



Bel-Games: A Formal Theory of Games of Incomplete Information Based on Belief Functions in the Coq Proof Assistant

Pierre Pomeret-Coquot, Hélène Fargier, Érik Martin-Dorel

► To cite this version:

Pierre Pomeret-Coquot, Hélène Fargier, Érik Martin-Dorel. Bel-Games: A Formal Theory of Games of Incomplete Information Based on Belief Functions in the Coq Proof Assistant. 2022. hal-03782650v1

HAL Id: hal-03782650

<https://ut3-toulouseinp.hal.science/hal-03782650v1>

Preprint submitted on 21 Sep 2022 (v1), last revised 27 Feb 2023 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License


Bel-Games: A Formal Theory of Games of Incomplete Information Based on Belief Functions in the Coq Proof Assistant

Pierre Pomeret-Coquot ✉ 

IRIT, Université de Toulouse III - Paul Sabatier, France

Hélène Fargier ✉ 

IRIT, CNRS, Toulouse, France

Érik Martin-Dorel ✉ 

IRIT, Université de Toulouse III - Paul Sabatier, France

Abstract

Decision theory and game theory are both interdisciplinary domains that focus on modeling and studying decision-making processes. On the one hand, decision theory aims to account for the possible behaviors of an agent with respect to an uncertain situation. It thus provides several frameworks to describe the decision-making processes in this context, including that of belief functions. On the other hand, game theory focuses on multi-agent decisions, typically with probabilistic uncertainty (if any), hence the so-called class of Bayesian games. In this paper, we extend Bayesian games to the theory of belief functions. We obtain a more expressive class of games we refer to as Bel games; it makes it possible to better capture human behaviors with respect to lack of information. Next, we prove an extended version of the so-called Howson–Rosenthal’s theorem, showing that Bel games can be turned into games of complete information, i.e., without any uncertainty. Doing so, we embed this class of games into classical game theory and thereby enable the use of existing algorithms. Using the Coq proof assistant, we formalize a theory of belief functions, and formally verify three different proofs of Howson–Rosenthal’s theorem.

2012 ACM Subject Classification Theory of computation → Proof theory; Theory of computation → Logic and verification; Theory of computation → Algorithmic game theory; Theory of computation → Solution concepts in game theory; Theory of computation → Representations of games and their complexity

Keywords and phrases Game of Incomplete Information, Belief Function Theory, Proof Assistant, MathComp/SSReflect

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.1

Related Version *Pen-and-paper proof article:* HAL [30]

Source code: GitHub repository

Funding *Pierre Pomeret-Coquot:* ANITI, funded by the French “Investing for the Future – PIA3” program under the Grant agreement n°ANR-19-PI3A-0004.

1 Introduction

The positioning of our work lies at the intersection of decision-making under uncertainty, game theory, and interactive theorem proving. While decision theory typically deals with single-agent decision making in an uncertain context, game theory gathers several frameworks that all involve several agents, and can model situations in many application areas such as economics, politics, logics, artificial intelligence, or biology... For example, Bayesian games

constitute one such framework in the scope of games of incomplete information (denoted “IGames” in the following), which has been well-studied by the community [14, 26].

In [10, 30], we introduce the framework of *Bel games* (which encompasses that of Bayesian games) based on belief functions, and generalize Howson–Rosenthal’s 2-player theorem to this framework and to any number of players. More specifically, we provide three algorithmic transforms that turn a Bel-game into a game of complete information (denoted “CGame” in the following), along with a pen-and-paper proof of correctness for these three transforms. In this paper, we formally define Bel-games using the Coq proof assistant, and present a formally-verified implementation of these transforms. The objective of this formalization effort is twofold: provide strong guarantees on the correctness of the results as well as make explicit necessary conditions and improve the sequence of proof steps; and setup a generic, formal library that allows one to formally reason about IGames. To the best of our knowledge, it is the first attempt to model IGames in a formal setting. Moreover, we focus on theoretical foundations that go beyond the classical setting of Bayesian games, relying on Belief functions theory, while addressing Nash equilibria with mixed strategies.

Related works Several formalization efforts have been carried out in game theory since 2006, each focusing on a somewhat different fragment: Vestergaard [38] then Le Roux [31], formalizing Kuhn’s existence of a Nash equilibrium in finite games in extensive form, using Coq; Lescanne et al. [22], studying rationality of infinite games in extensive form, using Coq; Martin-Dorel et al. [24], studying the probability of existence of winning strategies in Boolean finite games, using Coq; Bagnall et al. [2], formalizing well-known results of algorithmic game theory, using Coq; Dittmann [5], proving the positional determinacy of parity games, using Isabelle/HOL; Le Roux et al. [20], proving that a determinacy assumption implies the existence of Nash equilibrium in 2-player games, using Coq and Isabelle/HOL; this result being combined with that of Dittmann, using Isabelle/HOL [21].

Paper outline We start in Section 2 by reviewing a typical example of reasoning under uncertainty, where the usual framework of *Bayesian games* fails. In Section 3, we recall background knowledge about belief functions and formalize it in the Coq proof assistant. In Section 4, we formalize some standard notions of game theory such as hypergraphical games. Then in Section 5, we formalize Bel-games (which generalize Bayesian games), as well as the notion of Nash equilibria for these classes of IGames. In Section 6, we formally prove a generalized Howson–Rosenthal’s theorem (for n -player Bel games in lieu of 2-player Bayesian games) for three different transforms that enjoy slightly different properties. Finally, we draw some conclusions and perspectives for future work in Section 7.

2 Motivating Example: the Murder of Mr. Jones

To illustrate and motivate our work, we use the following example inspired by the *murder of Mr. Jones* [37], where the suspects are *Peter, Paul, and Mary*.

► **Example 1** (Peter, Quentin, and Rose). Two agents, named Agent 1 and Agent 2, are independently looking for a business association, with either Peter (P), Quentin (Q), or Rose (R). The point is that a crime has been committed, for which these three people only are suspected. Furthermore, a poor-quality surveillance video allow to estimate that there is a 50% chance that the culprit is a man (P or Q), and a 50% chance that it is a woman (R). As to the interest of the associations, making the deal with an innocent people leads to a payoff of \$6k (to be shared between the people making the deal), while associating with the culprit produces no payoff (\$0k). Moreover, Agent 1 is investigating about P and will know whether he is guilty before making the decision. Similarly, Agent 2 will know whether R is guilty.

It appears the Bayesian approach is not relevant here: it does not capture the intended knowledge revision. Indeed, if Agent 1 learns that P is innocent, the intended behavior of Agent 1 is to keep 50% chances per sex, so the probability of guilt should become $1/2$ for Q and $1/2$ for R . However, in a purely Bayesian view, equiprobability would be applied and the prior probability of guilt would be $1/4$ to P and $1/4$ to Q . Then, after conditioning, Agent 1 would get a probability of $1/3$ to Q and $2/3$ to R .

3 Formalization of Belief Functions for Individual Decision Making

3.1 Knowledge Representation

The theory of belief functions is a powerful toolset from decision theory and statistics, which generalizes probability theory. It encompasses two distinct approaches for reasoning under uncertainty: Dempster-Shafer theory of evidence (DS) [4, 35] and upper-lower probability theory (ULP) [4, 39].

► **Definition 2** (Basic probability assignment). *A set-function $m : 2^\Omega \rightarrow [0, 1]$ is a basic probability assignment (bpa, a.k.a. mass function) iff $m(\emptyset) = 0$, $\sum_{A \subseteq \Omega} m(A) = 1$ and $\forall A \subseteq \omega, m(A) \geq 0$.*

► **Definition 3** (Belief function and plausability measure). *A belief function $\text{Bel} : 2^\Omega \rightarrow [0, 1]$ is a monotonic measure which is subadditive ($\forall A, B$ distinct, $\text{Bel}(A \cup B) \geq \text{Bel}(A) + \text{Bel}(B)$) and verifies $\text{Bel}(\emptyset) = 0$ and $\text{Bel}(\Omega) = 1$. The dual plausibility measure Pl is defined by $\text{Pl}(A) = 1 - \text{Bel}(A^c)$, and is superadditive.*

► **Definition 4** (bpa-characterization of a belief function and a plausibility measure). *Both Bel and Pl can be encoded by a single bpa $m : 2^\Omega \rightarrow [0, 1]$ such that $\text{Bel}(A) = \sum_{B \subseteq A} m(B)$ and $\text{Pl}(A) = \sum_{B, B \cap A \neq \emptyset} m(B)$. Conversely, any bpa m defines a pair of dual belief and plausibility measures.*

We formalize these first definitions in Coq [3], relying on the MathComp library [23] that provides a comprehensive formalization of finite sets, functions with finite support (endowed with decidable equality) as well as big operators:¹

```
Variables (R : realFieldType) (W : finType).

Definition bpa_axiom (m : {ffun {set W} -> R}) :=
  [&& m set0 == 0 , \sum_A m A == 1 & [\forall A, m A >= 0]].

Structure bpa := { bpa_val :> {set W} -> R ; bpa_ax : bpa_axiom bpa_val }.

Definition Bel (m : bpa) : {set W} -> R := fun A => \sum_(B | B \subset A) m B.
Definition Pl (m : bpa) : {set W} -> R := fun A => \sum_(B | B :&: A != set0) m B.

Lemma BelE (m : bpa) (A : {set W}) : Bel m A = 1 - Pl m (~:A).
Lemma PlE (m : bpa) (A : {set W}) : Pl m A = 1 - Bel m (~:A).
```

► **Definition 5** (Focal elements and focal set). *Sets with a non-zero mass are called “focal elements”, and the set of focal elements, denoted by S_m , is called “focal set”:*

¹ MathComp/SSReflect notations: finite sets over $(X : \text{finType})$ are typed $\{\text{set } X\}$, common notations are: $A :&: B = A \cap B$, $A :| B = A \cup B$, $\sim : A = A^c$ and $\text{set0} = \emptyset$. Finite support functions from $(X : \text{finType})$ to $(Y : \text{Type})$ are typed $\{\text{ffun } X \rightarrow Y\}$; and for any $(Z : X \rightarrow \text{Type})$, dependent ffuncs, which map any $(x : X)$ to an element of $(Z x)$ are typed $\{\text{ffun forall } x : X, Z x\}$.

```

128
129 Definition focal_element (m : bpa) : pred {set W} := fun A => m A > 0.
130 Definition focalset (m : bpa) : {set {set W}} := [set A | focal_element m A].
131

```

Of course, Bel and Pl can be directly expressed as sums over the focal set:

```

133 Lemma Bel_focalsetE m A : Bel m A = \sum_(B in focalset m | B \subset A) m B.
134 Lemma Pl_focalsetE m A : Pl m A = \sum_(B in focalset m | B :&: A != set0) m B.
135
136

```

Belief functions can be interpreted from the viewpoint of two different uncertainty theories:

DS theory As in our example, knowledge is build from evidences about the actual state of the world ω^* . There are several “parts of beliefs” which all describe a piece of information, they are encoded in the bpa m : the mass $m(B)$ is the degree of confidence on the evidence that support $\omega^* \in B$, but nothing more (see Table 1). Under this interpretation, $\text{Bel}(B)$ (resp. $\text{Pl}(B)$) estimates to what extent the event B is implied by (resp. is compatible with) the knowledge encoded in m — m is seen as a generalized set [7].

ULP theory Our example can also be understood in the perspective of the ULP theory, where Bel and Pl delimit a family of probability measures $\mathcal{F} = \{\text{Pr} \mid \forall A, \text{Bel}(A) \leq \text{Pr}(A) \leq \text{Pl}(A)\}$, which gathers the probability measures that may describe the partial knowledge (also see Table 1). In this interpretation, $\text{Bel}(B)$ (resp. $\text{Pl}(B)$) is the minimal (resp. maximal) value of $\text{Pr}(B)$ in the family.

$A \subseteq \Omega$	\emptyset	$\{P\}$	$\{Q\}$	$\{R\}$	$\{P, Q\}$	$\{P, R\}$	$\{Q, R\}$	$\{P, Q, R\}$
$m(A)$	0	0	0	0.5	0.5	0	0	0
$\text{Bel}(A)$	0	0	0	0.5	0.5	0.5	0.5	1
$\text{Pl}(A)$	0	0.5	0.5	0.5	0.5	1	1	1
$\text{Pr}_x(A)$	0	x	$0.5 - x$	0.5	0.5	$0.5 + x$	$1 - x$	1

Table 1 Prior knowledge from Example 1 — in the DS theory, m directly describes the knowledge, in the ULP theory, m describes a family of probability measures $(\text{Pr}_x)_{x \in [0, 0.5]}$.

Next, we recall a “complexity” definition, which is used to characterize probability measures:

► **Definition 6** (k -additivity). A belief function is said to be k -additive iff the cardinality of its focal elements is at most k .

```

152
153 Definition k_additivity m := \max_(B in focalset m) #|B|.
154 Notation "k '-additive' m" := (k_additivity m <= k) ((*technicalities omitted*)).
155

```

► **Proposition 7** (Probabilitiy). If a belief function is 1-additive, i.e. if all focal elements are singletons, then $\text{Bel} = \text{Pl}$ is a probability measure whose distribution is $x \mapsto m(\{x\})$.

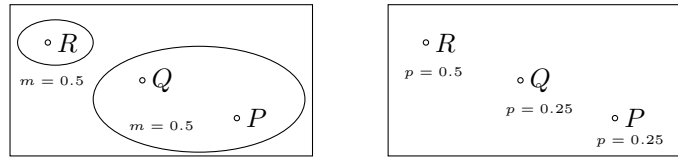
```

158
159 Structure proba := { proba_val :> bpa ; proba_ax : 1.-additive proba_val }.
160 Lemma PrE (p : proba) : Bel p =1 Pl p.
161 Definition dist (p : proba) := fun w => p [set w].
162 Lemma Pr_distE (p : proba) A : Bel p A = \sum_(w in A) dist p w.
163

```

Belief functions thus encompass probabilities. Though the following example makes it clearer how the use of a belief-function model can be “weaker”, and thereby, more expressive.

► **Example 8** (Knowledge representation). In our running example, the prior knowledge is just $m(\{P, Q\}) = m(\{R\}) = 1/2$. It is expressible “as is” using belief functions, but expressing it with a probability distribution requires to make the “equiprobability assumption” (Figure 1).



■ **Figure 1** Prior knowledge expressed by a belief function (left, circles denote focal elements) and the same knowledge expressed by a probability distribution (right, to assign probabilities to singletons, the equiprobability assumption has to be made).

3.2 Conditioning in the Belief Function Theory

Several conditioning rules for belief functions have been proposed, depending on the DS or ULP interpretation, and more generally on the kind of knowledge at work. All those conditionings generalize the Bayesian conditioning, however starting from the same “prior” bpa, they yield distinct “posteriors”: they indeed capture distinct operations such as knowledge revision, updating, focusing [8, 6, 13].

To formalize conditional events $(\cdot \mid C)$ — read “given C ” —, a precondition is necessary: on the technical side, it avoids division-by-zero, and on the semantics side, it means one cannot learn that an impossible event holds. Since it depends on the conditioning rule, we abstract this precondition in a predicate `revisable`. Furthermore, every conditioning rule has to verify the `conditioning_axiom` $\text{Bel}(C \mid C^c) = 0$, that is, if one learns that C holds, then one also learns that no evidence for C^c can hold. Formally speaking, we define a `conditioning` structure which encapsulates the precondition, the conditioning algorithm itself (which turns a `revisable` prior in its posterior “given C ”) and a proof of the `conditioning_axiom`:

```

Definition conditioning_axiom (revisable : bpa → pred {set W})
  (cond : ∀ m C, revisable m C → bpa)
  := ∀ m C (HC : revisable m C), Bel (cond m C HC) (∼:C) = 0.
Structure conditioning := { revisable : bpa → pred {set W} ;
  cond_val m C :> revisable m C → bpa ;
  cond_ax : @conditioning_axiom revisable cond_val }.

```

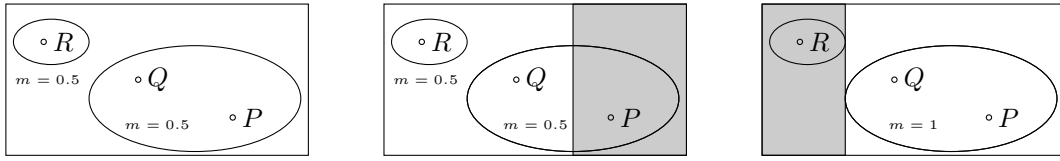
The most common conditioning is the so-called Dempster’s conditioning [4], which captures knowledge revision (i.e. fact learning). In the DS framework, it is understood as a transfer of parts of beliefs: $m(B)$ is transferred to $B \cap C$ if it is not empty, or discarded otherwise (then the posterior had to be renormalized). That is, the evidence now concerns $B \cap C$, the only possible worlds “given that C holds”. In the ULP framework, it is understood as a max-likelihood conditioning: the posterior probability family delimited by $\text{Bel}(\cdot \mid C)$ and $\text{Pl}(\cdot \mid C)$ is the conditioning of those prior probability measures which assign the maximal probability to the event C , that we now know for sure.

```

Definition Dempster_revisable m C := Pl m C != 0.
Definition Dempster_f (m : bpa) (C : {set W})
  := [ffun A : {set W} ⇒ if A == set0 then 0
  else \sum_ (B : {set W}) | (B \in focalset m) && (B :&: C == A)) m B / Pl m C].
Lemma Dempster_bpa_ax m C (HC : Dempster_revisable m C) : bpa_axiom (Dempster_f m C).
Definition Dempster_cond m C (HC : Dempster_revisable m C) : bpa
  := { | bpa_val := Dempster_fun m C ; bpa_ax := Dempster_bpa_ax HC | }.
Lemma Dempster_cond_axiom : @conditioning_axiom Dempster_revisable Dempster_cond.
Definition Dempster_conditioning : conditioning
  := { | cond_val := Dempster_cond ; cond_ax := Dempster_cond_axiom | }.

```

► **Example 9** (Knowledge revision). Dempster’s conditioning is the conditioning approach at work in our example (see [6] for details). Say ω^* denote the actual state of the world, which agents don’t know. Now, consider the case of Agent 1 learning that P is not the murderer: Agent 1 learns that $\omega^* \notin \{P\}$, i.e., $\omega^* \in \{Q, R\}$. In this case, the evidence concerning men now only concerns Q , so from the viewpoint of Agent 1, the knowledge becomes $m(\{Q\}) = m(\{R\}) = 0.5$, thus excluding P without changing the given odds depending on the sex of the murderer (Figure 2, center, the mass assigned to $\{P, Q\}$ was transferred to $\{P, Q\} \cap \{Q, R\} = \{Q\}$). On the contrary, now consider that Agent 2 learns that R is not the murderer: Agent 2 learns $\omega^* \notin \{R\}$, i.e., $\omega^* \in \{P, Q\}$. In this case, the evidence concerning women has to be discarded, so the knowledge becomes $m(\{P, Q\}) = 1$, thus excluding R , and now expressing nothing more than the murderer is a man (Figure 2, right, the mass assigned to $\{R\}$ was discarded since $\{R\} \cap \{P, Q\} = \emptyset$).



■ **Figure 2** Prior (left) and posteriors given $\{Q, R\}$ (center) and given $\{P, Q\}$ (right). White and gray areas denote possible and impossible events – circles denote focal elements.

223

Two other rules have been proposed and called strong and weak conditioning [29]; the former, also known as geometrical conditioning [36], is another “revision” rule; the latter is seldom used since it yields strange results (as for example it may be $\text{Bel}(C \mid C) < 1$)

```

224 Definition Strong_revisable (m : bpa) := fun C : {set W} => Bel m C != 0.
225 Definition Strong_cond (m : bpa) (C : {set W}) (HC : Strong_revisable m C) : bpa.
226   exists (fun A : {set W} =>
227     if (A != set0) && (A \subset C) then m A / Bel m C else 0). (*[...]*)
228 Lemma Strong_cond_axiom : @conditioning_axiom Strong_revisable Strong_cond.
229 Definition Strong_conditioning : conditioning
230   := { | cond_val := Strong_cond ; cond_ax := Strong_cond_axiom |}.
231
232 Definition Weak_revisable (m : bpa) := fun C : {set W} => Pl m C != 0.
233 Definition Weak_cond (m : bpa) (C : {set W}) (HC : Weak_revisable m C) : bpa.
234   exists (fun A => if A :&: C != set0 then m A / Pl m C else 0). (*[...]*)
235 Lemma Weak_cond_axiom : @conditioning_axiom Weak_revisable Weak_cond.
236 Definition Weak_conditioning : conditioning
237   := { | cond_val := Weak_cond ; cond_ax := Weak_cond_axiom |}.
238

```

Another well-known rule is Fagin-Halpern conditioning ([9, 16], already in [4]), which is a “focusing” rule suited for the ULP interpretation. The FH-conditioning of a bpa (which delimits a family of prior probabilities) leads to the bpa delimiting the family of posterior probabilities. At the time we are writing this paper, we did not focused on formalizing this conditioning, even if it could be interesting as a future work.

249 3.3 Informed Decision

A classical way to model decision processes is to use utility functions [32, 25]. Say there are several possible choices (a.k.a. actions); let A denote the set of all these actions. Depending on the actual state of the world $\omega^* \in \Omega$, an action $a \in A$ may lead to different outcomes, so

that the decision maker may be satisfied or not. Let $u : A \times \Omega \rightarrow \mathbb{R}$ be an utility function, assigning to any action $a \in A$ and to any state of the world $\omega \in \Omega$, the degree of satisfaction of the decision maker. Their preferences are captured by “scoring function”, which aggregates all ω ’s, considering of course the knowledge expressed over Ω by a bpa – the action which has the best score is the best action to be chosen (from the decision maker’s point of view). In a probabilistic setting, the classical “scoring function” is the expected utility.

A very common scoring function for belief functions is the Choquet discrete integral, which leads to the minimal expected utility, i.e., the minimal degree of satisfaction the decision maker can expect [33, 12]. Another line of thought is to consider that belief functions are particular cases of capacity measures, and to compute the global merit based on its Choquet value [11, 34]. It is easily defined using the bpa by:

$$\text{CEU}(a) = \sum_{B \in \mathcal{S}_m} m(B) \times \min_{\omega \in B} u(a, \omega) = \sum_{B \in \mathcal{S}_m} m(B) \times f_{u(a, \cdot)}^{\text{CEU}}(B)$$

This expression is a weighted sum indexed by focal elements, and it is correct to use \min since a focal element cannot be empty. The function f_u^{CEU} is formalized as:

```
Definition ceu_fun (u : W -> R) : {set W} -> R
:= fun B => match minS u B with Some r => r | None => 0 end.
```

Another rule, axiomatized by Jaffray [17, 18], is a kind of Hurwicz criterion (i.e., a linear combination over the minimal and maximal utility reached for each focal element) – the coefficient α_B may vary, depending on the minimal and maximal utility values over B .

$$\text{JEU}^\alpha(a) = \sum_{B \in \mathcal{S}_m} m(B) \times \left(\alpha_B \times \min_{\omega \in B} u(a, \omega) + (1 - \alpha_B) \times \max_{\omega \in B} u(a, \omega) \right) = \sum_{B \in \mathcal{S}_m} m(B) \times f_{u(a, \cdot)}^{\text{JEU}^\alpha}(B)$$

```
Definition jeu_fun (alpha : R -> R -> R) (u : W -> R) : {set W} -> R
:= fun B => match minS u B, maxS u B with
| Some rmin, Some rmax => let a := alpha rmin rmax in
a * rmin + (1-a) * rmax
| _, _ => 0 end.
```

Last, in the Transferable Belief Model [37], the decision rule is made by recovering a “pignistic” probability distribution BetP , at the very moment where the decision is made (i.e., the equiprobability assumption is made, but after conditionings, if any). It is also expressible as a weighted sum over focal elements:

$$\text{TBEU}(a) = \sum_{B \in \mathcal{S}_m} m(B) \times \frac{\sum_{\omega \in B} u(a, \omega)}{|B|} = \sum_{B \in \mathcal{S}_m} m(B) \times f_{u(a, \cdot)}^{\text{TBEU}}(B)$$

```
Definition tbeu_fun (u : W -> R) := fun B => \sum_(w in B) u w / #|B|%.R.
```

We express those three scoring functions as weighted sums of some “set-utility function” f . It makes it possible to abstract them by $\text{XEU}(a) = \sum_{B \in \mathcal{S}_m} m(B) \times f_{u(a, \cdot)}^{\text{XEU}}(B)$.

```
Definition XEU (m : bpa) (f : {set W} -> R) := \sum_(B in focalset m) m B * f B.
```

For any two extensionally equal function u and v , f_u^{XEU} and f_v^{XEU} also are extensionally equal. For the sake of the proof, we endow `ceu_fun` (resp. `jeu_fun`, `tbeu_fun`) with a Boolean predicate asserting this property, into the structure `CEU` (resp. `JEU`, `TBEU`).

It can be noted that `CEU`, `JEU` and `TBEU` generalize the expected utility criterion: using a 1-additive bpa (i.e., a probability measure), they all lead to the expected utility value:


```

300
301 Lemma CEU_EU (p : proba) u : XEU (CEU u) p = \sum_w dist p w * u w.
302 Lemma JEU_EU alpha (p : proba) u : XEU (JEU alpha u) p = \sum_w dist p w * u w.
303 Lemma TBEU_EU (p : proba) u : XEU (TBEU u) p = \sum_w dist p w * u w.
304

```

4 Formalization of Several Classes of Games of Complete Information

Game theory aims to model and study multi-agent decision making [25, 26]. In this paper, we focus on strategic (or simultaneous) games: agents make their own decision without knowing others' choices, but where the outcome of the game depends on every agent's choices.

4.1 Games of Complete Information

Let us first consider a situation where there is no uncertainty: each agent knows all the other agents involved, their possible actions and the utility they have depending on the chosen actions. One such CGame is defined by a tuple $G = (I, (A_i, u_i)_{i \in I})$ where I is a finite set of agents, then for each agent i , A_i is the set of their actions and $u_i : A \rightarrow \mathbb{R}$ is an utility function, assigning an utility value to each “profile” $a = (a_1, \dots, a_n) \in A = A_1 \times \dots \times A_n$. Agent i prefers the profile a to a' iff $u_i(a) > u_i(a')$. We formalize such “profiles-for-CGames” ($a \in \prod_{i \in I} A_i$) using MathComp's dependently-typed finite support functions; a CGame is then simply encoded by its utility functions:

```

318
319 Variables (R : numFieldType) (I : finType) (A : ∀ i : I, eqType).
320 Definition cprofile {I : finType} (A : I → finType) := {dffun ∀ i : I, A i}.
321 Definition cgame := cprofile A → I → R.
322

```

One of the most prominent solution concept in game theory is that of Nash equilibria [27], that is, profiles such that no agent has any incentive to move to another action:

```

325
326 Definition change_strategy I A (p : cprofile A) (i : I) (ai : A i) : cprofile A
327   := [ffun j => match boolP (i == j) with
328     | AltTrue h => eq_rect _ A ai _ (eqP h)
329     | AltFalse _ => p j   end].
330 Definition Nash_equilibrium (G : cgame) (a : cprofile A) : bool :=
331   [∀ i : I, [∀ ai : A i, ~ (G a i < G (change_strategy a ai) i)]]].
332 Definition Nash_equilibrium_prop (G : cgame) (a : cprofile A) : Prop :=
333   ∀ i : I, ∀ ai : A i, ~ (G a i < G (change_strategy a ai) i).
334 Lemma Nash_equilibriumP G a :
335   reflect (Nash_equilibrium_prop G a) (Nash_equilibrium G a).
336

```

► **Example 10.** Consider Example 1, but say it is known P is the murderer. It is then captured by the CGame $G = (I, (A_i, u_i)_{i \in I})$ where $I = \{1, 2\}$ is the set of agents, $A_i = \{P_i, Q_i, R_i\}$ is the set of actions of Agent i (choosing P , Q or R) and u_i 's are given in Table 2. In this game, both (Q_1, R_2) and (R_1, Q_2) are Nash equilibria: no agent has any incentive to change.

	P_2	Q_2	R_2
P_1	(0, 0)	(0, 3)	(0, 3)
Q_1	(3, 0)	(2, 2)	(3, 3)
R_1	(3, 0)	(3, 3)	(2, 2)

■ **Table 2** Utility functions of Example 10 (it is known that P is the murderer).

When there is some variability regarding action choices (e.g., for repeated games), it is meaningful to use mixed strategies. A mixed strategy ρ_i of Agent i is a probability over Agent i 's actions, and a mixed strategy profile $\rho = (\rho_1, \dots, \rho_n)$ is a vector of mixed strategies:

```
Definition mixed_cprofile := cprofile (fun i => proba_eqType R (A i)).
```

A mixed strategy profile ρ defines a probability over the set of pure strategy profiles, namely $p_\rho(a) = \prod_{i \in I} \rho_i(a_i)$. We then embed it in a `proba` structure:

```
Definition mk_prod_proba (p : ∀ i : X, proba R (A i)) : {ffun cprofile A → R}
:= [ffun a : cprofile A => \prod_i dist (p i) (a i)].
Definition prod_proba (p : ∀ i : I, proba R (A i)) (i0 : I) : proba R (cprofile A).
```

Last, the utility of a mixed strategy profile is the expected utility w.r.t. the probability over pure strategy profiles, and the notion of Nash equilibria extends straightforwardly:

```
Definition ms_utility (G : cgame R A) (mp : mixed_cprofile) (i : I) : R
:= \sum (p : cprofile A) (dist (prod_proba mp witnessI mp) p) * (G p i).
Definition ms_Nash_equilibrium (G : cgame R A) (mp : mixed_cprofile) : Prop
:= ∀ i : I, ∀ si : proba R (A i),
~ ms_utility G mp i < ms_utility G (change_strategy mp si) i.
```

We can also use the “mixed extension” of a game (i.e., a game where agents’ actions are mixed strategies) and thereby recover previous definitions [19]:

```
Definition mixed_cgame (G : cgame R A) : cgame R (fun i => proba_eqType R (A i))
:= fun mp i => ms_utility G mp i.
Lemma mixed_cgameE G mp i : ms_utility G mp i = (mixed_cgame G) mp i.
Lemma ms_NashE (G : cgame R A) (mp : mixed_cprofile) :
ms_Nash_equilibrium G mp <=> Nash_equilibrium_prop (mixed_cgame G) mp.
```

4.2 Hypergraphical Games

Some CGames can be expressed succinctly as hypergraphical games [28, 40], where the utility is not defined globally, but locally, split in several local games. It produces an hypergraph, where vertices denotes agents and hyperedges denote local games. Formally, a hypergraphical game is defined by a tuple $G = (I, E, (A_i)_{i \in I}, (u_i^e)_{e \in E, i \in e})$, where I is the set of agents, $E \subseteq 2^I$ is the set of local games (in any local game $e = \{i, j, k, \dots\}$, agents i, j, k, \dots are playing), A_i is the set of actions of agent i and $u_i^e : A_e \rightarrow \mathbb{R}$ is the utility function of agent i in the local game e ($A_e = \prod_{i \in e} A_i$ is the set of local profiles corresponding to e 's players). A hypergraphical game with 2-player local games is called a polymatrix.

In our formalization, local games are indexed by the finite type (`localgame : finType`); players playing a local game (`lg : localgame`) are those who verify the boolean predicate (`plays_in lg`). That is, `plays_in` denotes a family of sets of agents.

```
Variables (R : numFieldType) (I : finType) (A : ∀ i : I, finType).
Variables (localgame : finType) (plays_in : localgame → pred I).
```

For any local game, the type of local profiles is defined by:

```
Definition localprof (A : I → eqType) (lg : localgame)
:= {ffun s : {i : I | plays_in lg i}, A (val s)}.
```

In hypergraphical games, each agent chooses one action, and plays it in every local game they are involved in. The global utility of an agent is the sum of the locally obtained utilities:

$u_i(a) = \sum_{e \in E} u_i^e(a_e)$, where $a_e \in A_e$ is the restriction of a to indices of e .

```

396
397 Definition hg_game (u : ∀lg, localprof lg → {i : I & plays_in lg i} → R) : cgame
398   := fun a i ⇒ \sum_(s : {lg : localgame | plays_in lg i})
399     u (tag s) [ffun i ⇒ a (val i)] (exist _ i (tagged s)).
400

```

4.3 Games of Incomplete Information

Harsanyi had proposed [14] a model for decision-making situations where some agents have some uncertainty about other agents, their actions, their utility functions, and more generally about any parameter some agents know and some don't. To model such situations, the partially known parameters are expressed by “types”: each agent i has a “type” θ_i which encodes the parameters' values. Agent i knows (or learns) θ_i , but the other agents may ignore it. Harsanyi defined the model of IGames in a probabilistic setting, hence the so-called Bayesian games. In this setting, a probability measure expresses the knowledge on “type” configurations (it is the frame of discernment). IGames already had been extended to a possibilistic setting by Ben Amor et al. [1], and we extend it to the framework of belief functions (so it generalizes Bayesian games) in the next section.

Since agents know their own type before choosing their action, a pure strategy of Agent i becomes a function $\sigma_i : \Theta_i \rightarrow A_i$ which assigns the chosen action to Agent i 's “type” – Agent i , knowing that θ_i , will play $\sigma_i(\theta_i) \in A_i$. A strategy profile $\sigma = (\sigma_1, \dots, \sigma_n)$ is a vector of such functions, telling which action every agent will play depending on their actual “type”:

```

416
417 Definition iprofile I (T : I → finType) (A : I → eqType)
418   := cprofile (fun i ⇒ [eqType of {ffun T i → A i}]).
419

```

If the actual type configuration is $\theta = (\theta_1, \dots, \theta_n)$, then for any strategy profile σ we denote $\sigma^\theta = (\sigma_1(\theta_1), \dots, \sigma_n(\theta_n))$:

```

422
423 Definition proj_iprofile I (T : I → finType) A (p : iprofile T A)
424   := fun theta ⇒ [ffun i ⇒ p i (theta i)].
425

```

In the sequel, we will also need to “flatten”² an iprofile $T A$ that assigns to any $i \in I$ a strategy $\sigma_i \in (\Theta_i \rightarrow A_i)$, to get a cprofile (fun i_ti : {i : I & T i} => A (val i)), assigning to any dependent pair (i, θ_i) the strategy $\sigma_i(\theta_i) \in A_i$:

```

429
430 Definition iprofile_flatten I (T : I → finType) A (p : iprofile T A)
431   := [ffun i_ti ⇒ p (projT1 i_ti) (projT2 i_ti)].
432

```

In Bayesian games, the utility of Agent i having the “type” θ_i is given by the expected utility w.r.t. the conditioned probability distribution “given θ_i ”. If one considers mixed strategies, then the two probability distributions (over type configurations and pure strategy profiles) are merged, so the utility becomes the expectation over the joint distribution.

5 Bel-Games

Bel games are IGames where the knowledge is expressed by a belief function – so it generalizes Bayesian games. Mathematically, a Bel game is defined by a tuple $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$:

- I is the finite set of agents;

² For example, in an IGame with two players and two types per agents, $\sigma = (\sigma_1, \sigma_2)$ is flattened to $(\sigma_1(\theta_1), \sigma_1(\theta'_1), \sigma_2(\theta_2), \sigma_2(\theta'_2))$

- 441 ■ A_i is the set of actions of Agent i ;
- 442 ■ Θ_i is the finite set of “types” of Agent i
- 443 ■ $u_i : A \times \Theta \rightarrow \mathbb{R}$ is the utility function of Agent i , it depends on the joint action
- 444 $(a_1, \dots, a_n) \in A$ and on the type configuration; $(\theta_1, \dots, \theta_n) \in \Theta$
- 445 ■ $m : 2^\Theta \rightarrow [0, 1]$ is a bpa which describes the prior knowledge on type configurations.

446 The utility of Agent i having the “type” θ_i is the XEU value w.r.t. the posterior knowledge
 447 “given θ_i ” (i.e., given the event $\{\theta' \mid \theta'_i = \theta_i\} \subseteq 2^\Theta$). That is, the utility depends on both the
 448 conditioning and the decision rule. For instance, it may be:
 449 $\text{CEU}_{(i, \theta_i)}(\sigma) = \sum_{B \subseteq 2^\Theta} m(B \mid \theta_i) \times \min_{\theta' \in B} u(\sigma(\theta'), \theta')$.

450 ► **Example 11 (Bel game).** We now are able to express Example 1 by a Bel game $G =$
 451 $(I, (A_i, \Theta_i, u_i)_{i \in I}, m)$. The set of agents is $= \{1, 2\}$, their action sets are $A_i = \{P_i, Q_i, R_i\}$.
 452 Agent 1 will learn either that P is the murderer ($\omega^* \in \{P\}$) or that he is not ($\omega^* \in \{Q, R\}$):
 453 Agent 1’s “type” set is $\Theta_1 = \{P, \bar{P}\}$. Similarly, Agent 2 will learn either that R is the murderer
 454 ($\omega^* \in \{R\}$) or that she is not ($\omega^* \in \{P, Q\}$), so $\Theta_2 = \{R, \bar{R}\}$. The knowledge is expressed over
 455 $\Theta = \Theta_1 \times \Theta_2$: since $(P, \bar{R}) \equiv P$, $(\bar{P}, \bar{R}) \equiv Q$, $(\bar{P}, R) \equiv R$ and (P, R) is impossible, the knowledge
 is $m(\{(P, \bar{R}), (\bar{P}, \bar{R})\}) = m(\{(\bar{P}, \bar{R})\}) = 0.5$. Finally, utility functions are given in Table 3.

	P_2	Q_2	R_2		P_2	Q_2	R_2		P_2	Q_2	R_2
P_1	(0, 0)	(0, 3)	(0, 3)	P_1	(2, 2)	(3, 0)	(3, 3)	P_1	(2, 2)	(3, 3)	(3, 0)
Q_1	(3, 0)	(2, 2)	(3, 3)	Q_1	(0, 3)	(0, 0)	(2, 2)	Q_1	(3, 3)	(2, 2)	(3, 0)
R_1	(3, 0)	(3, 3)	(2, 2)	R_1	(3, 3)	(3, 0)	(2, 2)	R_1	(0, 3)	(0, 3)	(0, 0)

■ **Table 3** Utility functions of Example 11 for $\theta = (P, \bar{R})$ (left: P is the murderer), $\theta = (\bar{P}, \bar{R})$ (center, Q is the murderer) and $\theta = (\bar{P}, R)$ (right: R is the murderer). The configuration $\theta = (P, R)$ is impossible

456

457 ► **Example 12 (lGame’s strategy).** Let $\sigma = (\sigma_1, \sigma_2)$ be defined by $\sigma_1(P) = Q_1$, $\sigma_1(\bar{P}) = P_1$,
 458 $\sigma_2(R) = Q_2$, $\sigma_2(\bar{R}) = R_2$. σ is a pure strategy encoding that Agent 1 will choose Q when
 459 learning that P is the murderer, and choose P otherwise, and that Agent 2 will choose Q
 460 when learning that R is the murderer, and choose R otherwise. Since no agent, whatever is
 461 their type, has incentive to move to another strategy, σ is a Nash equilibrium.

462 Formally speaking, a Bel-Game is fully defined by two elements: the prior knowledge and
 463 agents’ preferences, that is, the bpa and the utility functions. The other parameters (types
 464 for agents I , their actions A and their “types” T) are just parameters of this pair:

```
465 Definition belgame (I : finType) (A : I → eqType) (T : I → finType) :=
466   (bpa R (cprofile T) * (cprofile A → cprofile T → agent → R))%type.
```

469 For the sake of readability, we introduce two short-hands: **Tconfig**, the type for “type”
 470 configurations; and **event_ti** := $\theta_i \mapsto \{\theta' \mid \theta'_i = \theta_i\}$:

```
471 Notation Tconfig := [finType of {dffun ∀ i : I, T i}].
472 Definition event_ti i (ti : T i) := [set t : Tconfig | t i == ti].
```

475 In a Bel-game, agents know their own “type” but may ignore others’ ones. For them to
 476 reason and evaluate their utility, they will need to consider all possible “types” of other
 477 players, and their respective knowledge. So, the prior has to be **revisable** w.r.t. all possible
 478 “types” – what we call a proper Bel-Game. We thus define the following Boolean predicate:

```

479
480 Definition proper_belgame A T (G : belgame A T) (cond : conditioning R Tconfig)
481 := [∀ i : agent, [∀ ti : T i, revisable cond G.1 (event_ti ti)]]].

```

In a proper Bel-game, the utility of a given iprofile for Agent i whose “type” is θ_i , is the XEU value of this iprofile w.r.t. the revised knowledge:

```

485 Definition belgame_utility A T (G : belgame A T) (cond: conditioning R Tconfig)
486 (xeu : xeu_box R Tconfig) (HG : proper_belgame G cond) (p : iprofile T A) (i :
487 agent) (ti : T i) : R
488 := let kn := cond G.1 (event_ti ti) (is_revisable HG ti) in
489 XEU kn (xeu (fun t => G.2 (proj_iprofile p t) t i)).
490
491

```

For Bel-games, the definition of Nash equilibria holds: an iprofile is a Nash equilibrium iff no agent with whatever “type”, has any incentive to change strategy:

```

494 Definition BelG_Nash_equilibrium_prop A T (G : belgame A T) (cond : conditioning R
495 Tconfig) (u : xeu_box R _) (H : proper_belgame G cond) (p : iprofile T A)
496 := ∀ i : I, ∀ ti : T i, ∀ ai : A i,
497 ~ (belgame_utility u H p ti < belgame_utility u H (change_istrategy p ti ai) ti).
498
499

```

6 Howson-Rosenthal-like transforms

Howson–Rosenthal’s theorem asserts the correctness of a transform, which casts a 2-player Bayesian game into an equivalent polymatrix (of complete information) [15]. This makes it possible to benefit from both theoretical and algorithmic results of classical game theory.

In the following, we formally define and prove the correctness of three Howson-Rosenthal-like transforms that we had devised in previous work [10, 30]. All these transforms cast n -player Bel-games into hypergraphical games; the produced games all have the same utility values, though different hypergraphs. These transforms can be applied depending on the conditioning and on the decision rule (cf. Table 4).

Transform	Conditioning	XEU	Space	Time
Direct transform	Dempster’s c.	any	$O(k \times \text{Size}(G)^k)$	$O(k \times \text{Size}(G)^k)$
Conditioned transform	any	any	$O(k \times \text{Size}(G)^k)$	$O(k \times \text{Size}(G)^k)$
TBM transform	any	TBEU	$O(k \times \text{Size}(G))$	$O(\text{Size}(G))$

Table 4 Transforms, conditioning and XEU they are suited for, and their worst-case complexity w.r.t the k -additivity of the bpa and the size of the input Bel-game (from [30])

The three transforms all follow the same approach: starting from a Bel game G , we construct the equivalent hypergraphical game \tilde{G} , the vertices of which represent every “type” of every agent, denoted by dependent pairs (i, θ_i) , with action set A_i . Its local games correspond to focal elements, so we benefit the hypergraphical game structure to compute an XEU (recall that global utility is the sum of local utilities, as XEU value is the weighted sum of utility w.r.t. focal elements). For all those transforms, let G be the input Bel-game and \tilde{G} ’s vertices be the dependent pairs (i, θ_i) with action set A_i :

```

516 Variables (R : realFieldType) (I : finType) (A : I → eqType) (T : I → finType)
517 (G : belgame R A T).
518
519 Definition HR_agent : finType := [finType of {i : I & T i}].
520
521 Definition HR_action (i_ti : HR_agent) : eqType := A (projT1 i_ti).

```

6.1 Direct transform

The direct transform holds only for Dempster's conditioning, which is made on-the-fly on prior focal elements. It is suited for any XEU.

Starting from a Bel Game, we construct a local game g_B for each prior focal element B . Vertex (i, θ_i) plays in B iff θ_i is possible in B , that is, if $\exists \theta' \in B, \theta_i = \theta'_i$. Its local utility in g_B is the “part of XEU” computed over B' , the subset of B on which the mass shall be transferred during Dempster's conditioning. Formally:

► **Definition 13** (Direct transform of a Bel game). *The direct transform of a Bel game $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$ is the hypergraphical game $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{(i, \theta_i)})_{(i, \theta_i) \in \tilde{I}}, (\tilde{u}_{(i, \theta_i)}^e)_{e \in \tilde{E}, (i, \theta_i) \in e})$:*

- $\tilde{I} = \{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i\},$
- $\tilde{E} = (e_B)_{B \subseteq \mathcal{S}_m}, \text{ where } e_B = \{(i, \theta_i) \mid \theta \in B, i \in I\},$
- $\tilde{A}_{(i, \theta_i)} = A_i,$
- For each $e_B \in \tilde{E}, (i, \theta_i) \in e_B$ and $\tilde{\sigma} \in \tilde{A}$, let $\tilde{v}_i^{\tilde{\sigma}}(\theta) = u_i(\tilde{\sigma}_\theta, \theta)$ in:
 $\tilde{u}_{(i, \theta_i)}^e(\tilde{\sigma}_{e_B}) = m(B) \times f_{\tilde{v}_i^{\tilde{\sigma}}}^{\text{XEU}}(B \cap \{\theta' \mid \theta'_i = \theta_i\}) / \text{Pl}(\{\theta' \mid \theta'_i = \theta_i\}).$

First, let G be a proper Bel-game w.r.t. Dempster's conditioning, xeu be any of the XEU:

```
Variable (proper_G : proper_belgame G (Dempster_conditioning R Tconfig))
        (xeu : XEU R Tconfig).
```

Then let \tilde{G} 's local games be indexed by focal elements, that is, sets of type configurations. A vertex (i, θ_i) plays in the local game g_B iff θ_i is possible in B :

```
Definition HRdirect_localgame := [finType of {set Tconfig}].
Definition HRdirect_plays_in (lg : HRdirect_localgame) (i_ti : HR_agent) : bool
:= [exists t : Tconfig, [&& t \in lg & t (projT1 i_ti) == projT2 i_ti]].
```

Then, the local utility functions are expressed using `HRcond_mkprofile` bookkeeping function, which constructs from a local profile p and a “type” configuration θ the `cprofile` $(p_{(1, \theta_1)}, \dots, p_{(n, \theta_n)})$. Local utility in a local game g_B is the part of the XEU computed from the prior focal element B . Note that Dempster's conditioning transfers masses from B to $B \cap \{\theta' \mid \theta'_i = \theta_i\} = B \cap (\text{event_ti } \theta_i)$, so the local utility correspond to an on the fly Dempster's conditioning. The resulting HG game is finally build from local utility functions:

```
Definition event_ti i (ti : T i) := [set t : Tconfig | t i == ti].
Definition HRdirect_mkprofile lg i_ti (Hi_ti : HRdirect_plays_in lg i_ti)
  (p : HRdirect_localprof lg) (t : Tconfig) : profile.
Definition HRdirect_u : ∀ lg, HRdirect_localprof lg → HRdirect_localagent lg → R
:= fun lg p x => let (i_ti, Hi_ti) := x in let (i, ti) := i_ti in
  G.1 lg * xeu R _ (fun t => G.2 (HRdirect_mkprofile Hi_ti p t) t i)
  (lg :&: (event_ti ti)) / Pl G.1 (event_ti ti).
Definition HRdirect_transform: cgame R HR_action := hg_game HRdirect_u.
```

Finally, we prove the correctness of this transform: it preserves utility values, thus also preserves Nash equilibria. For any `iprofile` σ , $\text{XEU}_{(i, \theta_i)}(\sigma) = \tilde{u}_{(i, \theta_i)}(\text{iprofile_flatten}(\sigma))$:

```
Theorem HRdirect_transform_correct (i : I) (ti : T i) (p : iprofile T A) :
  belgame_utility xeu proper_G p ti
= HRdirect_transform (iprofile_flatten p) (existT _ i ti).
Theorem HRdirect_eqNash (p : iprofile T A) :
  BelG_Nash_equilibrium_prop xeu proper_G p
<> Nash_equilibrium_prop HRdirect_transform (iprofile_flatten p).
```

6.2 Conditioned transform

The conditioned transform holds for any conditioning and XEU.

Starting from a Bel Game G , all the conditioning “given θ_i ” are pre-computed, let \mathcal{S}^* be the union of all posterior focal sets (i.e. the set of all possible focal elements given any θ_i). Each $B \in \mathcal{S}^*$ leads to a local game. As for the direct transform, a vertex (i, θ_i) plays in g_B if θ_i is possible in B . Its utility in g_B is the part of XEU computed over the posterior focal element B . Note that (i, θ_i) ’s local utility in B may be 0, if B is not focal in the posterior “given θ_i ”. Formally:

► **Definition 14** (Conditioned transform). *The conditioned transform of a Bel game $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$ is the hypergraphical game $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{(i, \theta_i)})_{(i, \theta_i) \in \tilde{I}}, (\tilde{u}_{(i, \theta_i)}^e)_{e \in \tilde{E}, (i, \theta_i) \in e})$:*

- $\tilde{I} = \{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i\}$,
- $\tilde{E} = (e_B)_{B \in \mathcal{S}^*}$, where $e_B = \{(i, \theta_i) \mid \theta \in B, i \in I\}$,
- $\tilde{A}_{(i, \theta_i)} = A_i$,
- For each $e_B \in \tilde{E}$, $(i, \theta_i) \in e_B$ and $\tilde{\sigma} \in \tilde{A}$, let $\tilde{v}_i^{\tilde{\sigma}}(\theta) = u_i(\tilde{\sigma}_\theta, \theta)$ in $\tilde{u}_{(i, \theta_i)}^{e_B}(\tilde{\sigma}_{e_B}) = m(B \mid \theta_i) \times f_{\tilde{v}_i^{\tilde{\sigma}}}^{\text{XEU}}(B)$.

First let `xeu` be any XEU, `cond` any conditioning and G be a proper belgame w.r.t `cond`:

```
Variables (xeu: XEU R Tconfig) (cond : conditioning R Tconfig)
          (proper_G: proper_belgame cond).
```

After similar definitions for `HRcond_localgame` and `HRcond_plays_in`, let:

```
Definition HRcond_u : ∀ lg, HRcond_localprof lg → HRcond_localagent lg → R
:= fun lg p x => let (i_ti, Hi_ti) := x in let (i, ti) := i_ti in
  let kn := cond G.1 (event_ti ti) (is_revisable proper_G ti) in
  kn lg * xeu (fun t => G.2 (HRcond_mkprofile Hi_ti p t) t i) lg.
Definition HRcond_transform : cgame R HR_action := hg_game HRcond_u.
```

Finally, we prove the correctness of the conditioned transform:

```
Theorem HRcond_transform_correct (i : I) (ti : T i) (p : iprofile T A):
  belgame_utility proper_G p ti
= HRcond_transform (iprofile_flatten p) (existT _ i ti).
Theorem HRcond_eqNash (p : iprofile T A),
  BelG_Nash_equilibrium_prop xeu proper_G p
<=> Nash_equilibrium_prop (HRcond_transform) (iprofile_flatten p).
```

6.3 TBM-transform

The TBM-transform is designed for the Transferable Belief Model [37], in which knowledge is first revised using Dempster’s conditioning, then at the moment of deciding, a pignistic probability distribution `BetP` is deduced from the posterior `bpa` (according to the equiprobability assumption) and the decision is made according to the Expected Utility criterion. Here we benefit `BetP`’s 1-additivity to produce a low-complexity hypergraph: local games correspond to single states of the world, that is, they all involve only n players.

► **Definition 15** (TBM- transform). *The TBM transform of a Bel game $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$ is the hypergraphical game $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{(i, \theta_i)})_{(i, \theta_i) \in \tilde{I}}, (\tilde{u}_{(i, \theta_i)}^e)_{e \in \tilde{E}, (i, \theta_i) \in e})$:*

- $\tilde{I} = \{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i\}$,
- $\tilde{E} = [Players(\{\theta\}) \mid \theta \in \Theta]$,

622 $\tilde{A}_{(i,\theta_i)} = A_i,$
 623 $\tilde{u}_{(i,\theta_i)}^{\theta}(\tilde{\sigma}_e) = \text{BetP}_{(i,\theta_i)}(\theta) \times u_i(\sigma(\theta), \theta).$

624 First let `cond` be a conditioning and G be a proper Bel-game w.r.t. `cond`. \tilde{G} 's local games
 625 are indexed by “type” configurations, and (i, θ_i) plays in $g_{\theta'}$ iff $\theta_i = \theta'_i$:

```
626 Variables (cond : conditioning R Tconfig) (proper_G: proper_belgame cond).
627 Definition HRTBM_localgame : finType := Tconfig.
628 Definition HRTBM_plays_in : HRTBM_localgame -> pred HR_agent
629 := fun lg i_ti => lg (projT1 i_ti) == projT2 i_ti.
630
631
```

632 Local utility is computed w.r.t. the “pignistic” distribution `BetP`:

```
633 Definition HRTBM_u : ∀ lg, HRTBM_localprof lg -> HRTBM_localagent lg -> R
634 := fun lg p x => let (i_ti, _) := x in let (i, ti) := i_ti in
635   let betp := BetP (cond G.1 (event_ti ti) (is_revisable proper_G ti)) in
636   dist betp lg * G.2 (HRTBM_mkprofile p) lg i.
637 Definition HRTBM_transform : cgame R HR_action := hg_game HRTBM_u.
638
639
```

640 Finally, we prove the correctness of the TBM transform:

```
641 Theorem HRTBM_transform_correct (i : I) (ti : T i) (p : iprofile T A) :
642   belgame_utility (TBEU _ _) proper_G p ti
643   = HRTBM_transform [ffun j_tj => p (projT1 j_tj) (projT2 j_tj)] (existT _ i ti).
644 Theorem HRTBM_eqNash (p : iprofile T A),
645   BelG_Nash_equilibrium_prop (TBEU _ _) proper_G p
646   <=> Nash_equilibrium_prop HRTBM_transform (iprofile_flatten p).
647
648
```

649 7 Conclusion and Perspectives

650 In addition to the “Bel game” model and the three transforms definition, the main contribution
 651 of this article is its formal verification in the COQ proof assistant. On the one hand, it
 652 provides strong guaranties on the correctness of the results, and allow to identify subtleties
 653 that were left implicit on the paper proof. Moreover, it helps to improve the proofs, both
 654 in their flow and in their prose. On the other hand, we provide a structured library for belief
 655 functions and hypergraphical games, as for some generic lemmas which we aim to propose
 656 for integration in MathComp/SSReflect library.

657 This work opens several research directions, both on the theoretic side and on the formal
 658 verification side. On one side, we aim at extending this result to other decision-theoretic
 659 approaches, as for instance partially-ordered utility aggregations for belief function and other
 660 non-additive-measure approaches (Choquet capacities of order 2, RDU). On the other side,
 661 we aim to formalize Fagin-Halpern’s conditioning rule, as for the Moebius inverse and more
 662 generally the points we left aside as they were not necessary for this result. Furthermore we
 663 like to encompass this work into a larger library on decision under uncertainty, as it helps
 664 model extension as much as reusability.

665 References

- 666 1 Nahla Ben Amor, Hélène Fargier, Régis Sabbadin, and Meriem Trabelsi. Possibilistic Games
 667 with Incomplete Information. In *Proceedings of the Twenty-Eighth International Joint Confer-*
 668 *ence on Artificial Intelligence*, pages 1544–1550. ijcai.org, 2019.
- 669 2 Alexander Bagnall, Samuel Merten, and Gordon Stewart. A Library for Algorithmic Game
 670 Theory in SSReflect/Coq. *Journal of Formalized Reasoning*, 10(1):67–95, 2017.

- 671 3 The Coq Development Team. *The Coq Proof Assistant: Reference Manual: version 8.12*, 2020.
672 URL: <https://coq.github.io/doc/V8.12.2/refman/>.
- 673 4 Arthur P. Dempster. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The*
674 *Annals of Mathematical Statistics*, 38:325–339, 1967.
- 675 5 Christoph Dittmann. Positional determinacy of parity games. Available at [https://www.](https://www.isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf)
676 [isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf](https://www.isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf), 2016.
- 677 6 Didier Dubois and Thierry Denoeux. Conditioning in Dempster-Shafer Theory: Prediction vs.
678 Revision. In *Belief Functions: Theory and Applications - Proceedings of the 2nd International*
679 *Conference on Belief Functions*, pages 385–392. Springer, 2012.
- 680 7 Didier Dubois and Henri Prade. On the Unicity of Dempster Rule of Combination. *International*
681 *Journal of Intelligent Systems*, 1(2):133–142, 1986.
- 682 8 Didier Dubois and Henri Prade. Focusing vs. belief revision: A fundamental distinction when
683 dealing with generic knowledge. In *Qualitative and quantitative practical reasoning*, pages
684 96–107. Springer, 1997.
- 685 9 Ronald Fagin and Joseph Y. Halpern. A New Approach to Updating Beliefs. In *Proceedings*
686 *of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 347–374, 1990.
- 687 10 Hélène Fargier, Érik Martin-Dorel, and Pierre Pomeret-Coquot. Games of incomplete in-
688 formation: A framework based on belief functions. In Jirina Vejnarová and Nic Wilson,
689 editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European*
690 *Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceed-*
691 *ings*, volume 12897 of *Lecture Notes in Computer Science*, pages 328–341. Springer, 2021.
692 doi:10.1007/978-3-030-86772-0_24.
- 693 11 Itzhak Gilboa. Expected Utility with Purely Subjective Non-Additive Probabilities. *Journal*
694 *of Mathematical Economics*, 16(1):65–88, 1987.
- 695 12 Itzhak Gilboa and David Schmeidler. Maxmin Expected Utility with Non-Unique Prior.
696 *Journal of Mathematical Economics*, 18(2):141–153, 1989.
- 697 13 Ruobin Gong and Xiao-Li Meng. Judicious judgment meets unsettling updating: Dilation,
698 sure loss and simpson’s paradox. *Statistical Science*, 36(2):169–190, 2021.
- 699 14 John C Harsanyi. Games with Incomplete Information Played by “Bayesian” Players, I–III.
700 Part I. The Basic Model. *Management Science*, 14(3):159–182, 1967.
- 701 15 Joseph T Howson Jr and Robert W Rosenthal. Bayesian Equilibria of Finite Two-Person
702 Games with Incomplete Information. *Management Science*, 21(3):313–315, 1974.
- 703 16 J-Y Jaffray. Bayesian Updating and Belief Functions. *IEEE Transactions on Systems, Man,*
704 *and Cybernetics*, 22(5):1144–1152, 1992.
- 705 17 Jean-Yves Jaffray. Linear Utility Theory for Belief Functions. *Operations Research Letters*,
706 8(2):107–112, 1989.
- 707 18 Jean-Yves Jaffray. Linear Utility Theory and Belief Functions: a Discussion. In *Progress in*
708 *decision, utility and risk theory*, pages 221–229. Springer, 1991.
- 709 19 Rida Laraki, Jérôme Renault, and Sylvain Sorin. *Mathematical foundations of game theory*.
710 Springer, 2019.
- 711 20 Stéphane Le Roux, Érik Martin-Dorel, and Jan-Georg Smaus. An Existence Theorem of
712 Nash Equilibrium in Coq and Isabelle. In *Proceedings Eighth International Symposium on*
713 *Games, Automata, Logics and Formal Verification, GandALF 2017*, volume 256 of *Electronic*
714 *Proceedings in Theoretical Computer Science*, pages 46–60, 2017.
- 715 21 Stéphane Le Roux, Érik Martin-Dorel, and Jan-Georg Smaus. Existence of Nash equilibria
716 in preference priority games proven in isabelle. In *Kurt Gödel Day and Czech Gathering of*
717 *Logicians*, 2021.
- 718 22 Pierre Lescanne and Matthieu Perrinel. "backward" coinduction, nash equilibrium and the
719 rationality of escalation. *Acta Informatica*, 49(3):117–137, 2012.
- 720 23 Assia Mahboubi and Enrico Tassi. *Mathematical Components*. Zenodo, January 2021.
- 721 24 Érik Martin-Dorel and Sergei Soloviev. A Formal Study of Boolean Games with Random
722 Formulas as Payoff Functions. In *22nd International Conference on Types for Proofs and*

- 723 *Programs, TYPES 2016*, volume 97 of *Leibniz International Proceedings in Informatics*, pages
724 14:1–14:22, 2016.
- 725 25 Oskar Morgenstern and John Von Neumann. *Theory of Games and Economic Behavior*.
726 Princeton University Press, 1953.
- 727 26 Roger B Myerson. *Game Theory*. Harvard university press, 2013.
- 728 27 John Nash. Non-Cooperative Games. *Annals of Mathematics*, pages 286–295, 1951.
- 729 28 Christos H. Papadimitriou and Tim Roughgarden. Computing Correlated Equilibria in
730 Multi-Player Games. *Journal of the Association for Computing Machinery*, 55(3):1–29, 2008.
- 731 29 Bernard Planchet. Credibility and Conditioning. *Journal of Theoretical Probability*, 2(3):289–
732 299, 1989.
- 733 30 Pierre Pomeret-Coquot, Hélène Fargier, and Érik Martin-Dorel. Games of Incomplete Inform-
734 ation: a Framework Based on Belief Functions. working paper or preprint, May 2022. URL:
735 <https://hal-univ-tlse3.archives-ouvertes.fr/hal-03658700>.
- 736 31 Stéphane Le Roux. Acyclic preferences and existence of sequential nash equilibria: A formal
737 and constructive equivalence. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and
738 Makarius Wenzel, editors, *Theorem Proving in Higher Order Logics, 22nd International
739 Conference, TPHOLs 2009, Munich, Germany, August 17-20, 2009. Proceedings*, volume
740 5674 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2009. doi:10.1007/
741 978-3-642-03359-9_21.
- 742 32 Leonard J Savage. *The Foundations of Statistics*. Wiley, 1954.
- 743 33 David Schmeidler. Integral Representation without Additivity. *Proceedings of the American
744 Mathematical Society*, 97(2):255–261, 1986.
- 745 34 David Schmeidler. Subjective Probability and Expected Utility without Additivity. *Economet-
746 rica: Journal of the Econometric Society*, pages 571–587, 1989.
- 747 35 Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- 748 36 Philippe Smets. Jeffrey’s Rule of Conditioning Generalized to Belief Functions. In *Uncertainty
749 in artificial intelligence*, pages 500–505. Elsevier, 1993.
- 750 37 Philippe Smets and Robert Kennes. The Transferable Belief Model. *Artificial Intelligence*,
751 66(2):191–234, 1994.
- 752 38 René Vestergaard. A constructive approach to sequential nash equilibria. *Inf. Process. Lett.*,
753 97(2):46–51, 2006. doi:10.1016/j.ipl.2005.09.010.
- 754 39 Peter Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman & Hall, 1991.
- 755 40 Elena Yanovskaya. Equilibrium Points in Polymatrix Games. *Lithuanian Mathematical
756 Journal*, 8:381–384, 1968.