Dependent Types for Extensive Games

Pierre Lescanne

University of Lyon, École normale supérieure de Lyon, CNRS (LIP),
46 allée d'Italie, 69364 Lyon, France

Extensive games are tools largely used in economics to describe decision processes of a community of agents. In this paper we propose a formal presentation based on the proof assistant Coq which focuses mostly on infinite extensive games and their characteristics. Coq proposes a feature called "dependent types", which means that the type of an object may depend on the type of its components. For instance, the set of choices or the set of utilities of an agent may depend on the agent herself. Using dependent types, we describe formally a very general class of games and strategy profiles, which corresponds somewhat to what game theorists are used to. We also discuss the notions of infiniteness in game theory and how this can be precisely described.

Keywords: extensive game, infinite game, sequential game, coinduction, Coq, proof assistant.

1. INTRODUCTION

Extensive games are used in formalization of economics and in decision processes. Rational decision is logic, but it is not exaggerated to claim that rational decision is essentially a computational process and therefore it should be based on computational logic, like the calculus of inductive construction of Coq or the Higher Order Logic of Isabelle [19] and on induction. Moreover, an adequate description of the decision process requires the framework to be infinite. Indeed there is no reason to assume that the process is a priori finite, since if we do so we put strong constraints on the model which prevents some behaviors, like for instance escalation. Beware, in the framework of games where agents interact, we do not say that the world is infinite, but we say that the agents believe that the world is infinite. Indeed, saying that the model is finite precludes the phenomenon of escalation, and proving, in that case, that escalation cannot exist is begging the question. Since we require a computational approach to infinite processes, the natural concept in modern logic is this of coinduction as proposed in [16, 13, 15]. But in this paper, by using dependent types, we revise our previous works. Thus we allow considering formal presentations of very general classes of games, for instance, games with very general sets of choices depending on agents or very general sets of utilities also depending on agents. For instance, an agent may have an infinity of choices and another may have only one choice, or two, whereas utilities are just ordered sets, even completely trivial ones in some counterexamples, which shows their generality. Similarly agents may have their own sets of utility. Agents rank flowers for their color whereas other agents rank them for their fragrance. By very small changes in the formalism, we may easily describe multistage games that are games in which agents move simultaneously at each stage.

One outcome of such a formalization is that it helps clarifying concepts in the theory of extensive or sequential games. In our case, it clarifies two concepts: first the *rationality* of agents (instrumental rationality vs value-rationality, see Section 6), second the *infiniteness* as explained in Section 8.

All the formalism has been developed in Coq [2]. The reader can find scripts on GitHub at
https://github.com/PierreLescanne/DependentTypesForExtensiveGames.

The paper has 8 sections. The second section presents games and strategy profiles. Section 3, Section 4 and Section 6 talk about concepts connected with finiteness. More specifically, Section 3 presents predicates which specify each a particular notion of infiniteness in games or strategy profiles, showing the diversity of infiniteness. Section 4 presents a counterexample and Section 6 speaks about escalation, a known concept in game theory. It relies on the definition of subgame perfect equilibrium given in Section 5. Section 8 considers the way infiniteness is addressed in books on game theory. Section 9 is the conclusion.

2. GAMES AND STRATEGY PROFILES

This presentation of extensive games differs from the one of [16, 13, 15, 1] in the use of dependent types. However it has connections with composition games [8, 11]. Indeed, for simplicity, in those papers, only binary games were considered[1], that is that only two choices were offered to the agents. In this paper, using dependent types, we can propose a more general framework. Associated with a game, a strategy profile is a description of the choices taken by the agents. The formal definitions of games and strategy profiles relies on three entities, a set of agents written Agent, a set of choices – depending on an agent a – written Choice a and a set of utilities – depending on an agent a – written Utility a. Moreover there is a preorder on Utility a. In particular, unlike most of the presentations of games, utilities need not be natural numbers, but can be any ordered set used by the agent. The sets of infinite games and of infinite strategy profiles are defined coinductively and are written Game and StratProf respectively.

Game. A game which does not correspond to a terminal position and which we call a node is written <|a,next|> and has two arguments:

—an *agent* a, the agent whom the node belongs to,
—a function next of type Choice a → Game.

We call *leaf* a terminal position. A leaf consists of a function

(∀ a:Agent, Utility a) → Game

that is a function from an agent a to an element of Utility a. Said otherwise a leaf is the utility assignment at the end of the game. A leaf is written <| f |>. Notice that the utility depends on the agent. A node game is made of an agent and of a function which returns a game given a choice. Recall that if the agent is a and the function is next, then the node game is written <|a,next|>. The formal definition of a game is given in Coq by:

```
CoInductive Game : Set :=
| gLeaf :  (forall a:Agent, Utility a) -> Game
| gNode : forall (a:Agent), (Choice a -> Game) -> Game.
```

[1] After Vestergaard [27] who introduced this concept for finite games and finite strategy profiles.

Since this defines a *coinductive*, it covers finite and infinite extensive games.

Example 1. *Figure 1 provides a picture of a game with* choices *blue* ⇢, *green* ⇝ *and red* ⇢ *for* A *and* black → *and* dotted ⇢ *for* B *and* {*weak*, *medium*, *strong*} *as* utilities *for* A, *and* $\mathbb{N}$ *as* utilities *for* B.

The picture of the game can be described as:

—Ⓐ *is the starting node.*

—*A transition like* (A) ⇢ (A) *means that the agent* A *takes the choice* red *to go to a node owned by* A.

—*A leaf "$A \mapsto weak, B \mapsto 2$" means that the game ends here and that the utilities are attributed:* A *receives* weak, B *receives* 2.

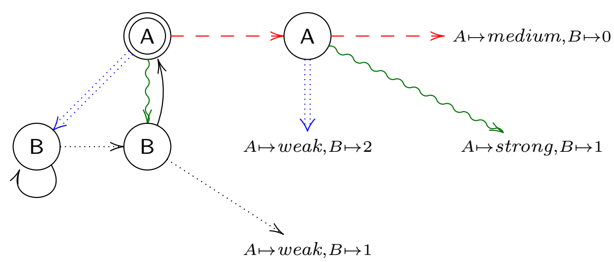


Fig. 1. A Game

Strategy profile. A strategy profile corresponds to a non terminal position. We call it a node and we write it ≪`a,c,next`≫. It has three components:

—an *agent* `a`, the agent whom the node belongs to,

—a *choice* `c`, which is the choice taken by agent on this specific node,

—a function `next` of type `Choice a` → `StratProf`.

A strategy profile which is a terminal position is a function

$$(\forall\ \texttt{a:Agent, Utility a}) \rightarrow \texttt{Game}$$

like for games. Indeed there is no choice. It is written ≪`f`≫. The inductive definition in Coq of a strategy profile is:

```
CoInductive StratProf  : Set :=
| sLeaf : (forall a:Agent, Utility a) -> StratProf
| sNode : forall (a:Agent),
              Choice a -> (Choice a -> StratProf) -> StratProf.
```

The two main differences with the approach of [16, 13, 15, 1] lie in the fact that the set of choices and the set of utilities are not fixed (the same for all agents, namely a pair) but depend on the agent (dependent type). This way we can describe a larger class of games. In Example 1, we have shown a game with choices and games actually depending on the agents. For instance, as we will see in Section 4, the sets of choices can easily be infinite. Since the built-in Coq equality is not adequate, we define coinductively an equality on games,

```
CoInductive gEqual: Game -> Game -> Prop :=
| gEqualLeaf: forall f, gEqual (<| f |>) (<| f |>)
| gEqualNode: forall (a:Agent)(next next':Choice a->Game),
  (forall (c:Choice a), gEqual (next c) (next' c)) ->
  gEqual (<|a,next|>) (<|a,next'|>).
```

written == in the following.

EXAMPLE 2. *Figure 2 is a strategy profile corresponding to the game of Example 1. We represent the choices by double tips.*

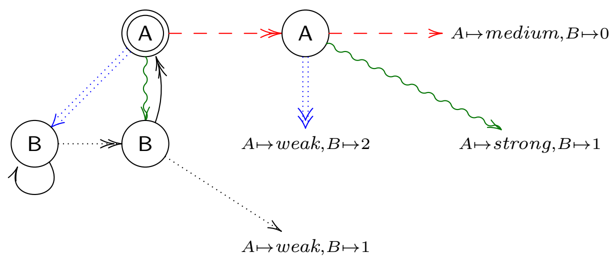


Fig. 2. A strategy profile

The game of a strategy profile. Every strategy profile is supported by a game, thus we can define the game of a strategy profile. It is obtained by "removing" the choices of the strategy profile.

```
Definition game : StratProf -> Game :=
cofix game_co (s : StratProf) : Game :=
  match s with
  | << f >> => <| f |>
  | << a, _, next >> => <| a, fun c : Choice a => game_co (next c) |>
  end.
```

Utility assignment. Since Coq accepts only terminating functions we define the utility assignment as a relation:

```
Inductive Uassign : StratProf ->  (forall a:Agent, Utility a) -> Prop :=
| UassignLeaf: forall f, Uassign (<<f>>) f
| UassignNode: forall  (a:Agent)(c:Choice a)
                       (ua: forall a',Utility a')
                       (next:Choice a -> StratProf),
    Uassign (next c) ua  -> Uassign (<<a,c,next>>) ua.
```

Notice that if we call s the strategy profile of Example 2, we have

Uassign s $(A \mapsto strong, B \mapsto 1)$.

We prove that Uassign is a functional relation, namely that

```
forall s ua ua', Uassign s ua -> Uassign s ua' -> ua=ua'.
```

Notice that for proving this property we need an inversion tactic which is somewhat subtle when dealing with dependent types [5, 18].[2] Moreover for all convergent strategy profiles (that is for all strategy profiles of interest, see next section) we can prove that the function is total, i.e., that there exists always a utility assignment associated with this convergent strategy profile.

3. SEVERAL NOTIONS ASSOCIATED WITH FINITENESS

On potentially infinite games and strategy profiles there are several predicates capturing notions of finiteness.

Finite Games. A game is finite if it has a finite number of positions. It is naturally defined inductively[3]. Clearly a leaf is finite. A game which is a node is finite if the set of the choices of the agent is finite and if, for all the choices, the next games are finite. This is made precise by the following definition. In this definition, `finite` is a predicate on *Set* given as a parameter which states that the set `Choice a` is finite.

```
Inductive Finite : Game -> Set :=
| finGLeaf: forall f, Finite <|f|>
| finGNode: forall (a:Agent)(next: Choice a -> Game),
              finite (Choice a) ->
              (forall c:Choice a, Finite (next c)) ->
              Finite <|a,next|>.
```

Finite strategy profiles would be defined likewise.

```
Inductive FiniteStratProf : StratProf -> Set :=
| finSLeaf: forall f, FiniteStratProf <<f>>
| finSNode: forall (a:Agent)(c:Choice a)(next: Choice a -> StratProf),
              finite (Choice a) ->
              (forall c':Choice a, FiniteStratProf (next c')) ->
              FiniteStratProf <<a,c,next>>.
```

Games with only finitely many strategy profiles. Osborne and Rubinstein [21] call “finite” a game with only finitely many strategy profiles.[4] In order not to interfere with the previous definition, we prefer to say that *the game is finitely broad*.[5] This is translated by the fact that for a game `g` to have only finitely many strategy profiles, there shall exist a list that collects all the strategy profiles that have this game `g` as underlying game. Since in Coq lists are finite this yields the desired property:

```
Definition FinitelyBroad (g:Game): Prop :=
  exists (l: list StratProf), forall (s:StratProf),
     game s == g <-> In s l.
```

[2]We thank Adam Chlipala and Jean-François Monin for their help on this specific example.

[3]Roughly speaking, an inductive (definition) is a well-founded definition with base cases and constructors.

[4]Actually they use the concept of “history” (path), instead of strategy profiles, but this is not essential.

[5]Denoted by the predicate `FinitelyBroad` on `Game` in Coq.

Games with only finite histories. A game has only finite histories if it has only finitely many paths (histories) from the root to the leaves. This can be described as follows:

```
Inductive FiniteHistoryGame : Game -> Prop :=
| finHorGLeaf: forall f, FiniteHistoryGame <|f|>
| finHorGNode: forall (a:Agent)(next: Choice a -> Game),
             (forall c':Choice a, FiniteHistoryGame (next c')) ->
             FiniteHistoryGame <|a,next|>.
```

Those games should not be confused with games with finite horizon. Notice that Osborne and Rubinstein [21] require a game with a finite horizon to have only finitely many strategy profiles (p. 90: "[Given a finite game] if the longest history is finite then the game has finite horizon"), whereas Osborne [20] does not require the set of strategy profiles associated to the game to be finite (see Section 8). For strategy profiles we have:

```
Inductive FiniteHistoryStratProf : StratProf -> Prop :=
| finHorSLeaf: forall f, FiniteHistoryStratProf <<f>>
| finHorSNode: forall (a:Agent) (c:Choice a)
                      (next: Choice a -> StratProf),
             (forall c':Choice a, FiniteHistoryStratProf (next c')) ->
             FiniteHistoryStratProf <<a,c,next>>.
```

Convergent strategy profiles. Convergent strategy profiles are strategy profiles to which we can always assign payoffs: we have just to follow the choice of the agent at each node we cross. Due to the convergence of the strategy profile we are sure to reach a leaf where payoffs are eventually attributed. Not all strategy profiles are convergent, but convergent strategy profiles are selected by a predicate. Said otherwise, a non leaf convergent strategy profile is a strategy profile, such that when we follow the choice of the agent we reach a convergent strategy profile. Since we say that we finitely reach a leaf, this definition is inductive. It can be obtained by just transforming the definition of `FiniteHistoryStratProf`. Indeed in the definition of `Convergent`, the "finiteness" does not apply to all paths (histories) leading to leaves, but applies only to paths corresponding to the choices of the agents. Mutatis mutandi, the expression

```
(forall c':Choice a, FiniteHistoryStratProf (next c'))
```

is just replaced by

```
Convergent (next c)
```

hence without the

```
forall c':Choice a
```

Like for the predicate `FiniteHistoryGame` a leaf is convergent. A strategy profile which is a node is convergent if the strategy subprofile for the choice made by the agent `a` (i.e., `next c`) is convergent.

```
Inductive Convergent: StratProf -> Prop :=
| ConvLeaf: forall f, Convergent <<f>>
```

```
| ConvNode: forall (a:Agent) (c:Choice a)
                   (next: Choice a -> StratProf),
            Convergent (next c) ->
            Convergent <<a,c,next>>.
```

The reader may notice the similarity of that definition with the one of finite histories for games. The strategy profile of Example 2 is convergent. We are now able to prove a theorem on the totality of `Uassign`:

```
Lemma ExistenceUassign:
  forall (s:StratProf),
    (Convergent s) -> exists (ua: forall a, Utility a), Uassign s ua.
```

Modality Always *and Predicate* Always Convergent*.* Convergence is extended to all the strategy subprofiles of a given strategy profile by a modality `Always`. `Always` applies to a predicate on `StratProf`, that is a function `P:StratProf` $\rightarrow$ `Prop` `Always P s` means that `P` is fulfilled by all subprofiles of `s`.

```
CoInductive Always (P:StratProf -> Prop) : StratProf -> Prop :=
| AlwaysLeaf : forall f, Always P (<<f>>)
| AlwaysNode : forall (a:Agent)(c:Choice a)
                      (next:Choice a->StratProf),
          P (<<a,c,next>>) -> (forall c', Always P (next c')) ->
               Always P (<<a,c,next>>).
```

`Always Convergent s` means that `s` is convergent and also all subprofiles are convergent. It plays a main role in the definition of other concepts related to strategy profiles, namely equilibria and escalation. Always convergent strategy profiles are the objects that game theorists are interested in. Indeed they are infinite strategy profiles in which it is always possible to assign a value (a payoff) to every game position. This way, one can compare the strength of two game positions. For instance, a strategy profile associated with the game of Figure 5 in which both players take choice `right` at each position is not convergent and obviously not always convergent. It cannot be considered for a strategy in the game because a payoff cannot be attributed.

4. A GAME WITH ONLY FINITE HISTORIES AND NO LONGEST HISTORY

In this section we show how `Coq` can be used to prove formally properties about games. Specifically we give an example of a game with only finite histories and no longest history as a counterexample to Osborne's (see [20] p. 157) definition of finite horizon. The game has two agents whom we call `Alice` and `Bob` and its definition uses a feature of dependent types, namely that the choices may depend on the agent. In this case, `Alice` has infinitely many choices, namely the set `nat` of natural numbers and `Bob` has one choice, namely the set `unit`. As they are singletons, the *utility* of *Alice* and the *utility* of *Bob* play no role for discriminating among strategy profiles by payoff, the `Coq` built-in `unit` which contains the only element `tt` model a singleton. In `Coq` we have:

```
Definition Choice :(AliceBob -> Set) :=
  fun a:AliceBob  => match a with Alice => nat | Bob => unit end.
```

and

```
Definition Utility: AliceBob -> Set := fun a => unit.
```

Notice that `Choice` and `Utility` are functions which take an agent and return a set. Said otherwise, the set of choices is the result of the function `Choice` applied to agents and the set of utilities is the result of the function `Utility` applied to agents. If the agent is `Alice`, the set of choices is `nat` and the set of utilities is `unit`. If the agent is Bob the set of choices and the set of utilities are `unit` (a singleton). In other words, the set of choices depends on the agents and the set of utilities looks like it depends on the agents, but doesn't. The game has infinitely many threadlike subgames of length n:

```
Fixpoint  ThreadlikeGame (n:nat): (Game AliceBob Choice Utility) :=
  match n with
    | 0 => <|fun (a:AliceBob) => match a with | Alice => tt
                                               | Bob => tt end|>
    | (S n) => <|Bob,fun c:Choice Bob
                     => match c with tt=>ThreadlikeGame n end|>
  end.
```

The game we are interested in is called `GameWFH` and is defined as a node with agent `Alice` and with next games `ThreadlikeGame n` for Alice's choice `n`:

```
Definition GameWFH:(Game AliceBob Choice Utility)  :=
  <| Alice, fun n:Choice Alice => ThreadlikeGame n |>.
```

Let us call `triv` the utility assignment `Alice => tt, Bob => tt`. We can picture `GameWFH` like in Figure 3. One can prove that `ThreadlikeGame n` has only finite histories:

```
Proposition FiniteHistoryGameWFH:
    FiniteHistoryGame AliceBob Choice Utility GameWFH.
```

Clearly `GameWFH` has no longest history.

5. SUBGAME PERFECT EQUILIBRIUM

An agent is rational if her strategy is based on a strategy profile which is a subgame perfect equilibrium. So let us present *subgame perfect equilibria*. Subgame perfect equilibria are specific strategy profiles that fulfill some "good" properties for the agent. Therefore they are presented by a predicate which we call SPE. In Coq this is a function of type `StratProf -> Prop`. A strategy profile, which is a node, is a subgame perfect equilibrium if first it is always convergent. This is necessary to be able to compute the utility assignments for the strategy subprofiles. Moreover the choice of the agent is better than or equal to other choices with respect to the utility assignment and all the strategy subprofiles of this strategy profile are themselves subgame perfect equilibria. A leaf is a subgame perfect equilibrium. This can be formalized in Coq (see Figure 4). Rational agents choose a strategy profile which is SPE.

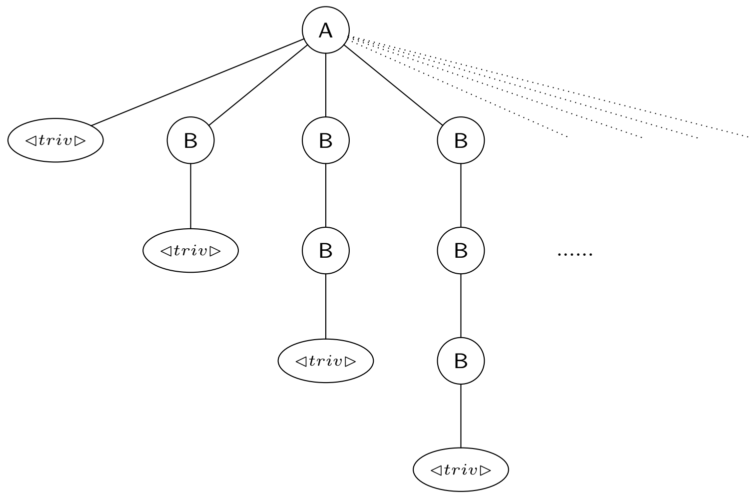


Fig. 3. Picture of game with finite histories and no longest history

```
CoInductive SPE : StratProf -> Prop :=
| SPELeaf : forall (f: forall a:Agent, Utility a), SPE <<f>>
| SPENode :  forall (a:Agent)
                    (c c':Choice a)
                    (next:Choice a->StratProf)
                    (ua ua':forall a':Agent, Utility a'),
               Always convergent <<a,c,next>> ->
               Uassign (next c') ua' ->  Uassign (next c) ua ->
               (pref a (ua' a) (ua a)) -> SPE (next c') ->
             SPE <<a,c,next>>.
```

Fig. 4. The predicate `SPE`

6. THE SIMPLEST ESCALATION

We discussed already the rationality of escalation in infinite games in other papers [16, 15]. Recall the story. In 1971, Shubik [25] proposed a game called the *dollar auction* (pictured in Figure 5), in which an apparently paradoxical[6] situation occurs. Agents keep betting despite this goes against their own interest, as observed from outside. This situation is called *escalation*. With the mathematical and logic tools of that time, this paradox could not be addressed and the behavior was attributed to the irrationality of the agents. Since Shubik [25], all the textbooks on game theory which address the dollar auction attribute the escalation to the irrationality of the agents, despite it was known since Max Weber [28, 29] in 1921 that there are two types of reasoning for agents: an agent can be *instrumentally rational* (zweckrational) or *value-rational* (wertrational). These two levels of rationality have been rediscovered by Stanovich [26] as *instrumental rationality* and

[6]This is paradoxical only from the point of view of an observer.

There are two bidders (A and B) for an object of **1$**. They bid one after the other by putting **5¢** in an urn with no return. Therefore payoff is negative after 20 turns. When a bidder gives up, the other gets the object. If a bidder decides to stop after $n + 20$ turns

—the bidder who does not have the object has a payoff of $-5n$¢ and
—the bidder who has the object has a payoff of $(100 - 5n)$¢.

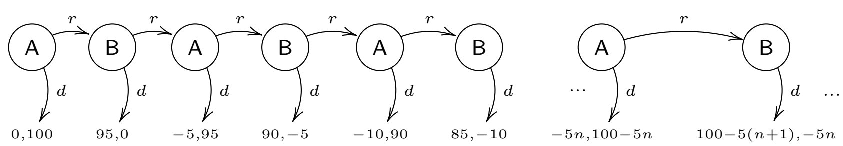


Fig. 5. The dollar auction

epistemic rationality[7] or *algorithmic mind* and *reflexive mind* in another terminology. See [13] Section 6.3 for a first discussion of a connection with escalation. If agents are *value-rational*, which is the behavior of agents involved in games and, among them, the dollar auction, they can enter rationally in an escalation. Weber wrote:

> From the point of view of instrumental rationality, value-rationality is always irrational.

We would say "from the point of view of an external observer, value-rationality looks irrational". If an agent is *instrumentally rational* (zweckrational), which assumes a degree of abstraction (abstraction from the closed world in which she lives and which we would call the "game"), she aims at avoiding escalation, since she looks at the game like she would be an external observer. Weber wrote:

> An instrumentally rational (zweckrational) action is determined by expectations as to the behavior of objects in the environment and of other human beings.

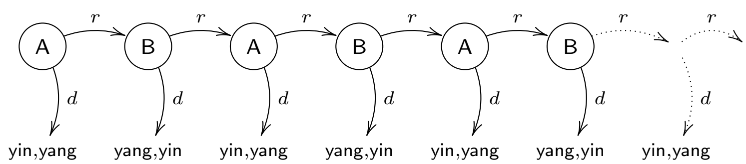
In this attitude of instrumental rationality (Zweckrationalität), an agent can potentially revise her belief. In other words, before she realizes that she enters an escalation, she believes in an infinite world of resources. After she realizes the danger of an escalation she revises her belief toward a belief in a finite world of resources.

An escalation occurs as follows: at each step, the agent, who has the play, chooses an action which corresponds to a strategy profile which is an SPE. Since at each step the two possible actions *stop* (`down`) or *continue* (`right`) belong to a subgame perfect equilibrium (the two are possible at each step), it is perfectly rational (zweckrational) for each agent to continue forever. This is shown formally in Coq.

In a previous work, the present author addressed the escalation in the dollar auction [16], but the concept of escalation can be abstracted in a game where other

[7]Beware of confusing terminology: What Stanovich calls *instrumentally rational* is what Weber calls *wertrational* and what his translators call *value-rational*. What Stanovich calls *epistemicly rational* is what Weber calls *zweckrational* and his translators call *instrumentally rational*.

considerations are eliminated. In this framework, escalation is a somewhat simple concept and consists on adjusting the types. The simplest escalation is probably as follows. It may occur in a game in which there are two agents `Alice` and `Bob`, where each agent has two choices `down` and `right` and in which there are two non ordered utilities `yin` and `yang`. We use `yin` and `yang` to insist on the fact that there is no need for numbers and no need for an actual order among the utility values.



This is basically the game studied in [15], with the difference that the preference in `Utility = {yin, yang}` is just the equality. In other words, agents do not need to prefer one item over the other, just a trivial preference may lead to an escalation. The agents are like Buridan's ass [30], they may not know what to choose and therefore go forever. This may look strange, but as shown by the Coq script, the proof is based on exactly the same proof technique as the one of the rationality of the escalation of the dollar auction [25] as shown by the two following Coq statements and proofs.[8]

```
Lemma AlongGoodAndDivergentInDollar :
  exists (s:StratProf dollar.Agent dollar.Choice dollar.Utility),
    AlongGood dollar.Agent dollar.Choice dollar.Utility dollar.pref s
    /\ Divergent s.
Proof.
  exists (dollarAcBc 0).
  split.
  apply AlongGoodDolAcBc.
  apply DivergenceDolAcBc.
Qed.
```

and the proof of the escalation for the *YinYang game*:

```
Lemma AlongGoodAndDivergentInYinYang :
  exists (s:StratProf yinYang.Agent yinYang.Choice yinYang.Utility),
    AlongGood yinYang.Agent yinYang.Choice yinYang.Utility yinYang.pref s
    /\ Divergent s.
Proof.
  exists yinYangAcBc.
  split.
  apply AlongGoodYyAcBc.
  apply DivergenceYyAcBc.
Qed.
```

[8]Notice that the parameters of StratProf are explicit and that the proof is made in a framework in which only the choice might depend on the agent. The utility is set to be the same for all the agents.

The statement `AlongGoodAndDivergentIn Game` says that there exists a strategy profile of the game (Dollar or YinYang) which is both good (aka rational) and divergent. This is exactly what we expect when looking for escalation.

7. MULTI-STAGE GAMES

Multi-stage games are introduced in [7] (Section 3.2). We view them as games in which a node does not belong to a single agent and the choices or the moves of all the agents are simultaneous. Let us call `MSGame` the multi-stage games. The simultaneous or collective choice corresponds to the type:

```
(forall a:  Agent, Choice a) -> MSGame
```

or written with products:

$$\prod_{\mathtt{a} \in \mathtt{Agent}} \mathtt{Choice\ a}.$$

Leaves are almost unchanged. The function `next` is of type

$$\mathtt{next:}\quad (\prod_{\mathtt{a} \in \mathtt{Agent}} \mathtt{Choice\ a}) \to \mathtt{MSGame}$$

and a node is just the function next:

```
CoInductive MSGame :=
| msgLeaf: (forall a: Agent, Utility a) -> MSGame
| msgNode: ((forall a: Agent, Choice a) -> MSGame) -> MSGame.
```

Example 3. *T show the complexity of multistage games, we draw in Figure 6 a picture of a simple multistage game with the same choices and utilities as Example 1.*

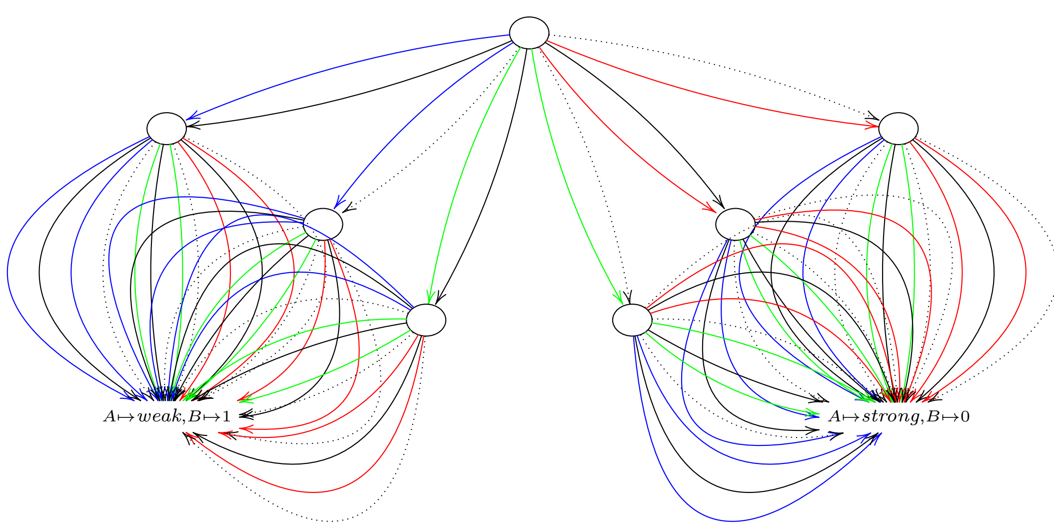


Fig. 6. A simple multistage game

8. INFINITE AND INFINITE

On disputa longuement si infini et illimité signifient la même chose.
One discussed at length whether infinite and unlimited mean the same thing.

Marguerite Yourcenar, L'œuvre au Noir [31], in English The Abyss [32]

The debate on the sort of infinities started at the Renaissance. For instance, Giordano Bruno wrote his *De l'Infinito, Universo e Mondi* [3] where he claimed the infiniteness of the world. The above citation of Marguerite Yourcenar in her well documented novel *The Abyss* [32], which takes place at the same period (the end of the XVI$^{\text{th}}$ century) relates the nature of the debate. At the beginning of this research, the fact that, following Shubik [25], we should consider infinite games, hence coinductive games was rather clear, the result we obtained from the proof assistant COQ and the reaction of the readers who assimilated an infinite game to repeated games show that the debate of the Renaissance was not closed and should be made clear. In this section, we look at the way infiniteness is dealt with in textbooks on game theory.

Two views of infiniteness

Infiniteness is discussed by Poincaré in his book *Science et méthode* [22], where he distinguishes *mathematical infinite* which we would call today *potential infinite*, and *actual infinite*. Poincaré did not believe in the necessity of an actual infinite in mathematics, but today we do accept a mathematical concept of actual infinite which is the foundation of the theory of coinduction and infinite games. Let us discuss these two concepts in the case of words on the alphabet $\{a, b\}$. $\{a, b\}^+$ represents all the (finite) words made with the letters a and b, like $a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb$, *etc.* One can also write:

$$\{a,b\}^+ \;=\; \bigcup_{n=0}^{\infty} \{a,b\}\,\{a,b\}^n. \tag{1}$$

$\{a, b\}^+$ is the least fixpoint of the equation:

$$X \;=\; \{a,b\} \cup \{a,b\}X$$

There are infinitely many such words. This is a first kind of infinite, indeed we can build words of all finite lengths. $\{a, b\}^\omega$ is the set of infinite words. Each infinite word can be seen as a function $\mathbb{N} \to \{a, b\}$. An infinite word represents another kind of infinite. For instance the infinite word $ababab...$ or $(ab)^\omega$ corresponds to the function if $\mathsf{even}(n)$ $then$ a $else$ b and is a typical example of an actual infinite. $\{a, b\}^\omega$ is a solution of the fixpoint equation:

$$X \;=\; \{a,b\}X.$$

In $\{a, b\}^+$ there is no infinite object, but only approximations, whereas in $\{a, b\}^\omega$ there are only infinite objects.

Figure 7 represents the two notions of infiniteness. On the left, the vault ceiling of Nasir ol Molk Mosque in Chiraz[9] pictures potential infiniteness. On the right, a drawing[10] inspired by M.C. Escher's Waterfall pictures actual infiniteness.

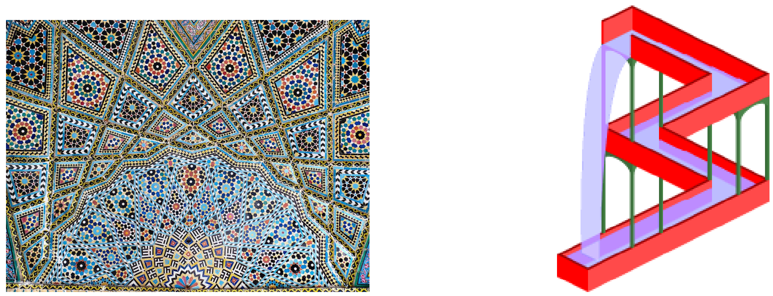


Fig. 7. Two pictures of infinite

Common Knowledge. Common Knowledge is a central concept in game theory, it relies on the concept of knowledge of an agent, which is a modality i.e., an operator of modal logic.[11] Modality K_a (knowledge of agent a) follows the laws of modal logic S_5. For this and the group G of agents, we create a modality E_G (shared knowledge):

$$E_G(\varphi) = \bigwedge_{a \in G} K_a(\varphi).$$

The common knowledge modality is

$$C_G(\varphi) \;=\; \bigwedge_{n=0}^{\infty} E_G^n(\varphi). \tag{2}$$

Usually there is no ambiguity on the group of agents, thus instead C_G and E_G one write just C and E. Clearly C has the flavor of $+$ as shown by the analogy between equation (1) and equation (2) and their fixpoint definitions.

Infinite and fixpoint. Infinite objects are associated with fixpoints. For instance, $\{a, b\}^+$ is the least fixpoint of the equation:

$$X \;=\; \{a, b\} \cup \{a, b\}X$$

whereas $C(\varphi)$ is the least fixpoint of

$$X \;\Leftrightarrow\; \varphi \wedge E(X).$$

which means that $C(\varphi)$ is a solution:

$$C(\varphi) \;\Leftrightarrow\; \varphi \wedge E(C(\varphi)).$$

---

[9]From Wikimedia, due to User:Pentocelo.
`https://commons.wikimedia.org/wiki/File:Nasr_ol_Molk_mosque_vault_ceiling_2.jpg`
[10]From Wikimedia, due to :User:AndrewKepert.
`https://commons.wikimedia.org/wiki/File:Escher-Waterfall.png`
[11]We follows the presentation of [12, 14], which took its origin from [6].

Infinite in textbooks. In general in textbooks on game theory "infinite" is a vague notion which is not defined precisely and words like "ad infinitum" ([7] p. 542, [10] p. 27) or "infinite regress" ([7] p. 543) or three dots are used. It is often said that infinite games resemble repeated games, but this is not true, since repeated games are typically potential infinite presentations of infinite games, i.e., approximations – only sequences of games are considered, not their limit – whereas infinite games are defined by coinduction.

Two main mistakes are worth noticing.

—In [10], Hargreaves and Varoufakis define common knowledge as follows:
  (a) each person is instrumentally rational[12]
  (b) each person knows (a)
  (c) each person knows (b)
  (d) each person knows (c)
  ... and so on *ad infinitum*.

  but they add "The idea reminds one of what happens when a camera is pointing to a television screen that conveys the image recorded by the very same camera: *an infinite self-reflection*", showing that they clearly mixed up the two kinds of notions. Indeed clearly the infinite self-reflection illustrates an actual infinite, a little like the infinite word $(ab)^\omega$ or the Escher's waterfall, whereas, as we said, common knowledge is a potential infinite. An expression like *ad libitum* should have been preferred and the image of a swing going further and further or a tessellation, like this of Figure 7 should have been more appropriate.

—In [20], Osborne uses the "length of longest terminal history" to define finite horizon, without checking whether this longest history actually exists. A counterexample is shown in Section 4. We gather that he means the "least upper bound on $\overline{\mathbb{N}}$ of the lengths of the histories".

9. CONCLUSION

This paper has shown the elegance of dependent types in formalizing a domain where it applies naturally. Indeed, the main results of non probabilistic game theory do not depend on the fact that the choices or the utilities are of a specific type and despite the inherent technicality of dependent types in proof assistants, the concepts can be defined elegantly at a rather high level of abstraction.

The idea of applying a proof assistant in social science is not new. Let us mention the proof of Arrow's theorem by Nipkow [19], the work of Capretta [4] and of the author [14, 17] in logic of common knowledge and the work of Le Roux et al. [24] on Nash equilibria. All these approaches, including this of the present paper, show that the first step is a formalization of the field and a clarification of the concepts and the last step is a set of consequences that can be drawn from the formalization. In between all the steps are rigorously justified. Obviously this pays off since notions that were unclear become clearer (like rationality and infiniteness), results are strengthened (like the existence of a secure equilibrium among Nash equilibria [24]) and facts that were considered as established (like the irrationality of escalation) are refuted. It should be therefore interesting to address

[12] In the sense of Stanovich, or value-rational (wertrational) in the sense of Weber.

fields of social science, where logic aspects are dominant. Social choices and game theory are good candidates. For instance, one challenge is *belief revision*, with a specific application to the dollar auction. The use of a modern proof assistant would be especially appropriate for belief revision, since its current theory addresses essentially propositional calculus [9], whereas in the dollar auction, the belief to be revised is *finiteness* which a typical second-order statement.

References

[1] Samson Abramsky and Viktor Winschel. Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting. *arXiv*, 2012.

[2] Bruno Barras, Samuel Boutin, Cristina Cornes, Judicaël Courant, Yann Coscoy, David Delahaye, Daniel de Rauglaudre, Jean-Christophe Filliâtre, Eduardo Giménez, Hugo Herbelin, Gérard Huet, Henri Laulhère, César Muñoz, Chetan Murthy, Catherine Parent-Vigouroux, Patrick Loiseleur, Christine Paulin-Mohring, Amokrane Saïbi, and Benjamin Werner. *The Coq Proof Assistant Reference Manual.* INRIA, version 6.3.11 edition, May 2000.

[3] Giordano Bruno. De l'infinito, universo e mondi. Stampato in Venetia, 1594.

[4] Venanzio Capretta. Common knowledge as a coinductive modality. In Erik Barendsen, Herman Geuvers, Venanzio Capretta, and Milad Niqui, editors, *Reflections on Type Theory, Lambda Calculus, and the Mind*, pages 51–61. ICIS, Faculty of Science, Radbout University Nijmegen, 2007. Essays Dedicated to Henk Barendregt on the Occasion of his 60th Birthday.

[5] Adam Chlipala. *Certified Programming with Dependent Types.* MIT Press, 2011. `http://adam.chlipala.net/cpdt/`.

[6] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge.* The MIT Press, 1995.

[7] Drew Fudenberg and Jean Tirole. *Game Theory.* MIT Press, 1991.

[8] Neil Ghani and Jules Hedges. A compositional approach to economic game theory. *ArXiv.org*, abs/1603.04641, 2016.

[9] Sven Ove Hansson. *The Stanford Encyclopedia of Philosophy*, chapter Logic of Belief Revision. Edward N. Zalta (ed.), Winter 2017 Edition.

[10] Shaun Hargreaves-Heap and Yanis Varoufakis. *Game Theory: A Critical Introduction.* Routledge, 2004.

[11] Jules Hedges, Paulo Oliva, Evguenia Shprits, Viktor Winschel, and Philipp Zahn. Selection equilibria of higher-order games. In Yuliya Lierler and Walid Taha, editors, *Practical Aspects of Declarative Languages - 19th International Symposium, PADL 2017, Paris, France, January 16-17, 2017, Proceedings*, volume 10137 of *Lecture Notes in Computer Science*, pages 136–151. Springer, 2017.

[12] Pierre Lescanne. Mechanizing common knowledge logic using COQ. *Ann. Math. Artif. Intell.*, 48(1-2):15–43, 2006.

[13] Pierre Lescanne. Rationality and escalation in infinite extensive games. *ArXiv.org*, abs/1112.1185, 2011.

[14] Pierre Lescanne. Common knowledge logic in a higher order proof assistant. In Andrei Voronkov and Christoph Weidenbach, editors, *Programming Logics*

- *Essays in Memory of Harald Ganzinger*, volume 7797 of *Lecture Notes in Computer Science*, pages 271–284. Springer, 2013.

[15] Pierre Lescanne. A simple case of rationality of escalation. In Reiko Heckel and Stefan Milius, editors, *CALCO*, volume 8089 of *Lecture Notes in Computer Science*, pages 191–204. Springer, 2013.

[16] Pierre Lescanne and Matthieu Perrinel. "Backward" coinduction, Nash equilibrium and the rationality of escalation. *Acta Inf.*, 49(3):117–137, 2012.

[17] Pierre Lescanne and Jérôme Puisségur. Dynamic logic of common knowledge in a proof assistant. *ArXiv.org*, abs/0712.3146, 2007.

[18] Jean-François Monin and Xiaomu Shi. Handcrafted inversions made operational on operational semantics. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings*, volume 7998 of *Lecture Notes in Computer Science*, pages 338–353. Springer, 2013.

[19] Tobias Nipkow. Social choice theory in HOL. *J. Autom. Reasoning*, 43(3):289–304, 2009.

[20] Martin J. Osborne. *An Introduction to Game Theory.* Oxford, 2004.

[21] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory.* The MIT Press, Cambridge, Massachusetts, 1994.

[22] Henri Poincaré. *Science et méthode.* Ernest Flammarion, Paris, 1908. Translated in English as [23].

[23] Henri Poincaré. *Science and Method.* Thomas Nelson and Sons, 1914. Translated by Francis Maitland, available online at `https://archive.org/details/sciencemethod00poinuoft`.

[24] Stéphane Le Roux, Arno Pauly, and Jean-François Raskin. Minkowski games. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 50:1–50:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[25] Martin Shubik. The dollar auction game: A paradox in noncooperative behavior and escalation. *Journal of Conflict Resolution*, 15(1):109–111, 1971.

[26] Keith E. Stanovich. *Rationality and the reflective mind.* Oxford University Press, 2011.

[27] René Vestergaard. A constructive approach to sequential Nash equilibria. *Inf. Process. Lett.*, 97:46–51, 2006.

[28] Max Weber. *Wirschaft und Gesellshaft.* Mohr Siebeck, 1922.

[29] Max Weber. *Economy and Society.* University of California Press, 1968. Guenther Roth and Claus Wittich, *Ed.*

[30] Wikipedia. Buridan's ass — wikipedia, the free encyclopedia, 2016. [Online; accessed 13-July-2016].

[31] Marguerite Yourcenar. *L'Œuvre au Noir.* Gallimard, 1968.

[32] Marguerite Yourcenar. *The Abyss.* Farrar, Straus & Giroux, 1976.