the CS department at Stanford) said:

> *People have said you don't understand something until you've taught it in a class.*
> *The truth is you don't really understand something until you've been taught it to a proof assistant, until you've code it into Coq.*

Constructive reasoning

In constructive logic:

- One has a proof of $A \Rightarrow B$ if one can build a proof of B from a proof of A.

Constructive reasoning

In constructive logic:

- One has a proof of $A \Rightarrow B$ if one can build a proof of B from a proof of A.
- The existence of an object holds, only if one can construct it,

Constructive reasoning

In constructive logic:

- One has a proof of $A \Rightarrow B$ if one can build a proof of B from a proof of A.
- The existence of an object holds, only if one can construct it,
- The Curry-Howard correspondence plays a key role,

  i.e. proofs are programs.

Inductive and coinductive reasoning

Inductive definition is for describing and formalizing finite objects.

Inductive and coinductive reasoning

Inductive definition is for describing and formalizing
finite objects.

Coinductive definition is for describing and formalizing
infinite objects.

Inductive and coinductive reasoning

Inductive definition is for describing and formalizing
finite objects.

Coinductive definition is for describing and formalizing
infinite objects.

Both induction and coinduction are fixed points, but

- an inductive definition is a least fixed point and
- a coinductive induction is a greatest fixed point.

Three words: sequential, infinite, reasoning

Mechanizing the reasoning

Finite sequential games in Coq

Illogic conflict of escalation revisited

Finite sequential games as inductive objects

A finite sequential games is described by induction from its subgames.

Without loss of generality, I restrict to binary sequential games.

A *binary finite sequential game* is

- either a **node**, assigned to a **player**, with **two subgames**,
- or a **leaf**.

Finite sequential games as inductive objects

A finite sequential games is described by induction from its subgames.

Without loss of generality, I restrict to binary sequential games.

A *binary finite sequential game* is

- either a **node**, assigned to a **player**, with **two subgames**,
- or a **leaf**.

Inductive *FinGame : Set :=*
| *gLeaf : Utility_fun* $\rightarrow$ *FinGame*
| *gNode : Agent* $\rightarrow$ *FinGame* $\rightarrow$ *FinGame* $\rightarrow$ *FinGame.*

Utility and utility functions

Utility is given.

Utility_fun is a function which associates a utility with an agent:

Definition $Utility\_fun := Agent \rightarrow Utility.$

Induction principle on finite games

- If P holds for finite games that are just leaves,

Induction principle on finite games

- If P holds for finite games that are just leaves,
- If P holds for a finite game f_1 and
  if P holds for a finite game f_2,
  then it holds for the "composition" of f_1 and f_2.

Induction principle on finite games

- If P holds for finite games that are just leaves,
- If P holds for a finite game f_1 and
  if P holds for a finite game f_2,
  then it holds for the "composition" of f_1 and f_2.

Then P holds for all finite game.

Induction principle on finite games

- If P holds for finite games that are just leaves,
- If P holds for a finite game f_1 and
  if P holds for a finite game f_2,
  then it holds for the "composition" of f_1 and f_2.

Then P holds for all finite game.

$$\frac{\forall u : \mathit{Utility\_fun}, P(\mathit{gLeaf}\ \ u) \qquad [\forall f0 : \mathit{FinGame}, P\ f0 \wedge \forall f1 : \mathit{FinGame}, P\ f1] \to \forall a : \mathit{Agent}, P\ (\mathit{gNode}\ a\ f0\ f1)}{\forall f : \mathit{FinGame}, P\ f}$$

Induction principle on finite games

- If P holds for finite games that are just leaves,
- If P holds for a finite game f_1 and
  if P holds for a finite game f_2,
  then it holds for the "composition" of f_1 and f_2.

Then P holds for all finite game.

$$\frac{\forall u : \mathit{Utility\_fun}, P(\mathit{gLeaf}\ \ u) \qquad [\forall f0 : \mathit{FinGame}, P\ f0 \wedge \forall f1 : \mathit{FinGame}, P\ f1] \rightarrow \forall a : \mathit{Agent}, P\ (\mathit{gNode}\ a\ f0\ f1)}{\forall f : \mathit{FinGame}, P\ f}$$

Induction principle on finite games

- If P holds for finite games that are just leaves,
- If P holds for a finite game f_1 and
  if P holds for a finite game f_2,
  then it holds for the "composition" of f_1 and f_2.

Then P holds for all finite game.

$$\frac{\forall u : Utility\_fun, P(gLeaf\ \ u) \qquad [\forall f0 : FinGame, P\ f0 \wedge \forall f1 : FinGame, P\ f1] \rightarrow \forall a : Agent, P\ (gNode\ a\ f0\ f1)}{\forall f : FinGame, P\ \ f}$$

Induction principle on finite games

- If P holds for finite games that are just leaves,
- If P holds for a finite game f_1 and
  if P holds for a finite game f_2,
  then it holds for the "composition" of f_1 and f_2.

Then P holds for all finite game.

$$\frac{\forall u : Utility\_fun, P(gLeaf\ \ u) \qquad [\forall f0 : FinGame, P\ f0 \land \forall f1 : FinGame, P\ f1] \to \forall a : Agent, P\ (gNode\ a\ f0\ f1)}{\forall f\ : FinGame, P\ \ f}$$

A “finite” strategy is also an inductive

Inductive *FinStrategy* : *Set* :=
| *sLeaf* : *Utility_fun* → *FinStrategy*
| *sNode* : *Agent* → *Choice* → *FinStrategy* → *FinStrategy* → *FinStrategy*.

From finite strategy to utility function

```
Fixpoint f2u (s:FinStrategy) : Utility_fun :=
match s with
| (sLeaf uf) ⇒ uf
| (sNode a left sl sr) ⇒ (f2u sl)
| (sNode a right sl sr) ⇒ (f2u sr)
end.
```

a-convertibility

$s \twoheadleftarrow a \twoheadrightarrow s'$ is a relation between strategies.

s is a-convertible to s'

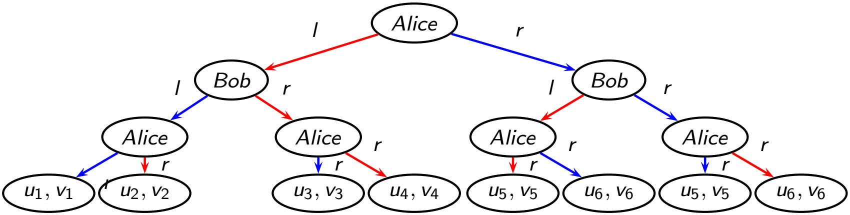
if one goes from s to s' by changing the choices of agent a.

a-convertibility

$s \twoheadleftarrow a \twoheadrightarrow s'$ is a relation between strategies.

s is a-convertible to s'

if one goes from s to s' by changing the choices of agent a.

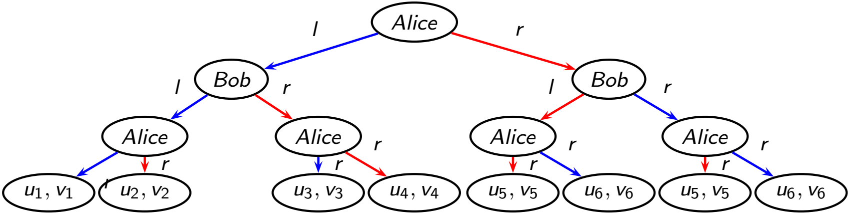


a-convertibility

$s \twoheadleftarrow a \twoheadrightarrow s'$ is a relation between strategies.

s is a-convertible to s'

if one goes from s to s' by changing the choices of agent a.



a-convertibility

$s \twoheadleftarrow a \twoheadrightarrow s'$ is a relation between strategies.

s is a-convertible to s'

if one goes from s to s' by changing the choices of agent a.

$\twoheadleftarrow a \twoheadrightarrow$ is defined as an inductive.

- *sLeaf uf* $\twoheadleftarrow a \twoheadrightarrow$ *sLeaf uf*.
- (*sNode a c s1 s2*) $\twoheadleftarrow a \twoheadrightarrow$ (*sNode a c' s1' s2'*)

  if *s1* $\twoheadleftarrow a \twoheadrightarrow$ *s1'* and *s2* $\twoheadleftarrow a \twoheadrightarrow$ *s2'*.

  a is the same

  c and c' do not have to be the same,
- (*sNode a' c s1 s2*) $\twoheadleftarrow a \twoheadrightarrow$ (*sNode a' c s1' s2'*)

  if *s1* $\twoheadleftarrow a \twoheadrightarrow$ *s1'* and *s2* $\twoheadleftarrow a \twoheadrightarrow$ *s2'*.

  a and a' do not both have to be the same,

  c has to be the same.

a-convertibility

I proved in Coq that the $\twoheadleftarrow a \twoheadrightarrow$ is an equivalence relation.

a-convertibility

I proved in COQ that the $\twoheadleftarrow a \twoheadrightarrow$ is an equivalence relation.

We are now equipped to define the predicate Nash equilibrium on finite strategy.

Nash equilibrium

Inductive *FinNashEq*: *FinStrategy* $\rightarrow$ *Prop* :=
| *NE* : $\forall$ (*s*:*FinStrategy*),
 ($\forall$ (*a*:*Agent*) (*s'*:*FinStrategy*), $s \twoheadleftarrow a \twoheadrightarrow s' \rightarrow$ (*f2u s' a* $\preceq$ *f2u s a*)) $\rightarrow$
 FinNashEq s.

Backward induction

On finite strategies.

Inductive BI: $FinStrategy \rightarrow Prop :=$

$\mid$ $BILeaf$: $\forall$ uf:$Utility\_fun$, BI $(sLeaf\ uf)$

$\mid$ $BINode\_left$: $\forall$ $(a{:}Agent)$ $(sl\ sr{:}\ FinStrategy)$,
$BI\ sl \rightarrow BI\ sr \rightarrow (f2u\ sr\ a \preceq f2u\ sl\ a) \rightarrow BI\ (sNode\ a\ left\ sl\ sr)$

$\mid$ $BINode\_right$: $\forall$ $(a{:}Agent)$ $(sl\ sr{:}\ FinStrategy)$,
$BI\ sl \rightarrow BI\ sr \rightarrow (f2u\ sl\ a \preceq f2u\ sr\ a) \rightarrow BI\ (sNode\ a\ right\ sl\ sr)$.

Backward induction

On finite strategies.

Inductive BI: $FinStrategy \rightarrow Prop :=$

| $BILeaf$: $\forall\ uf{:}Utility\_fun,\ BI\ (sLeaf\ \ uf)$

| $BINode\_left$: $\forall\ (a{:}Agent)\ (sl\ sr{:}\ FinStrategy),$
$BI\ sl \rightarrow BI\ sr \rightarrow (f2u\ sr\ a \preceq f2u\ sl\ a) \rightarrow BI\ (sNode\ a\ left\ sl\ sr)$

| $BINode\_right$: $\forall\ (a{:}Agent)\ (sl\ sr{:}\ FinStrategy),$
$BI\ sl \rightarrow BI\ sr \rightarrow (f2u\ sl\ a \preceq f2u\ sr\ a) \rightarrow BI\ (sNode\ a\ right\ sl\ sr).$

If s is BI then s is a Nash equilibrium.

Backward induction

On finite strategies.

Inductive BI: $FinStrategy \rightarrow Prop :=$

| $BILeaf$: $\forall$ uf:$Utility\_fun$, BI ($sLeaf$ uf)

| $BINode\_left$: $\forall$ (a:$Agent$) (sl sr: $FinStrategy$),
$BI\ sl \rightarrow BI\ sr \rightarrow (f2u\ sr\ a \preceq f2u\ sl\ a) \rightarrow BI\ (sNode\ a\ left\ sl\ sr)$

| $BINode\_right$: $\forall$ (a:$Agent$) (sl sr: $FinStrategy$),
$BI\ sl \rightarrow BI\ sr \rightarrow (f2u\ sl\ a \preceq f2u\ sr\ a) \rightarrow BI\ (sNode\ a\ right\ sl\ sr)$.

If s is BI then s is a Nash equilibrium.

Theorem $BI\_is\_FinNashEq$: $\forall$ s, $BI\ s \rightarrow FinNashEq\ s$.

The coinductive *InfGame*

CoInductive *InfGame* : *Set* :=
| *igNode* : *Agent* → *InfGame* → *FinGame* → *InfGame*.

The coinductive *InfGame*

CoInductive *InfGame* : *Set* :=
| *igNode* : *Agent* → *InfGame* → *FinGame* → *InfGame*.

On the left an infinite game, on the right a finite game.

The coinductive *InfGame*

CoInductive *InfGame* : *Set* :=
| *igNode* : *Agent* → *InfGame* → *FinGame* → *InfGame*.

On the left an infinite game, on the right a finite game.

The concept of infinite strategy is also defined as a coinductive:

CoInductive *InfStrategy* : *Set* :=
| *iNode* : *Agent* → *Choice* → *InfStrategy* → *FinStrategy* → *InfStrategy*.

Finite horizon, limited horizon and backward induction

In classical game theory, backward induction relies on finite horizon, i.e. all paths are finite.

Finite horizon, limited horizon and backward induction

In classical game theory, backward induction relies on finite horizon, i.e. all paths are finite.

I substitute the notion of strategy with limited horizon,
i.e., the "path" of the strategy goes eventually to the right.

Finite horizon, limited horizon and backward induction

In classical game theory, backward induction relies on finite horizon, i.e. all paths are finite.

I substitute the notion of strategy with limited horizon,

i.e., the “path” of the strategy goes eventually to the right.

hence is finite

From infinite strategy to utility function

The utility function *i2u* is no more a function, but a relation, since it is no more total.

From infinite strategy to utility function

The utility function *i2u* is no more a function, but a relation, since it is no more total.

It returns a value only on strategies which go eventually to the right.

The predicate eventually to the right

I introduce a predicate on strategies, called eventually to the right and written *EvtRight*.

The predicate eventually to the right

I introduce a predicate on strategies, called eventually to the right and written *EvtRight*.

On strategies that go eventually to the right, one gets

existence and uniqueness of the utility
associated with each agent.

The predicate eventually to the right

I introduce a predicate on strategies, called eventually to the right and written *EvtRight*.

On strategies that go eventually to the right, one gets
existence and uniqueness of the utility
associated with each agent.

Lemma *Existence_i2u*: ∀ (*a*:*Agent*) (*s*:*InfStrategy*),
EvtRight s → ∃ *u*:*Utility*, *i2u a u s*.

Lemma *Uniqueness_i2u*: ∀ (*a*:*Agent*) (*u v*:*Utility*) (*s*:*InfStrategy*),
EvtRight s → *i2u a u s* → *i2u a v s* → *u*=*v*.

Nash equilibria

Inductive *InfNashEq*: *InfStrategy* $\to$ *Prop* :=
| *INE* : $\forall$ (*s*: *InfStrategy*),
EvtRight s $\to$
($\forall$ (*a*:*Agent*) (*s'*:*InfStrategy*) (*u u'*: *Utility*),
EvtRight s' $\to$ (*s'*$\leftarrow$ *a* $\to$ *s*) $\to$ (*i2u a u' s'*) $\to$ (*i2u a u s*) $\to$ (*u'* $\preceq$ *u*)) $\to$
InfNashEq s.

Nash equilibria

$$
\begin{array}{l}
\textbf{Inductive } \mathit{InfNashEq}\text{: } \mathit{InfStrategy} \to \mathit{Prop} := \\
\mid \mathit{INE} : \forall\ (s\text{: } \mathit{InfStrategy}), \\
\quad \mathit{EvtRight}\ s \to \\
\quad (\forall\ (a\text{:}\mathit{Agent})\ (s'\text{:}\mathit{InfStrategy})\ (u\ u'\text{: } \mathit{Utility}), \\
\quad \mathit{EvtRight}\ s' \to (s' \leftarrow a \to\ s) \to (\mathit{i2u}\ a\ u'\ s') \to (\mathit{i2u}\ a\ u\ s) \to (u' \preceq u)) \to \\
\qquad \mathit{InfNashEq}\ s.
\end{array}
$$

Sub Game Perfect Equilibria

CoInductive *SGPE*: *InfStrategy* $\rightarrow$ *Prop* :=

| *SGPEnode_left*: $\forall$ (*a*:*Agent*)(*u*:*Utility*) (*sl*: *InfStrategy*) (*sr*: *FinStrategy*),
AlwEvtRight sl $\rightarrow$ *SGPE sl* $\rightarrow$ *BI sr* $\rightarrow$ *i2u a u sl* $\rightarrow$ (*f2u sr a* $\preceq$ *u*) $\rightarrow$
SGPE (*iNode a left sl sr*)

| *SGPEnode_right*: $\forall$ (*a*:*Agent*) (*u*:*Utility*) (*sl*: *InfStrategy*) (*sr*: *FinStrategy*),
AlwEvtRight sl $\rightarrow$ *SGPE sl* $\rightarrow$ *BI sr* $\rightarrow$ *i2u a u sl* $\rightarrow$ (*u* $\preceq$ *f2u sr a*) $\rightarrow$
SGPE (*iNode a right sl sr*).

Sub Game Perfect Equilibria

CoInductive *SGPE*: *InfStrategy* $\to$ *Prop* :=

| *SGPEnode_left*: $\forall$ (*a*:*Agent*)(*u*:*Utility*) (*sl*: *InfStrategy*) (*sr*: *FinStrategy*),
AlwEvtRight sl $\to$ *SGPE sl* $\to$ *BI sr* $\to$ *i2u a u sl* $\to$ (*f2u sr a* $\preceq$ *u*) $\to$
SGPE (*iNode a left sl sr*)

| *SGPEnode_right*: $\forall$ (*a*:*Agent*) (*u*:*Utility*) (*sl*: *InfStrategy*) (*sr*: *FinStrategy*),
AlwEvtRight sl $\to$ *SGPE sl* $\to$ *BI sr* $\to$ *i2u a u sl* $\to$ (*u* $\preceq$ *f2u sr a*) $\to$
SGPE (*iNode a right sl sr*).

Sub Game Perfect Equilibria are Nash equilibria

Theorem *SGPE_is_InfNashEq* :
$\forall$ *s*:*InfStrategy*, *EvtMaxU s* $\rightarrow$ *SGPE s* $\rightarrow$ *InfNashEq s*.

Sub Game Perfect Equilibria are Nash equilibria

Theorem *SGPE_is_InfNashEq* :
$\forall$ *s*:*InfStrategy*, *EvtMaxU s* $\rightarrow$ *SGPE s* $\rightarrow$ *InfNashEq s*.

Three words: sequential, infinite, reasoning

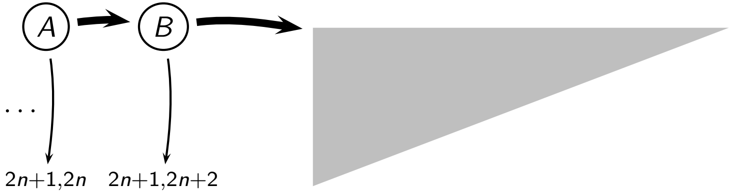
Mechanizing the reasoning

Finite sequential games in Coq

Illogic conflict of escalation revisited

Never give up is not a Nash equilibrium

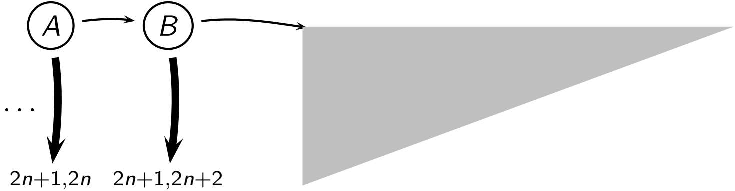
In Shubik's game, we can proof that the strategy never give up



is not a Nash equilibrium.

Always give up is a SubGame Perfect Equilibrium

The strategy always give up



is a SubGame Perfect Equilibrium.

Conclusion

- Reasoning on infinite sequential games is subtle,

Conclusion

- Reasoning on infinite sequential games is subtle,
  - For instance, I am stuck on proving that
    the strategy *Always give up* is a Nash equilibrium.

Conclusion

- Reasoning on infinite sequential games is subtle,
  - For instance, I am stuck on proving that the strategy *Always give up* is a Nash equilibrium.
- Reductio ad absurdum is not needed.

Conclusion

- Reasoning on infinite sequential games is subtle,
  - For instance, I am stuck on proving that the strategy *Always give up* is a Nash equilibrium.
- Reductio ad absurdum is not needed.

Conclusion

- Reasoning on infinite sequential games is subtle,
  - For instance, I am stuck on proving that the strategy *Always give up* is a Nash equilibrium.
- Reductio ad absurdum is not needed.

Fin!