# 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. 2023. hal-03782650v2

HAL Id: hal-03782650
https://ut3-toulouseinp.hal.science/hal-03782650v2
Preprint submitted on 27 Feb 2023

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. publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

## Bel-Games:

# A Formal Theory of Games of Incomplete Information <br> Based on Belief Functions in the Coq Proof Assistant* 

Pierre Pomeret-Coquot $\square$ (1)<br>IRIT, Université de Toulouse III - Paul Sabatier, France<br>Hélène Fargier $\square$ ©<br>IRIT, CNRS, Toulouse, France<br>Érik Martin-Dorel $\square$ (c)<br>IRIT, Université de Toulouse III - Paul Sabatier, France


#### Abstract

- Abstract

Decision theory and game theory are both interdisciplinary domains that focus on modelling and analyzing 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 use the Coq/SSReflect proof assistant to formally prove the results we obtained in [29]. First, we formalize a general theory of belief functions with finite support, and structures and solutions concepts from game theory. On top of that, we extend Bayesian games to the theory of belief functions, so that 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 provide three different proofs of 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 thus enable the use of existing algorithms.


2012 ACM Subject Classification Theory of computation $\rightarrow$ Logic and verification; Theory of computation $\rightarrow$ Type theory; Theory of computation $\rightarrow$ Higher order logic; Theory of computation $\rightarrow$ Algorithmic game theory; Theory of computation $\rightarrow$ Solution concepts in game theory; Theory of computation $\rightarrow$ Representations of games and their complexity

Keywords and phrases Game of Incomplete Information, Belief Function Theory, Coq Proof Assistant, SSReflect Proof Language, MathComp Library

Related Version Paper-and-pencil proof article: doi:10.1016/j.ijar.2022.09.010 [29]
Formal proofs repository: https://github.com/pPomCo/belgames

Funding Pierre Pomeret-Coquot: ANITI, funded by the French "Investing for the Future - PIA3" program under the Grant agreement $n^{\circ}$ ANR-19-PI3A-0004.

[^0]
## 1 Introduction

From a mathematical perspective, measure theory is a fundamental domain to learn and use, notably given its direct application to integration and probability theory. Several works thus focused on formalizing measure theory in type theory, e.g., relying on reference textbooks [14]. Next, probability play a key role in the context of game theory, gathering several multi-agent frameworks that can model situations in many application areas such as economics, politics, logics, artificial intelligence, biology, and so on. In particular, the framework of Bayesian games (a class of games of incomplete information), has been well-studied by the decision theory community [13, 25]. However, using probability and additive measures appears to be unsatisfactory to model subtle decision-making situations with uncertainty.

In this work, we aim to show that the belief function theory also is amenable to formal proof, and makes it possible to formally verify the correctness of three state-of-the-art algorithms. In [11, 29], we introduced the notion of Bel games, which faithfully models games of incomplete information where the uncertainty is expressed within the DempsterShafer theory of belief functions. This framework naturally encompass Bayesian games, as belief functions generalize probability measures. Also, we generalized the Howson-Rosenthal theorem to the framework of Bel games and proposed three transforms which make it possible to cast any Bel game into an equivalent game of complete information (without any uncertainty). Furthermore, these transforms preserve the space complexity of the original Bel game (they produce a game with a succinct representation, corresponding to the class of so-called hypergraphical games).

Contributions. In the present paper, we consolidate the mathematical results previously published in [29], presenting a formal verification of our algorithms using the Coq proof assistant [6]. First, we formalize a general theory of belief functions. Then, we formalize structures and solution concepts for "standard" games, Bayesian games, and Bel games, and we formally prove the correctness of the three transform algorithms, in order to provide strong confidence on these results. The software artifact obtained was released under the MIT license and is available within the official Coq projects OPAM archive. Our formalization effort also resulted in more background lemmas, integrated in the MathComp library. To the best of our knowledge, it is the first time the theory of belief functions is mechanized in a formal proof assistant, and applied to the domain of (formal) game theory of incomplete information.

Related works. Several formalization efforts have been carried out in game theory since 2006, each focusing on a somewhat different fragment: Vestergaard [34] then Le Roux [30], formalizing Kuhn's existence of a Nash equilibrium in finite games in extensive form, using Coq; Lescanne et al. [21], studying rationality of infinite games in extensive form, using Coq; Martin-Dorel et al. [23], studying the probability of existence of winning strategies in Boolean finite games, using Coq; Bagnall et al. [4], formalizing well-known results of algorithmic game theory, using Coq; Dittmann [8], proving the positional determinacy of parity games, using Isabelle/HOL; Le Roux et al. [19], 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 [20]. Game theory aside, numerous works have been carried out in proof assistants to formalize probability and/or measure theory. Regarding the Coq proof assistant, we can mention the recent works by Affeldt et al. (on information theory based on discrete probability theory [2]; and measure theory based on MathComp [1]) and Boldo et al. [5], focusing on Lebesgue's integral theory.

Paper outline. We start by introducing a motivating example, for which the Bayesian
approach fails but the Dempster-Shafer approach succeeds. Then, we incrementally present the Coq formalization of Bel games, giving successively the main mathematical definitions and lemmas and their formal Coq counterpart, throughout the paper: Section 3 presents the Dempster-Shafer theory of belief functions; then Section 4 focuses on complete-information games (including hypergraphical games) while Section 5 deals with Bel games; then Section 6 is devoted to our formalization of Howson-Rosenthal's generalized theorem in the scope of n-player Bel games. Finally, Section 7 gives concluding remarks and perspectives.

## 2 Motivating Example: the Murder of Mr. Jones

- Example 1 (Inspired by the Murder of Mr. Jones, [33J). Two players, named Player 1 and Player 2, are independently looking for an 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 allows 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 $\$ 6 \mathrm{k}$, to be shared between the people making the deal (that is, $\$ 2 \mathrm{k}$ or $\$ 3 \mathrm{k}$ depending on if the players choose the same partner or not); associating with the culprit produces no payoff ( $\$ 0 \mathrm{k}$ ). Moreover, Player 1 is investigating about $P$ and will know whether he is guilty before making the decision. Similarly, Player 2 will know whether $R$ is guilty.

The Bayesian approach (claiming any knowledge shall be described by a single subjective probability) is not well-suited here. Assume Player 1 learns that $P$ is innocent. It should not impact the evidence of $50 \%$ chance per sex, so the probability of guilt should become $1 / 2$ for $Q$ and $1 / 2$ for $R$. However, in a purely Bayesian view, a prior probability must be made explicit (e.g. by equiprobability assumption: $1 / 4$ for $P, 1 / 4$ for $Q$ and $1 / 2$ for $R$ ), so after conditioning the posterior probability would not give $50 \%$ chance per sex anymore: equiprobability and conditioning "given $\neg P$ " yields $1 / 3$ for $Q$ and $2 / 3$ for $R$ : learning that $P$ is innocent would increase the odds against $R$ ! In the sequel, we will reuse this example to highlight how the framework of belief functions better captures uncertain knowledge.

## 3 Formalization of Belief Functions for Mono-Agent Decision Making

Modelling mono-agent decision making under uncertainty involves three main tasks. First, knowledge has to be expressed in a well-suited representation, encoding what is known without making extra assumptions. Then, if the agent may learn or observe some event before the decision, one shall identify the relevant conditioning rule. Finally, the agent's preferences must be captured by a compatible decision rule. In this work, we focus on real-valued utility-based decision rules, which evaluate uncertain outcomes so that the agent prefers outcomes with a bigger score. For example, modelling well-known variable phenomena can perfectly be captured in a probabilistic setting: a probability represents the variability; conditional probability updates knowledge; and preferences over uncertain outcomes may be captured by expected utility. Still, this approach may be unsuccessful to model other kinds of uncertainty.

In the sequel, we rely on belief function theory, which generalizes probability theory and enables capturing both variability and ignorance. In this section, we focus on a single decision maker, while the material from Section 4 will deal with multi-agent decision making.

### 3.1 Belief functions

The theory of belief functions is a powerful toolset from decision theory and statistics. It encompasses two distinct approaches for reasoning under uncertainty: the Dempster-Shafer theory of evidence (DS) [7, 31] and the upper-lower probability theory (ULP) [7, 35]. Both approaches consider a finite set of possible "states of the world" $\Omega=\left\{\omega_{1}, \ldots, \omega_{n}\right\}$, one of which being the actual state of the world $\omega^{*}$, and three functions $m$, Bel and $\mathrm{Pl}: 2^{\Omega} \rightarrow[0,1]$ which all map subsets of $\Omega$ to real numbers. Those functions are deducible one from another. In the DS theory, the mass function $m$ is the basic knowledge about the world ( $m(A)$ is the part of belief supporting the evidence $\omega^{*} \in A$ : it asserts that $A$ holds but cannot be refined into smaller claims). The non-additive continuous measures Bel and Pl indicate how much a proposition is implied by (resp. is compatible with) the knowledge. By contrast, the ULP theory suppose that there is an unknown probability $\operatorname{Pr}^{*}$ which is bounded by Bel and $\mathrm{Pl}: \forall A, \operatorname{Bel}(A) \leq \operatorname{Pr}^{*}(A) \leq \operatorname{Pl}(A)$; then $m$ is just a concise representation of the family of compatible probabilities.

Example 1 can be understood in both theories. In the DS theory, $m(\{P, Q\})=m(\{R\})=$ $1 / 2$ directly encodes the given evidences. In the ULP theory, one rather considers the family of probabilities which satisfy $\operatorname{Pr}(\{P, Q\})=\operatorname{Pr}(\{R\})=1 / 2$ (see Table 1).

| $A \subseteq \Omega$ | $\emptyset$ | $\{P\}$ | $\{Q\}$ | $\{R\}$ | $\{P, Q\}$ | $\{P, R\}$ | $\{Q, R\}$ | $\{P, Q, R\}$ |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $m(A)$ | 0 | 0 | 0 | 0.5 | 0.5 | 0 | 0 | 0 |
| $\operatorname{Bel}(A)$ | 0 | 0 | 0 | 0.5 | 0.5 | 0.5 | 0.5 | 1 |
| $\operatorname{Pl}(A)$ | 0 | 0.5 | 0.5 | 0.5 | 0.5 | 1 | 1 | 1 |
| $\operatorname{Pr}_{x}(A)$ | 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 $\left(\operatorname{Pr}_{x}\right)_{x \in[0,0.5]}$.

In this work, we follow the DS approach and we base our formalization on the mass function $m$. To do so, we use the Coq proof assistant, relying on the MathComp library [22] that provides a comprehensive formalization of finite sets, functions with finite support (endowed with decidable equality) as well as big operators. ${ }^{1}$

- Definition 1 (Frame of discernment). A frame of discernment is a finite set $\Omega$, representing the possible states of the world. One of them is the actual state of the world $\omega^{*}$.
- Definition 2 (Events). $A$ set $A \subseteq \Omega$ is an event which represents the proposition " $\omega^{*} \in A$ ". All set functions we consider ( $m$, Bel, Pl) map events to real numbers within $[0,1]$.

The set $\Omega$ and the carrier of $m$, Bel and Pl are formalized in MathComp by asserting:
Variable (W : finType) (R : realFieldType).

- Definition 3 (Basic probability assignment). A basic probability assignment (bpa), a.k.a. mass function, is a set-function $m: 2^{\Omega} \rightarrow[0,1]$ such that

$$
\begin{equation*}
m(\emptyset)=0 \quad \text { and } \quad \sum_{A \subseteq \Omega} m(A)=1 \quad \text { and } \quad \forall A \subseteq \Omega, m(A) \geq 0 . \quad \text { Formally: } \tag{1}
\end{equation*}
$$

[^1]```
Definition bpa_axiom (m : {ffun {set W} -> R}) :=
    [&& m set0 == 0, \sum_A m A == 1 & [ }\forall\textrm{A},\textrm{m A >= 0]].
Structure bpa := { bpa_val :> {ffun {set W} -> R} ; bpa_ax : bpa_axiom bpa_val }.
```

- Definition 4 (Belief function, plausibility measure). Given a bpa $m$ over $\Omega$, the associated belief function Bel : $2^{\Omega} \rightarrow[0,1]$ and plausibility measure $\mathrm{Pl}: 2^{\Omega} \rightarrow[0,1]$ are defined by:

$$
\operatorname{Bel}(A)=\sum_{B \subseteq A} m(B) \quad \text { and } \quad \operatorname{Pl}(A)=\sum_{B \cap A \neq \emptyset} m(B) . \quad \text { Formally: }
$$

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

- Proposition 1 (Duality). For any $A \subseteq \Omega, \operatorname{Pl}(A)=1-\operatorname{Bel}\left(A^{c}\right)$ and $\operatorname{Bel}(A)=1-\operatorname{Pl}\left(A^{C}\right)$.

Proposition 2 (Super- and sub-additivity). Bel is super-additive while Pl is sub-additive: for disjoint sets $A, B \subseteq \Omega, \operatorname{Bel}(A \cup B) \geq \operatorname{Bel}(A)+\operatorname{Bel}(B)$ and $\operatorname{Pl}(A \cup B) \leq \operatorname{Pl}(A)+\operatorname{Pl}(B)$.

Proposition 3 (Bounds). $0=\operatorname{Bel}(\emptyset)=\operatorname{Pl}(\emptyset) \leq \operatorname{Bel}(A) \leq \operatorname{Pl}(A) \leq \operatorname{Bel}(\Omega)=\operatorname{Pl}(\Omega)=1$, for any $A \subseteq \Omega$. These propositions have been formally proved, with the following statements:

```
Lemma BelE (m : bpa) (A : {set W}) : Bel m A = 1 - Pl m (~:A).
Lemma Bel_super m A B : [disjoint A & B] }->\mathrm{ Bel m (A :|: B) >= Bel m A + Bel m B.
Lemma BelO (m : bpa) : Bel m set0 = 0. (* and likewise for Pl *)
```

Definition 5 (Focal elements, focal set). Given a bpa m over $\Omega$, any subset $A \subseteq \Omega$ with a non-zero mass $m(A)$ is called focal element, and the set of focal elements of $m$ is called the focal set of $m$ and denoted by $\mathcal{S}_{m}$. Formally:

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

- Proposition 4 (Focal elements, focal set). Given a bpa mover $\Omega$, Definition 4 can straightforwardly be rephrased by rewriting the sums over the focal set:
$\operatorname{Bel}(A)=\sum_{\substack{B \in \mathcal{S}_{m} \\ B \subseteq A}} m(B) \quad$ and $\quad \operatorname{Pl}(A)=\sum_{\substack{B \in \mathcal{S}_{m} \\ B \cap A \neq \emptyset}} m(B) . \quad$ Formally:

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

Next, we recall a standard "complexity definition" about belief functions, that will prove useful to characterize probability measures:

Definition 6 ( $k$-additivity). Given a bpa $m$ over $\Omega$ and $k \in \mathbb{N}^{*}$, we say that the bpa $m$ is $k$-additive iff the cardinality of its focal elements is at most $k$. Formally:
Definition $k_{-}$additivity $m:=\backslash \max _{-}(B$ in focalset $m$ ) \#|B|.

- Proposition 5 (Probability measure). Given a bpa $m$ over $\Omega$, if $m$ is 1-additive, i.e. if all focal elements are singletons, then $\mathrm{Bel}=\mathrm{Pl}$ is a discrete probability measure, associated with the distribution $x \in \Omega \mapsto m(\{\omega\})$. Formally:
Structure proba := \{ proba_val :> bpa ; proba_ax : k_additivity proba_val ==1 \}. Lemma PrE (m : proba) (A : \{set W\}) : Bel mA=Pl mA. Definition dist (m : proba) := fun $\mathrm{w} \Rightarrow \mathrm{m}$ [set w]. (*[set w] corresponds to $\{w\} *$ ) Lemma Pr_distE (m : proba) A : Bel m A = \sum_(w in A) dist m w.


### 3.2 Conditioning in the Belief Function Theory

Conditioning is the operation that captures knowledge revision (fact learning) as well as focusing (hypothesis) $[10,9,12]$. By turning a prior bpa into a posterior "given an event $C$ ", one updates the knowledge so it now asserts that $C$ is certain. Several conditioning rules for belief functions have been proposed, depending on the DS or ULP interpretation (cf. Section 3.1) and on the kind of update it involves. Starting from the same prior bpa, they yield distinct posteriors - they indeed capture distinct operations.

Before dealing with conditional events $(\cdot \mid C)$-read "given $C$ "—a precondition happens to be 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 the definition of this precondition is specific to each conditioning rule, we abstract it away in the form of a revisable predicate, which indicates whether an event can be assumed.

- Definition 7 (Conditioning). Given a bpa $m$, a predicate revisable ${ }_{m}: \Omega \rightarrow\{1,0\}$ and an event $C \subseteq \Omega$ such that revisable ${ }_{m}(C)$ holds, a conditioning turns $m$ into the bpa $m\left(\left.\cdot\right|_{\text {cond }} C\right)$ such that $\operatorname{Bel}\left(\left.C^{c}\right|_{\text {cond }} C\right)=0$. Formally:

```
Definition conditioning_axiom (revisable : bpa -> pred {set W})
    (cond : }\forall\textrm{m}C, revisable m C -> bpa) :=
    |m C (Hrev : revisable m C), Bel (cond m C Hrev) (~:C) = 0.
```

In other words, assuming the event $C$ is revisable implies that if one learns that $C$ holds, then one also learns that no evidence for the complement can hold. Next, we formalize a conditioning structure that encapsulates the revisable predicate, the conditioning algorithm itself-which turns a revisable prior in its posterior "given $C$ "-and a proof of the conditioning_axiom:

```
Structure conditioning := { revisable : bpa -> pred {set W} ;
    cond_val :> }\forall\textrm{m}C, revisable m C > bpa ;
    cond_ax : conditioning_axiom cond_val }.
```

The most common conditioning is the so-called Dempster's conditioning [7], which captures knowledge revision (i.e., fact learning). In the DS framework, it is understood as a transfer of parts of beliefs: learning that $C$ holds, $m(B)$ is transferred to $B \cap C$ if it is not empty, or discarded otherwise (then the posterior has to be renormalized due to Equation (1)). That is, the evidence now concerns $B \cap C$, the only possible states of the world "given that $C$ holds". In the ULP framework, it is understood as a max-likelihood conditioning: the posterior probability family delimited by $\operatorname{Bel}(\cdot \mid C)$ and $\operatorname{Pl}(\cdot \mid C)$ is the conditioning of those prior probabilities which assign the maximal probability to event $C$, that we now know for sure.

- Definition 8 (Dempster's conditioning). For any bpa $m$ and any event $C$ such that $\mathrm{Pl}(C) \neq 0$, Dempster's conditioning defines the bpa: $m\left(\left.A\right|_{D} C\right)=\sum_{B \cap C=A \neq \emptyset} m(B) / \operatorname{Pl}(C)$.

```
Definition Dempster_revisable m C := Pl m C != 0.
Definition Dempster_fun (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].
Program Definition Dempster_cond m C (Hrev : Dempster_revisable m C) : bpa :=
    {| bpa_val := Dempster_fun m C ; bpa_ax := _ |}.
Program Definition Dempster_conditioning : conditioning :=
    {| cond_val := Dempster_cond ; cond_ax := _ |}.
```

- Example 2 (Knowledge revision, follow-up of Example 1/Table 1). Dempster's conditioning is the conditioning approach fitting our example (see [9] for details). Suppose e.g. the murderer
is $Q$; Player 1 learns $\omega^{*} \notin\{P\}$, i.e., $\omega^{*} \in\{Q, R\}$. From this viewpoint, the evidence concerning men now only concerns $Q$ : the knowledge becomes $m(\{Q\})=m(\{R\})=0.5$ (Fig. 1, center). Player 2 learns $\omega^{*} \notin\{R\}$, i.e., $\omega^{*} \in\{P, Q\}$. From this viewpoint, the evidence about women is discarded: the knowledge becomes $m(\{P, Q\})=1$ (Fig. 1, right).


Figure 1 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.

Proposition 6 (Dempster's conditioning, Pl). For any bpa $m$ and any event $C$ such that $\operatorname{Pl}(C) \neq 0$, we have $\operatorname{Pl}\left(\left.A\right|_{D} C\right)=\operatorname{Pl}(A \cap C) / \operatorname{Pl}(C)$. Formally:

```
Lemma Demspter_condE m C (Hrev : revisable m C) :
    \forallA, Pl (Dempster_conditioning m C Hrev) A = Pl m (A :&: C) / Pl m C.
```

Two other rules have been proposed and called strong (resp. weak) conditioning [28]; the former, also known as geometrical conditioning [32], is another rule capturing knowledge revision; the latter is seldom used since it yields non-intuitive results (e.g., it may happen that $\operatorname{Bel}(C \mid C)<1)$. We also formalize these two rules below.

- Definition 9 (Strong conditioning). For any bpa $m$ and any event $C$ such that $\operatorname{Bel}(C) \neq 0$, the weak conditioning is defined by the bpa $m\left(\left.A\right|_{S} C\right)=m(A) / \operatorname{Bel}(C)$ if $A \subseteq C$, 0 otherwise.

```
Definition Strong_revisable m C := Bel m C != 0.
Definition Strong_fun (m : bpa) (C : {set W}) := [ffun A : {set W} =>
    if (A != set0) && (A \subset C) then m A / Bel m C else 0].
Program Definition Strong_cond m C (Hrev : Strong_revisable m C) : bpa :=
    {| bpa_val := Strong_fun m C ; bpa_ax := _ |}.
Program Definition Strong_conditioning : conditioning :=
    {| cond_val := Strong_cond ; cond_ax := _ |}.
```

- Proposition 7 (Strong conditioning, Bel). For any bpa $m$ and any event $C$ such that $\operatorname{Bel}(C) \neq 0$, we have $\operatorname{Bel}\left(\left.A\right|_{S} C\right)=\operatorname{Bel}(A \cap C) / \operatorname{Bel}(C)$. Formally:

```
Lemma Strong_condE m C (Hrev : revisable m C) :
    A, Bel (Strong_conditioning m C Hrev) A = Bel m (A :&: C) / Bel m C.
```

- Definition 10 (Weak conditioning). For any bpa $m$ and any event $C$ such that $\operatorname{Pl}(C) \neq 0$, the weak conditioning is defined by the bpa: $m\left(\left.A\right|_{W} C\right)=m(A) / P l(B)$ if $A \cap B \neq \emptyset$, 0 otherwise. Formally:
Definition Weak_revisable m C := Pl m C $!=0$.
Definition Weak_fun (m : bpa) (C : \{set W\}) := [ffun A : \{set W\} $\Rightarrow$
if $A: \&: C \quad!=$ set 0 then $m A / P l m C$ else 0$]$.
Program Definition Weak_cond m C (Hrev : Weak_revisable m C) : bpa :=
\{| bpa_val $:=$ Weak_fun $m C$; bpa_ax $:=$ _ $\mid\}$.
Program Definition Weak_conditioning : conditioning :=
$\{\mid$ cond_val $:=$ Weak_cond ; cond_ax $:=$ _ |\}.
- Proposition 8 (Weak conditioning, Bel). For any bpa $m$ and any event $C$ such that $\operatorname{Pl}(C) \neq 0$, we have $\operatorname{Bel}\left(\left.A\right|_{W} C\right)=(\operatorname{Bel}(A)-\operatorname{Bel}(A \backslash C)) / \operatorname{Pl}(C)$. Formally:

```
Lemma Weak_condE m C (Hrev : revisable m C) :
    \forallA, Bel (Weak_conditioning m C Hrev) A = (Bel m A - Bel m (A :\: C)) / Pl m C.
```


### 3.3 Decision Making with Belief Functions

Consider a single agent decision involving several actions; let $A$ denote the set of all these actions. Also, assume that the outcome of choosing any $a \in A$ is not certain: it may lead to several oucomes depending on the actual state of the world $\omega^{*}$. The agent's preferences on outcomes (which are left implicit here) are expressed by a real-valued utility function $u: A \times \Omega \rightarrow \mathbb{R}: u(a, \omega)>u\left(a^{\prime}, \omega^{\prime}\right)$ would mean the agent prefers the outcome of $a$ when $\omega^{*}=\omega$ to the outcome of $a^{\prime}$ when $\omega^{*}=\omega^{\prime}$. For any action $a$, let $u_{a}: \Omega \rightarrow \mathbb{R}$ denote the partial application of $u: u_{a}$ provides the utility of $a$ depending on the state of the world $\omega$.

Preferences under uncertainty are then defined on $u_{a}$ 's: a relation $u_{a} \succ u_{a^{\prime}}$ would encode the fact the agent prefers $a$ to $a^{\prime}$. In a probabilistic setting, it is meaningful to consider $u_{a}$ 's expectation w.r.t. the probability (hence the name expected utility). Using bpa's, several approaches were defined, each modelling various preferences when facing ignorance. In [29], we analyzed three standard functions that generalize expected utility. They provide real values, and thus lead to completely ordered preferences over actions (since every two actions are directly comparable from their score). We denoted them CEU, JEU, and TBEU, respectively, for Choquet-, Jaffray-, and Transferable Belief-Expected Utility. We have shown they are all expressible as the integration of a particular $\varphi_{u_{a}}^{\mathrm{XEU}}$ function (resp. $\varphi_{u_{a}}^{\mathrm{CEU}}$, $\varphi_{u_{a}}^{\mathrm{JEU}}$, and $\left.\varphi_{u_{a}}^{\mathrm{TBEU}}\right)$ over the powerset $2^{\Omega}$. Those $\varphi_{u_{a}}^{\mathrm{XEU}}$ functinos are themselves parametrized by $u_{a}=\omega \mapsto u(a, \omega)$, that is, by the utility function when $a$ is chosen. As a result, these three scoring functions can be captured by instances of a single higher-order function, which we named XEU.

- Definition 11 (Generalized expected utility). For any bpa m, any utility function $u$ : $A \times \Omega \rightarrow \mathbb{R}$, and any $a \in A$, let us pose $u_{a}=\omega \mapsto u(a, \omega)$. Let $\varphi:(\Omega \rightarrow \mathbb{R}) \rightarrow\left(2^{\Omega} \rightarrow \mathbb{R}\right)$ be a parameter function. We then consider the following generalized expected utility of $a$ :
$\operatorname{XEU}(m)\left(\varphi\left(u_{a}\right)\right)=\sum_{B \in \mathcal{S}_{m}} m(B) \times \varphi\left(u_{a}\right)(B) . \quad$ Formally:

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

Let us review these $\varphi^{\mathrm{XEU}}$ functions, their underlying intuition and formal definition in Coq.
A very common scoring function for belief functions is the Choquet discrete integral (CEU). It models a somehow pessimistic agent. In the ULP interpretation, Bel and Pl delimit a family of probabilities; the CEU computes the minimal expected utility that the family allows. In the DS interpretation, each mass is an evidence supporting an event, for which the CEU only consider its worst-case utility if the considered choice is made.

- Definition 12 (Choquet expected utility). For any bpa m, any utility function $u: A \times \Omega \rightarrow \mathbb{R}$ and any action $a \in A$, the Choquet expected utility of $u_{a}: \Omega \rightarrow \mathbb{R}$ is:

$$
\begin{aligned}
& \operatorname{CEU}(m)\left(u_{a}\right)=\sum_{B \in \mathcal{S}_{m}} m(B) \times \min _{\omega \in B} u_{a}(\omega)=\operatorname{XEU}(m)\left(\varphi^{\mathrm{CEU}}\left(u_{a}\right)\right) \text {, } \\
& \text { with } \varphi^{\mathrm{CEU}}\left(u_{a}\right)(B)=\min _{\omega \in B} u_{a}(\omega) .
\end{aligned}
$$

This expression is a weighted sum indexed by the set of focal elements, which is nonempty: using the min operator is legit. Formally, the functions $\varphi^{\mathrm{CEU}}$ and CEU are defined as follows:

```
Definition fCEU (u_a : W -> R) : {set W} -> R :=
    fun B m match minS u_a B with Some r m r | None = 0 end.
Definition CEU (m : bpa) (u_a : W -> R) := XEU m (fCEU u_a).
```

Another rule, axiomatized by Jaffray [16, 17], is a kind of Hurwicz criterion (i.e., a linear combination over the min. and max. utility reached for each focal element). The parameter coefficients make it possible to locally modulate the pessimism of the modelled agent.

- Definition 13 (Jaffray expected utility). For any bpa m, any utility function $u: A \times \Omega \rightarrow \mathbb{R}$ and any action $a \in A$, the Jaffray expected utility of $u_{a}: \Omega \rightarrow \mathbb{R}$ is parameterized by a family of coefficients $\alpha_{\left(x_{*}, x^{*}\right)} \in[0,1]$ for each possible utility values $x_{*} \leq x^{*}$. For any $B \neq \emptyset$, let us pose $B_{*}=\min _{\omega \in B} u_{a}(\omega)$ and $B^{*}=\max _{\omega \in B} u_{a}(\omega)$. The Jaffray expected utility of $u_{a}$ is:

```
\(\operatorname{JEU}^{\alpha}(m)\left(u_{a}\right)=\sum_{B \in \mathcal{S}_{m}} m(B) \times\left(\alpha_{\left(B_{*}, B^{*}\right)} \times B_{*}+\left(1-\alpha_{\left(B_{*}, B^{*}\right)}\right) \times B^{*}\right)=\operatorname{XEU}(m)\left(\varphi^{\mathrm{JEU}^{\alpha}}\left(u_{a}\right)\right)\),
    with \(\varphi^{\mathrm{JEU}^{\alpha}}\left(u_{a}\right)(B)=\alpha_{\left(B_{*}, B^{*}\right)} \times B_{*}+\left(1-\alpha_{\left(B_{*}, B^{*}\right)}\right) \times B^{*}\). Formally:
Definition fJEU ( \(\alpha: R \rightarrow R \rightarrow R\) ) (u_a : W \(\rightarrow \mathrm{R}\) ) : \{set W \} \(\rightarrow \mathrm{R}:=\)
    fun \(B \Rightarrow\) match minS u_a \(B\), maxS u_a B with
    | Some rmin, Some rmax \(\Rightarrow\) let alp \(:=\alpha\) rmin rmax in alp \(*\) rmin \(+(1-a l p) * r m a x\)
    | _, _ \(\Rightarrow 0\) end.
Definition JEU \(\alpha\) (m : bpa) (u_a : W \(\rightarrow\) R) \(:=\) XEU m (fJEU \(\alpha\) u_a).
```

Last, in the Transferable Belief Model [33], the decision rule is made by recovering a "pignistic" probability distribution ${ }^{2}$ BetP (namely, a probability that serves only for the choice) at the very moment where the decision is made (so, the equiprobability assumption is made, but after conditionings, if any). The score of an action is then the expected utility w.r.t. BetP : $\omega \mapsto \sum_{\substack{B \in \mathcal{S}_{m} \\ \omega \in B}} m(B) /|B|$, that we show to be equivalent to the following definition.

- Definition 14 (Transferable Belief Model expected utility). For any bpa m, any utility function $u: A \times \Omega \rightarrow \mathbb{R}$ and any action $a \in A$, the TBEU of $u_{a}: \Omega \rightarrow \mathbb{R}$ is defined by:

```
    \(\operatorname{TBEU}(m)\left(u_{a}\right)=\sum_{B \in \mathcal{S}_{m}} m(B) \times \sum_{\omega \in B} u_{a}(\omega) /|B|=\operatorname{XEU}(m)\left(\varphi^{\operatorname{TBEU}}\left(u_{a}\right)\right)\)
    with \(\varphi^{\mathrm{TBEU}}\left(u_{a}\right)(B)=\sum_{\omega \in B} u_{a}(\omega) /|B|\). Formally:
```



```
Definition TBEU (m : bpa) (u_a : W \(\rightarrow\) R) := XEU m (fTBEU u_a).
```

- Proposition 9. CEU, JEU, and TBEU all generalize the expected utility criterion: if the bpa $m$ is 1-additive (i.e., it is a discrete probability measure), all these instances of XEU yield the same value as that of the expected utility:

```
Lemma CEU_EU (p : proba) u_a : CEU p u_a = \sum_w dist p w * u w.
Lemma JEU_EU alpha (p : proba) u_a : JEU alpha p u_a = \sum_w dist p w * u w.
Lemma TBEU_EU (p : proba) u_a : TBEU p u_a = \sum_w dist p w * u w.
```

In these formal proofs, the key ingredient is the fact that the criteria satisfy the natural property that $\forall u, \forall \omega \in \Omega, \varphi(u)(\{\omega\})=u(\omega)$.

## 4 Formalization of Several Classes of Games of Complete Information

Game theory is a subdomain of multi-agent decision making [24, 25]. In this paper, we focus on simultaneous games, in which players make their choice (called action or pure strategy) without knowing others' choices in advance; the outcome of an action depends on the choices of other agents. A typical problem amounts to identifying which actions are relevant from the viewpoint of a player, assuming others don't cooperate but strive to increase their own utility. In this section, we consider situations where there is no uncertainty.

[^2]
### 4.1 Games of Complete Information

- Definition 15 (Game of complete information). A CGame is a tuple $G=\left(I,\left(A_{i}, u_{i}\right)_{i \in I}\right)$ where $I$ is a finite set of players; for each Player $i, A_{i}$ is the set of their actions; $u_{i}: A \rightarrow \mathbb{R}$ is an utility function, assigning an utility value to each "action profile", i.e., a vector of actions, also called "pure strategy profile" $a=\left(a_{1}, \ldots, a_{n}\right) \in A=A_{1} \times \cdots \times A_{n}$. Player $i$ prefers the outcome of profile a to that of $a^{\prime}$ iff $u_{i}(a)>u_{i}\left(a^{\prime}\right)$.

We formalize such "profiles-for-CGames" ( $a \in \prod_{i \in I} A_{i}$ ) using MathComp's dependently-typed finite support functions, hence:
Definition cprofile (I : finType) (A : I $\rightarrow$ eqType) $:=\{f f u n \forall i: I, A$ i\}.
Definition cgame (I : finType) (A : I $\rightarrow$ eqType) $:=$ cprofile A $\rightarrow$ I $\rightarrow$ R.
One of the most prominent solution concept in game theory is that of Nash equilibrium [26]:

- Definition 16 (Nash equilibrium). A pure Nash equilibrium is a profile such that no player has any incentive to "deviate". For any pure strategy profile a and any Player $i$, let $a_{-i}$ be the restriction of a to the actions of Players $j \neq i, a_{i}^{\prime}$ an action of Player $i$, then $a_{i}^{\prime} \cdot a_{-i}$ denotes the profile a where the strategy of Player $i$ has been switched to $a_{i}^{\prime}$ (called change_strategy a a'_i in Coq). A profile $a$ is a pure Nash equilibrium iff $\forall i, \forall a_{i}^{\prime}, u_{i}(a) \nless u_{i}\left(a_{i} . a_{-i}\right)$ :

```
Definition change_strategy (p : cprofile A) (i : I) (a'_i : A i) : cprofile A
Definition Nash_equilibrium (G : cgame) (a : cprofile A) : bool :=
    [\foralli : I, [\foralla'_i : A i, ~~ (G a i < G (change_strategy a a'_i) i)]].
```

- Example 3. Consider Example 1 anew; suppose one knows $P$ is the murderer. The situation is captured by the CGame $G=\left(I,\left(A_{i}, u_{i}\right)_{i \in I}\right)$ where $I=\{1,2\}$ is the set of players, $A_{i}=\left\{P_{i}, Q_{i}, R_{i}\right\}$ the set of actions of Player $i$ (choosing $P, Q$ or $R$ ) and the $u_{i}$ 's of Table 2.

|  | $P_{2}$ | $Q_{2}$ | $R_{2}$ |
| :---: | :---: | :---: | :---: |
| $P_{1}$ | $(0,0)$ | $(0,3)$ | $(0,3)$ |
| $Q_{1}$ | $(3,0)$ | $(2,2)$ | $(3,3)$ |
| $R_{1}$ | $(3,0)$ | $(3,3)$ | $(2,2)$ |

Table 2 Utility functions of Example 3 (it is known that $P$ is the murderer). The pair $\left(u_{1}\left(a_{1} \cdot a_{2}\right), u_{2}\left(a_{1} \cdot a_{2}\right)\right)$ is read at the intersection of line $a_{1}$ and column $a_{2}$.

Here, both $\left(Q_{1}, R_{2}\right)$ and ( $R_{1}, Q_{2}$ ) are Nash equilibria.
When there is some variability regarding action choices (e.g., for repeated games), it is meaningful to look for mixed strategies. A mixed strategy $\rho_{i}$ of Player $i$ is a probability over $A_{i}$, and a mixed strategy profile $\rho=\left(\rho_{1}, \ldots, \rho_{n}\right)$ is a vector of mixed strategies:

```
Definition mixed_cprofile := cprofile (fun i => [eqType of proba R (A i)]).
```

A mixed strategy profile $\rho$ defines a probability over the set of pure strategy profiles, namely $p_{\rho}(a)=\prod_{i \in I} \rho_{i}\left(a_{i}\right)$. We package this data in a proba structure:

```
Definition mk_prod_proba (p : \foralli : X, proba R (A i)) : {ffun cprofile A }->\textrm{R}}:
    [ffun a : cprofile A = \prod_i dist (p i) (a i)].
Definition prod_proba (p : \foralli : 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 equilibrium extends straightforwardly:
Definition ms_util (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 := $\forall i$ (si : proba R (A i) ), ~ ms_util G mp i < ms_util G (change_strategy mp si) i.

A standard reduction [18, Def. 4.6.1] amounts to viewing a mixed equilibrium of a game $\left(N,\left(A_{i}, u_{i}\right)_{i \in N}\right)$ as a pure equilibrium in the mixed extension $\left(N,\left(\mathcal{A}_{i}, u_{i}\right)_{i \in N}\right)$, where $\mathcal{A}_{i}$ is the set of mixed strategies over $A_{i}$. Formally:

```
Definition mixed_cgame (G:cgame R A) : cgame R (fun i m [eqType of proba R (A i)])
    := fun mp i = ms_util 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 (mixed_cgame G) mp.
```


### 4.2 Hypergraphical Games

Some games of complete information can be expressed succinctly as hypergraphical games [27, 36], where the utility is not defined globally but locally (namely, split in several "local games"). This yields an hypergraph, where vertices denotes players and hyperedges denote local games. Formally, a hypergraphical game is a tuple $G=\left(I, E,\left(A_{i}\right)_{i \in I},\left(u_{i}^{e}\right)_{e \in E, i \in e}\right)$, where $I$ is the set of players, $E \subseteq 2^{I}$ is the set of local games (in any local game $e=$ $\{a, b, c, \ldots\}$, Players $a, b, c \ldots$ are playing), $A_{i}$ is the set of actions of Player $i$ and $u_{i}^{e}: A_{e} \rightarrow \mathbb{R}$ is the utility function of Player $i$ in the local game $e\left(A_{e}=\prod_{i \in e} A_{i}\right.$ is the set of local profiles related 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); plays_in thus formalizes $E$ as a family of sets of players:

```
Variables (localgame : finType) (plays_in : localgame -> pred I).
```

For any local game 1 g , local profiles are profiles that involve only players which plays_in lg :

```
Definition localprof (lg : localgame) :=
    {ffun }\forall\textrm{s}: {i : I | plays_in lg i}, A (val s)}
```

In hypergraphical games, every player chooses one action, and plays it in every local game they are involved in. The global utility of a player is the sum of the locally obtained utilities: $u_{i}(a)=\sum_{\substack{e \in E \\ i \in e}} u_{i}^{e}\left(a_{e}\right)$, where $a_{e} \in A_{e}$ is the restriction of $a$ to indices of $e$. Thus, an hypergraphical game is a CGame that is specified by its local utility functions:

```
Definition hg_game (u : \foralllg, localprof lg -> {i : I & plays_in lg i} -> R) : cgame
    := fun a i # \sum_(s : {lg : localgame | plays_in lg i})
        u (tag s) [ffun i m a (val i)] (exist _ i (tagged s)).
```


## 5 Bel Games

Harsanyi has proposed [13] a model for decision-making situations where players may have some uncertainty about other players, their actions, their utility functions, or more generally about any parameter of the game. To model such situations, the partially known parameters are expressed by so-called types: ${ }^{3}$ each Player $i$ has a set of possible types $\Theta_{i}$. Each type $\theta_{i} \in \Theta_{i}$ represents a possible parameter describing Player $i$ 's characteristics and knowledge. Every Player $i$ knows (or learns) their own type $\theta_{i} \in \Theta_{i}$ at the time of choosing an action. It may or may not be correlated with other players' types, so it is possible to model players that are not aware of other players' type as well as players with some knowledge about them.

[^3]Harsanyi defined the model of games of incomplete information where players' knowledge is given by a subjective probability and preferences agree with the expected utility: the so-called class of Bayesian games. In this setting, a probability measure expresses the knowledge on type configurations (the frame of discernment being the cartesian product of all players' types, that is, $\left.\Omega=\prod_{i \in N} \Theta_{i}=\Theta\right)$. Games of incomplete information already were extended to a possibilistic setting by Ben Amor et al. [3], and we extend it further to Belief functions (so generalizing both Bayesian games and possibilistic games) [11, 29].
$\rightarrow$ Definition 10 (Bel game). A Bel game [29] is defined by a tuple $G=\left(I,\left(A_{i}, \Theta_{i}, u_{i}\right)_{i \in I}, m\right)$ :

- I is the finite set of players;
- $A_{i}$ is the set of actions of Player $i ; \Theta_{i}$ is the finite set of types of Player $i$;
- $u_{i}: A \rightarrow \Theta \rightarrow \mathbb{R}$ is the utility function of Player $i$; it depends on the joint action $\left(a_{1}, \ldots, a_{n}\right) \in A:=\prod_{i \in I} A_{i}$ and on the type configuration $\left(\theta_{1}, \ldots, \theta_{n}\right) \in \Theta:=\prod_{i \in I} \Theta_{i} ;$ - $m: 2^{\Theta} \rightarrow[0,1]$ is a bpa which describes the prior knowledge.

Formally speaking, a Bel game is fully defined by two elements: the bpa (prior knowledge) and the utility functions (the players' preferences). This pair is parameterized by three types I, the players; A, the family of actions $\left(A_{i}\right)_{i}$; and T , the family of types $\left(\Theta_{i}\right)_{i}$ :

```
Definition belgame (I : finType) (A : I > eqType) (T : I > finType) :=
```

    (bpa R (cprofile \(T\) ) * (cprofile \(A \rightarrow\) cprofile \(T \rightarrow\) player \(\rightarrow R\) )).
    - Example 4 (Bel game). We now are able to endow Example 1 with a Bel game $G=$ $\left(I,\left(A_{i}, \Theta_{i}, u_{i}\right)_{i \in I}, m\right)$. The set of players is $=\{1,2\}$, their action sets are $A_{i}=\left\{P_{i}, Q_{i}, R_{i}\right\}$. Player 1 will learn either that $P$ is the murderer $\left(\omega^{*} \in\{P\}\right)$ or that he is not $\left(\omega^{*} \in\{Q, R\}\right)$ : Player 1's type set is $\Theta_{1}=\{P, \bar{P}\}$. Similarly, Player 2 will learn either that $R$ is the murderer $\left(\omega^{*} \in\{R\}\right)$ or that she is not $\left(\omega^{*} \in\{P, Q\}\right)$, so $\Theta_{2}=\{\mathrm{R}, \overline{\mathrm{R}}\}$. The knowledge is expressed over $\Theta=\Theta_{1} \times \Theta_{2}$ : since $(\mathrm{P}, \overline{\mathrm{R}}) \equiv P,(\overline{\mathrm{P}}, \overline{\mathrm{R}}) \equiv Q,(\overline{\mathrm{P}}, \mathrm{R}) \equiv R$ and $(\mathrm{P}, \mathrm{R})$ is impossible, the knowledge is $m(\{(\mathrm{P}, \overline{\mathrm{R}}),(\overline{\mathrm{P}}, \overline{\mathrm{R}})\})=m(\{(\overline{\mathrm{P}}, \mathrm{R})\})=0.5$. 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 4 for $\theta=(\mathrm{P}, \overline{\mathrm{R}})$ (left, $P$ is the murderer), $\theta=(\overline{\mathrm{P}}, \overline{\mathrm{R}})$ (center, $Q$ is the murderer) and $\theta=(\overline{\mathrm{P}}, \mathrm{R})$ (right, $R$ is the murderer). Configuration $\theta=(\mathrm{P}, \mathrm{R})$ can't occur.

Since players know their own type before choosing their action, a pure strategy of Player $i$ becomes a function $\sigma_{i}: \Theta_{i} \rightarrow A_{i}$ : having the type $\theta_{i}$, Player $i$ will play $\sigma_{i}\left(\theta_{i}\right) \in A_{i}$. Next, a strategy profile $\sigma=\left(\sigma_{1}, \ldots, \sigma_{n}\right)$ is a vector of such functions:

Definition iprofile I A $T:=$ cprofile (fun $i \Rightarrow$ [eqType of $\{f f u n T i \rightarrow A i\}$ ).
If the actual type configuration is $\theta=\left(\theta_{1}, \ldots, \theta_{n}\right)$, then for any strategy profile $\sigma$ we denote by $\sigma^{\theta}=\left(\sigma_{1}\left(\theta_{1}\right), \ldots, \sigma_{n}\left(\theta_{n}\right)\right) \in A$ the action profile that will actually be played:
Definition proj_iprofile I A T (p : iprofile A T) : cprofile A :=
fun theta $\Rightarrow$ [ffun $i \Rightarrow p$ i (theta i)].
In the following, we denote by $u_{i, \sigma}: \Theta \rightarrow \mathbb{R}$ the function mapping states of the world $\omega$ to the corresponding utility of $\sigma$ for Player $i$. It is defined by $u_{i, \sigma}(\theta)=u_{i}\left(\sigma^{\theta}, \theta\right)$.

In Bayesian games, the global utility of a strategy profile $\sigma$ for Player $i$ with type $\theta_{i}$ is the expected utility w.r.t. the conditioned probability distribution "given $\theta_{i}$ ". In Bel games, both expectation and conditioning have to be made explicit, to properly model agents' preferences and knowledge updates. For example, studying a Bel game with Dempster's conditioning and

CEU expectation implies that the utility of a given strategy profile $\sigma$ for Agent $i$ with type $\theta_{i}$ is $\sum_{B \subset \Omega} m\left(\left.B\right|_{D} \theta_{i}\right) \times \min _{\theta^{\prime} \in B} u_{i}\left(\sigma^{\theta^{\prime}}, \theta^{\prime}\right)$. Doing so, we need to ensure that conditioning is meaningful and technically possible, that is, the bpa is revisable given any type of any player. For the sake of readability, we now introduce two shorthands: Tn, representing the set $\Theta$ gathering all type configurations; and event_ti $:=\theta_{i} \mapsto\left\{\theta^{\prime} \in \Theta \mid \theta_{i}^{\prime}=\theta_{i}\right\}$ :

```
Notation Tn := [finType of {dffun \foralli : I, T i}].
Definition event_ti i (ti : T i) := [set t : Tn | t i == ti].
```

A proper Bel games, in which conditioning is safe, shall satisfy the predicate:
Definition proper_belgame A T (G : belgame A T) (cond : conditioning R Tn) : bool $:=[\forall i$ : player, $[\forall \mathrm{ti}: \mathrm{T} i$, revisable cond $G .1$ (event_ti ti)]].

- Definition 11 (Utility in a Bel game). For any Bel game $G=\left(I,\left(A_{i}, \Theta_{i}, u_{i}\right)_{i \in I}, m\right)$, any conditioning cond for which $G$ is proper and any XEU parameter $\varphi^{\mathrm{XEU}}:(\Theta \rightarrow \mathbb{R}) \rightarrow 2^{\Theta} \rightarrow \mathbb{R}$, the utility of the pure strategy profile $\sigma$ for Player $i$ having type $\theta_{i} \in \Theta_{i}$, is the integration of $u_{i, \sigma}=\theta \mapsto u_{i}\left(\sigma^{\theta}, \theta\right)$, i.e., $\operatorname{XEU}\left(m\left(\left.\cdot\right|_{\text {cond }} \theta_{i}\right)\right)\left(\varphi^{\mathrm{XEU}}\left(u_{i, \sigma}\right)\right)$.
Definition belgame_utility A $T$ ( $G$ : belgame A $T$ ) (cond: conditioning $R \mathrm{Tn}$ )
fXEU (HG : proper_belgame G cond) (p : iprofile A T) (i : player) (ti : T i) : R
$:=$ let $\mathrm{kn}:=$ cond $G .1$ (event_ti ti) (is_revisable HG ti) in XEU kn (fXEU (fun $t \Rightarrow G .2$ (proj_iprofile p t) ti)).
Also, for Bel games, the definition of Nash equilibrium applies: an iprofile is a Nash equilibrium iff no player, whatever is this player's type, has any incentive to deviate:

Definition BelG_Nash_equilibrium A $T$ ( $G$ : belgame A $T$ ) (cond : conditioning R Tn) fXEU (H : proper_belgame G cond) (p : iprofile A T) :=
$\forall i: I, \forall t i: T i, \forall a i: A i$,
~ (belgame_utility u H p ti < belgame_utility u H (change_istrategy p ti ai) ti).

- Example 5 (Utility of a strategy). Let $\sigma=\left(\sigma_{1}, \sigma_{2}\right)$ be defined by $\sigma_{1}(\mathrm{P})=Q_{1}, \sigma_{1}(\overline{\mathrm{P}})=P_{1}$, $\sigma_{2}(\mathrm{R})=Q_{2}, \sigma_{1}(\overline{\mathrm{R}})=R_{2} . \sigma$ is a pure strategy asserting that Player 1 will choose $Q$ when learning that $P$ is the murderer, and choose $P$ otherwise, and that Player 2 will choose $Q$ when learning that $R$ is the murderer, and choose $R$ otherwise.
Considering Dempster's conditioning, the Choquet expected utility of $\sigma$ for Player 1 with type $\overline{\mathrm{P}}$ is the integration of $\varphi^{\mathrm{CEU}}\left(u_{i, \sigma}\right)$ w.r.t. the posterior bpa $m(\cdot \mid \overline{\mathrm{P}})$. Recall Example 2, the posterior bpa "given $\overline{\mathrm{P}}$ " has two focal elements: $\{Q\}$ and $\{R\}$, both with mass $1 / 2$. Considering type configurations, those focal elements are $\{(\overline{\mathrm{P}}, \overline{\mathrm{R}})\}$ and $\{(\overline{\mathrm{P}}, \mathrm{R})\}$.

$$
\begin{aligned}
\operatorname{XEU}\left(m\left(\left.\cdot\right|_{D} \overline{\mathrm{P}}\right)\right)\left(\varphi^{\mathrm{CEU}}\left(u_{1, \sigma}\right)\right) & =\sum_{B \in \mathcal{\mathcal { S } _ { m ( \cdot | _ { D } \overline { \mathrm { P } } ) }}} m\left(\left.B\right|_{D} \overline{\mathrm{P}}\right) \times \min _{\theta \in B} u_{1}\left(\sigma^{\theta}, \theta\right) \\
& =0.5 \times u_{1}\left(\left(P_{1}, Q_{2}\right),(\overline{\mathrm{P}}, \mathrm{R})\right)+0.5 \times u_{1}\left(\left(P_{1}, R_{2}\right),(\overline{\mathrm{P}}, \overline{\mathrm{R}})\right)=3 .
\end{aligned}
$$

One may check that for every player and type, $\sigma$ 's CEU equals 3 , the best possible score. Since no player, whatever is their type, has incentive to deviate, $\sigma$ is a Nash equilibrium.

## 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 game (of complete information) [15]. Bayesian games thus benefit from both theoretical and algorithmic results of classical game theory.

In the following, we formally define and prove correct three Howson-Rosenthal-like transforms that we have devised in previous work [11, 29]. All these transforms cast $n$-player Bel games into hypergraphical games; the games so obtained all have the same utility values,
though different hypergraphs. These transforms can be applied safely, depending on the conditioning and on the decision rule (cf. Table 4): Dempster's conditioning is hard-coded into the Direct transform while the TBM transform's low complexity comes from properties of the distribution BetP considered by the TBEU.

| Transform | Conditioning | XEU | Space | Time |
| ---: | :---: | :---: | :---: | :---: |
| Direct transform | Dempster's c. | any | $O\left(k \times \operatorname{Size}(G)^{k}\right)$ | $O\left(k \times \operatorname{Size}(G)^{k}\right)$ |
| Conditioned transform | any | any | $O\left(k \times \operatorname{Size}(G)^{k}\right)$ | $O\left(k \times \operatorname{Size}(G)^{k}\right)$ |
| TBM transform | any | TBEU | $O(k \times \operatorname{Size}(G))$ | $O(\operatorname{Size}(G))$ |

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 (taken from [29]).

The three transforms all follow the same approach: starting from a Bel game $G$, they build the equivalent hypergraphical game $\tilde{G}$, with pairs $\left(i, \theta_{i}\right)$ as "abstract" players (i.e., $\tilde{G}$ 's vertices), denoting every type of every player of $G$. The local games correspond to focal elements, so Player $\left(i, \theta_{i}\right)$ plays in a local game $\lg$ iff the type $\theta_{i}$ is possible in the corresponding focal element. Doing so, we benefit from the hypergraphical game structure to compute an XEU (recall that global utility is the sum of local utilities and that the XEU value is the weighted sum of utilities w.r.t. focal elements). For all those transforms, let ( G : belgame A T) be the input Bel game that has to be turned into a hypergraphical game named $\tilde{G}$. $\tilde{G}$ 's players are pairs $\left(i, \theta_{i}\right)$, their action sets still are $A_{i}$ :

```
Definition HR_player : finType := [finType of {i : I & T i}].
Definition HR_action (i_ti : HR_player) : eqType := A (projT1 i_ti).
```

Strategy profiles of $G$ and of $\tilde{G}$ are in one-to-one correspondance. Every strategy profile $\sigma$ : iprofile A T of $G$, that is, $\sigma: \prod_{i \in I}\left(\Theta_{i} \rightarrow A_{i}\right)$, is flattened to $\tilde{\sigma}$ : cprofile (fun $\mathrm{i}_{\mathrm{t}} \mathrm{ti}:\{\mathrm{i}: \mathrm{I} \& \mathrm{~T}$ i\} $\Rightarrow \mathrm{A}(\mathrm{val} \mathrm{i})) \mathrm{in} \tilde{G}$, that is, $\tilde{\sigma}: \prod_{\left(i, \theta_{i}\right) \in I \times \Theta_{i}} A_{i}$. E.g. in a 2-player game with 2 types per player, $\sigma=\left(\sigma_{1}, \sigma_{2}\right)$ is flattened to $\left(\sigma_{1}\left(\theta_{1}\right), \sigma_{1}\left(\theta_{1}^{\prime}\right), \sigma_{2}\left(\theta_{2}\right), \sigma_{2}\left(\theta_{2}^{\prime}\right)\right)$. This "dependent uncurrying" is performed by the following function:

```
Definition flatten I (T : I }->\mathrm{ finType) A (sigma : iprofile A T) :=
```

    [ffun i_ti \(\Rightarrow\) sigma (projT1 i_ti) (projT2 i_ti)].
    
### 6.1 The Direct Transform

The direct transform only holds for Dempster's conditioning, which is made on-the-fly on prior focal elements. It is suitable for any XEU. Starting from a Bel game $G$, we construct a local game $e_{B}$ for each prior focal element $B$. Vertex $\left(i, \theta_{i}\right)$ plays in $B$ iff $\theta_{i}$ is possible in $B$, that is, if $\exists \theta^{\prime} \in B, \theta_{i}=\theta_{i}^{\prime}$. Its local utility in $e_{B}$ is the "part of XEU" computed over $B^{\prime}$, the subset of $B$ on which the mass shall be transferred during Dempster's conditioning.

- Definition 17 (Direct transform of a Bel game). The direct transform of a Bel game $G=$ $\left(I,\left(A_{i}, \Theta_{i}, u_{i}\right)_{i \in I}, m\right)$ is the hypergraphical game $\tilde{G}=\left(\tilde{I}, \tilde{E},\left(\tilde{A}_{\left(i, \theta_{i}\right)}\right)_{\left(i, \theta_{i}\right) \in \tilde{I}},\left(\tilde{u}_{\left(i, \theta_{i}\right)}^{e}\right)_{e \in \tilde{E},\left(i, \theta_{i}\right) \in e}\right)$ :
- $\tilde{I}=\left\{\left(i, \theta_{i}\right) \mid i \in I, \theta_{i} \in \Theta_{i}\right\}, \tilde{E}=\left(e_{B}\right)_{B \subseteq \mathcal{S}_{m}}, e_{B}=\left\{\left(i, \theta_{i}\right) \mid \theta \in B, i \in I\right\}, \tilde{A}_{\left(i, \theta_{i}\right)}=A_{i}$,
- for each $e_{B} \in \tilde{E},\left(i, \theta_{i}\right) \in e_{B}$ and $\tilde{\sigma} \in \tilde{A}$, let us pose $\tilde{v}_{i}^{\tilde{\sigma}}(\theta)=u_{i}\left(\tilde{\sigma}^{\theta}, \theta\right)$ in:
$\tilde{u}_{\left(i, \theta_{i}\right)}^{e_{B}}\left(\tilde{\sigma}_{e_{B}}\right)=m(B) \times\left(\varphi^{\mathrm{XEU}}\left(\tilde{v}_{i}^{\tilde{\sigma}}\right)\left(B \cap\left\{\theta^{\prime} \mid \theta_{i}^{\prime}=\theta_{i}\right\}\right)\right) / \operatorname{Pl}\left(\left\{\theta^{\prime} \mid \theta_{i}^{\prime}=\theta_{i}\right\}\right)$.
Formally, let $G$ be a proper Bel game w.r.t. Dempster's conditioning and fXEU a $\varphi$ function:
Variable (proper_G : proper_belgame G (Dempster_conditioning R Tn)) (fXEU : \{ffun $\mathrm{Tn} \rightarrow \mathrm{R}\} \rightarrow$ \{ffun $\{$ set Tn$\} \rightarrow \mathrm{R}\}$ ).

Then, let $\tilde{G}$ 's local games be indexed by focal elements, i.e., sets of type configurations:
Definition HRdirect_localgame $:=$ [finType of \{set Tn\}].
A vertex $\left(i, \theta_{i}\right)$ plays in the local game $e_{B}$ iff $\theta_{i}$ is possible in $B$ :

```
Definition HRdirect_plays_in (lg : HRdirect_localgame) (i_ti : HR_player) : bool
    := [\existst : Tn, [&& t \in lg & t (projT1 i_ti) == projT2 i_ti]].
```

Then, local utility functions are given by a function which constructs from a local profile $p$ and a type configuration $\theta$, the cprofile $\left(p_{\left(1, \theta_{1}\right)}, \ldots, p_{\left(n, \theta_{n}\right)}\right)$. This function has type:

```
Definition HRdirect_mkprofile lg i_ti (Hi_ti : HRdirect_plays_in lg i_ti)
    (p : HRdirect_localprof lg) (t : Tn) : profile.
```

Local utility in a local game $e_{B}$ is the part of the XEU computed from the prior focal element B. Note that Dempster's conditioning transfers masses from $B$ to $B \cap\left\{\theta^{\prime} \in \Theta \mid \theta_{i}^{\prime}=\theta_{i}\right\}$ $=B \cap$ (event_ti $\theta_{i}$ ) so the local utility amounts to an on-the-fly Dempster's conditioning. The resulting HG game is finally built from local utility functions:

```
Definition HRdirect_u : \foralllg, HRdirect_localprof lg > HRdirect_localplayer lg > R
    := fun lg p x = let (i_ti, Hi_ti) := x in let (i, ti) := i_ti in
        G.1 lg * fXEU [ffun t = G.2 (HRdirect_mkprofile Hi_ti p t) t i]
                        (lg :&: (event_ti ti)) / Pl G.1 (event_ti ti).
Definition HRdirect : cgame R HR_action := hg_game HRdirect_u.
```

- Theorem 12 (Correctness of the direct transform). For any proper Bel game G, Player $i$ with type $\theta_{i}$, XEU function $\varphi^{\mathrm{XEU}}$, and profile $\sigma$, we have $\mathrm{XEU}\left(m\left(\left.\cdot\right|_{D} \theta_{i}\right)\right)\left(\varphi^{\mathrm{XEU}}\left(u_{i, \sigma}\right)\right)=$ $\tilde{u}_{\left(i, \theta_{i}\right)}(\mathrm{flatten}(\sigma))$. Thence, Nash equilibria of $G$ and $\tilde{G}$ are in one-to-one correspondence:

```
Theorem HRdirect_correct (i : I) (ti : T i) (p : iprofile A T) :
    belgame_utility fXEU properG p ti = HRdirect (flatten p) (existT _ i ti).
Theorem HRdirect_eqNash (p : iprofile A T) :
    BelG_Nash_equilibrium fXEU proper_G p << Nash_equilibrium HRdirect (flatten p).
```


### 6.2 The Conditioned Transform

The conditioned transform holds for any conditioning and XEU. Starting from a Bel game $G$, all the conditioning "given $\theta_{i}$ " 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 $\theta_{i}$ ). Each $B \in \mathcal{S}^{*}$ leads to a local game. As in the direct transform, a vertex $\left(i, \theta_{i}\right)$ plays in $e_{B}$ if $\theta_{i}$ is possible in $B$. Its utility in $e_{B}$ is the part of XEU computed over the posterior focal element $B$. Note that $\left(i, \theta_{i}\right)$ 's local utility in $B$ may be 0 , if $B$ is not focal in the posterior "given $\theta_{i}$ ". Formally speaking:

- Definition 18 (Conditioned transform). The conditioned transform of a Bel game $G=$ $\left(I,\left(A_{i}, \Theta_{i}, u_{i}\right)_{i \in I}, m\right)$ is the hypergraphical game $\tilde{G}=\left(\tilde{I}, \tilde{E},\left(\tilde{A}_{\left(i, \theta_{i}\right)}\right)_{\left(i, \theta_{i}\right) \in \tilde{I}},\left(\tilde{u}_{\left(i, \theta_{i}\right)}^{e}\right)_{e \in \tilde{E},\left(i, \theta_{i}\right) \in e}\right)$ :
- $\tilde{I}=\left\{\left(i, \theta_{i}\right) \mid i \in I, \theta_{i} \in \Theta_{i}\right\}, \tilde{E}=\left(e_{B}\right)_{B \in \mathcal{S}^{*}}, e_{B}=\left\{\left(i, \theta_{i}\right) \mid \theta \in B, i \in I\right\}, \tilde{A}_{\left(i, \theta_{i}\right)}=A_{i}$,
- for each $e_{B} \in \tilde{E},\left(i, \theta_{i}\right) \in e_{B}$ and $\tilde{\sigma} \in \tilde{A}$, let us pose $\tilde{v}_{i}^{\tilde{\sigma}}(\theta)=u_{i}\left(\tilde{\sigma}^{\theta}, \theta\right)$ in: $\tilde{u}_{\left(i, \theta_{i}\right)}^{e_{B}}\left(\tilde{\sigma}_{e_{B}}\right)=m\left(B \mid \theta_{i}\right) \times f_{\tilde{v}_{i}^{\tilde{\sigma}}}^{\mathrm{XEU}}(B)$.
Formally, let fXEU be any $\varphi^{\mathrm{XEU}}$, cond any conditioning, and $G$ be proper w.r.t. cond:

```
Variables (fXEU: (Tn > R) > {set Tn} >> R)
    (cond : conditioning R Tn) (proper_G : proper_belgame G cond).
```

After similar definitions for HRcond_localgame and HRcond_plays_in, we define:
Definition HRcond_u : $\forall \mathrm{lg}$, HRcond_localprof $\mathrm{lg} \rightarrow$ HRcond_localplayer $\mathrm{lg} \rightarrow \mathrm{R}$
$:=$ fun $\lg p \mathrm{x} \Rightarrow$ let (i_ti, Hi_ti) $:=\mathrm{x}$ in let ( $\mathrm{i}, \mathrm{ti}$ ) $:=$ i_ti in let $\mathrm{kn}:=$ cond $G .1$ (event_ti ti) (is_revisable proper_G ti) in $\mathrm{kn} \lg *$ fXEU [ffun $\mathrm{t} \Rightarrow \mathrm{G} .2$ (HRcond_mkprofile Hi_ti p t) t i] lg.
Definition HRcond : cgame R HR_action := hg_game HRcond_u.

- Theorem 13 (Correctness of the conditioned transform). For any proper Bel game G, Player $i$ with type $\theta_{i}$, conditioning $c$, XEU function $\varphi^{\mathrm{XEU}}$, profile $\sigma$ : $\mathrm{XEU}\left(m\left(\left.\cdot\right|_{c} \theta_{i}\right)\right)\left(\varphi^{\mathrm{XEU}}\left(u_{i, \sigma}\right)\right)=$ $\tilde{u}_{\left(i, \theta_{i}\right)}(\mathrm{flatten}(\sigma))$. Thence, Nash equilibria of $G$ and $\tilde{G}$ are in one-to-one correspondence:

```
Theorem HRcond_correct (i : I) (ti : T i) (p : iprofile A T):
    belgame_utility fXEU proper_G p ti = HRcond (flatten p) (existT _ i ti).
Theorem HRcond_eqNash (p : iprofile A T),
    BelG_Nash_equilibrium fXEU proper_G p < Nash_equilibrium (HRcond) (flatten p).
```


### 6.3 The TBM Transform

The TBM transform is designed for the Transferable Belief Model [33], in which knowledge is first revised using Dempster's conditioning, then decision is eventually made w.r.t. a probability distribution BetP which is deduced from the bpa $m$ (Definition 14). Here we benefit from 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 19 (TBM transform). Let $G=\left(I,\left(A_{i}, \Theta_{i}, u_{i}\right)_{i \in I}, m\right)$ be a Bel game; it is TBMtransformed into the hypergraphical game $\tilde{G}=\left(\tilde{I}, \tilde{E},\left(\tilde{A}_{\left(i, \theta_{i}\right)}\right)_{\left(i, \theta_{i}\right) \in \tilde{I}},\left(\tilde{u}_{\left(i, \theta_{i}\right)}^{e}\right)_{e \in \tilde{E},\left(i, \theta_{i}\right) \in e}\right)$ s.t.:
- $\tilde{I}=\left\{\left(i, \theta_{i}\right) \mid i \in I, \theta_{i} \in \Theta_{i}\right\}, \tilde{E}=(\operatorname{Players}(\theta))_{\theta \in \Theta}, \tilde{A}_{\left(i, \theta_{i}\right)}=A_{i}$,
- $\tilde{u}_{\left(i, \theta_{i}\right)}^{e_{\theta}}\left(\tilde{\sigma}_{e}\right)=\operatorname{BetP}_{\left(i, \theta_{i}\right)}(\theta) \times u_{i}\left(\sigma^{\theta}, \theta\right)$.

Formally, let cond be a conditioning and $G$ be a proper Bel game w.r.t. cond; $\tilde{G}$ 's local games are indexed by type configurations, and $\left(i, \theta_{i}\right)$ plays in $e_{\theta^{\prime}}$ iff $\theta_{i}=\theta_{i}^{\prime}$ :

```
Variables (cond : conditioning R Tn) (proper_G : proper_belgame cond).
Definition HRTBM_localgame : finType := Tn.
Definition HRTBM_plays_in : HRTBM_localgame > pred HR_player :=
    fun lg i_ti => lg (projT1 i_ti) == projT2 i_ti.
```

Local utilities are computed w.r.t. the "pignistic" distribution BetP:

```
Definition HRTBM_u : \foralllg, HRTBM_localprof lg > HRTBM_localplayer lg > R :=
    fun lg p x = let (i_ti, _) := x in let (i, ti) := i_ti in
    let betp := BetP (cond G.1 (event_ti ti) (is_revisable proper_G ti)) in
    dist betp lg * G.2 (HRTBM_mkprofile p) lg i.
Definition HRTBM : cgame R HR_action := hg_game HRTBM_u.
```

- Theorem 14 (Correctness of the TBM transform). For any proper Bel game G, Player $i$ with type $\theta_{i}$, conditioning $c$, and profile $\sigma, \operatorname{TBEU}\left(m\left(\left.\cdot\right|_{c} \theta_{i}\right)\right)\left(\varphi^{\operatorname{TBEU}}\left(u_{i, \sigma}\right)\right)=\tilde{u}_{\left(i, \theta_{i}\right)}($ flatten $(\sigma))$. Thence, Nash equilibria of $G$ and $\tilde{G}$ are in one-to-one correspondence:

```
Theorem HRTBM_correct (i : I) (ti : T i) (p : iprofile A T) :
    belgame_utility fTBEU proper_G p ti = HRTBM (flatten p) (existT _ i ti).
Theorem HRTBM_eqNash (p : iprofile A T),
    BelG_Nash_equilibrium fTBEU proper_G p << Nash_equilibrium HRTBM (flatten p).
```


## 7 Conclusion and Perspectives

In this paper, a Coq/SSReflect formalization of Bel games has been presented. It provides a generic class of games of incomplete information, built upon the Dempster-Shafer theory of belief functions. This framework makes it possible to capture (lack of) knowledge better than usual game models based on probability. Following Howson's and Rosenthal's approach, three different algorithms transforming such incomplete games into standard complete-information games have been formalized, and the fact that these transforms preserve equilibria (as long
as we comply with Table 4) has been formally verified. Thus, Bel games can ultimately be solved using state-of-the-art, effective algorithms for complete games.

This work provides strong guaranties on the correctness of the transforms, so that game theorists may rely on them without any concern about correctness. Furthermore, the formalization allowed us to identify subtleties that were left implicit in the definitions (e.g., the conditioning precondition), as well as to help improving the proofs, both in their flow and in their prose. Last, generic lemmas that proved useful during our formalization effort have been proposed for integration in the MathComp library.

This work opens several research directions, both on the theoretical side and on the formal verification side. On the one hand, we aim at extending this result with other decision-theoretic approaches, e.g., partially-ordered utility aggregations for belief function and other non-additive-measure approaches (Choquet capacities of order 2, RDU). On the other hand, we would like to focus on complexity proofs which, albeit not safety-critical, play a key role when choosing one transform over the other. Eventually, we would like to encompass this work into a larger library of decision under uncertainty, fostering further developments on related models and proofs.

## __ References

1 Reynald Affeldt and Cyril Cohen. Measure construction by extension in dependent type theory with application to integration, 2022. URL: https://arxiv.org/abs/2209.02345.
2 Reynald Affeldt, Jacques Garrigue, and Takafumi Saikawa. A library for formalization of linear error-correcting codes. Journal of Automated Reasoning, 64(6):1123-1164, 2020.
3 Nahla Ben Amor, Hélène Fargier, Régis Sabbadin, and Meriem Trabelsi. Possibilistic Games with Incomplete Information. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pages 1544-1550. ijcai.org, 2019.
4 Alexander Bagnall, Samuel Merten, and Gordon Stewart. A Library for Algorithmic Game Theory in SSReflect/Coq. Journal of Formalized Reasoning, 10(1):67-95, 2017.
5 Sylvie Boldo, François Clément, Florian Faissole, Vincent Martin, and Micaela Mayero. A Coq formalization of Lebesgue integration of nonnegative functions. Journal of Automated Reasoning, 66(2):175-213, 2022. URL: https://doi.org/10.1007/s10817-021-09612-0.
6 The Coq Development Team. The Coq Proof Assistant, 2022. URL: https://doi.org/10. 5281/zenodo. 1003420.
7 Arthur P. Dempster. Upper and Lower Probabilities Induced by a Multivalued Mapping. The Annals of Mathematical Statistics, 38:325-339, 1967.
8 Christoph Dittmann. Positional determinacy of parity games. Available at https://www. isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf, 2016.
9 Didier Dubois and Thierry Denoeux. Conditioning in Dempster-Shafer Theory: Prediction vs. Revision. In Belief Functions: Theory and Applications - Proceedings of the 2nd International Conference on Belief Functions, pages 385-392. Springer, 2012.
10 Didier Dubois and Henri Prade. Focusing vs. belief revision: A fundamental distinction when dealing with generic knowledge. In Qualitative and quantitative practical reasoning, pages 96-107. Springer, 1997.
11 Hélène Fargier, Érik Martin-Dorel, and Pierre Pomeret-Coquot. Games of incomplete information: A framework based on belief functions. In Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, volume 12897 of Lecture Notes in Computer Science, pages 328-341. Springer, 2021.
12 Ruobin Gong and Xiao-Li Meng. Judicious judgment meets unsettling updating: Dilation, sure loss and simpson's paradox. Statistical Science, 36(2):169-190, 2021.
13 John C Harsanyi. Games with Incomplete Information Played by "Bayesian" Players, I-III. Part I. The Basic Model. Management Science, 14(3):159-182, 1967.

14 Johannes Hölzl and Armin Heller. Three chapters of measure theory in Isabelle/HOL. In Interactive Theorem Proving, pages 135-151, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
15 Joseph T Howson Jr and Robert W Rosenthal. Bayesian Equilibria of Finite Two-Person Games with Incomplete Information. Management Science, 21(3):313-315, 1974.
16 Jean-Yves Jaffray. Linear Utility Theory for Belief Functions. Operations Research Letters, 8(2):107-112, 1989.
17 Jean-Yves Jaffray. Linear Utility Theory and Belief Functions: a Discussion. In Progress in decision, utility and risk theory, pages 221-229. Springer, 1991.
18 Rida Laraki, Jérôme Renault, and Sylvain Sorin. Mathematical foundations of game theory. Springer, 2019.
19 Stéphane Le Roux, Érik Martin-Dorel, and Jan-Georg Smaus. An Existence Theorem of Nash Equilibrium in Coq and Isabelle. In Proceedings Eighth International Symposium on Games, Automata, Logics and Formal Verification, volume 256 of Electronic Proceedings in Theoretical Computer Science, pages 46-60, 2017.
20 Stéphane Le Roux, Érik Martin-Dorel, and Jan-Georg Smaus. Existence of Nash equilibria in preference priority games proven in isabelle. In Kurt Gödel Day and Czech Gathering of Logicians, 2021.
21 Pierre Lescanne and Matthieu Perrinel. "backward" coinduction, nash equilibrium and the rationality of escalation. Acta Informatica, 49(3):117-137, 2012.
22 Assia Mahboubi and Enrico Tassi. Mathematical Components. Zenodo, January 2021.
23 Érik Martin-Dorel and Sergei Soloviev. A Formal Study of Boolean Games with Random Formulas as Payoff Functions. In 22nd International Conference on Types for Proofs and Programs, TYPES 2016, volume 97 of Leibniz International Proceedings in Informatics, pages 14:1-14:22, 2016.
24 Oskar Morgenstern and John Von Neumann. Theory of Games and Economic Behavior. Princeton University Press, 1953.
25 Roger B Myerson. Game Theory. Harvard university press, 2013.
26 John Nash. Non-Cooperative Games. Annals of Mathematics, pages 286-295, 1951.
27 Christos H. Papadimitriou and Tim Roughgarden. Computing Correlated Equilibria in Multi-Player Games. Journal of the Association for Computing Machinery, 55(3):1-29, 2008.
28 Bernard Planchet. Credibility and Conditioning. Journal of Theoretical Probability, 2(3):289299, 1989.
29 Pierre Pomeret-Coquot, Hélène Fargier, and Érik Martin-Dorel. Games of incomplete information: A framework based on belief functions. International Journal of Approximate Reasoning, 151:182-204, 2022.
30 Stéphane Le Roux. Acyclic preferences and existence of sequential nash equilibria: A formal and constructive equivalence. In Proc. Theorem Proving in Higher Order Logics, 22nd International Conference, volume 5674 of Lecture Notes in Computer Science, pages 293-309. Springer, 2009.
31 Glenn Shafer. A Mathematical Theory of Evidence. Princeton University Press, 1976.
32 Philippe Smets. Jeffrey's Rule of Conditioning Generalized to Belief Functions. In Uncertainty in artificial intelligence, pages 500-505. Elsevier, 1993.
33 Philippe Smets and Robert Kennes. The Transferable Belief Model. Artificial Intelligence, 66(2):191-234, 1994.
34 René Vestergaard. A constructive approach to sequential nash equilibria. Inormation. Processing Letters, 97(2):46-51, 2006. doi:10.1016/j.ipl.2005.09.010.
35 Peter Walley. Statistical Reasoning with Imprecise Probabilities. Chapman \& Hall, 1991.
36 Elena Yanovskaya. Equilibrium Points in Polymatrix Games. Lithuanian Mathematical Journal, 8:381-384, 1968.


[^0]:    * Copyright © Pierre Pomeret-Coquot and Hélène Fargier and Érik Martin-Dorel;

    Licensed under Creative Commons License CC-BY 4.0

[^1]:    ${ }^{1}$ MathComp notations: $\{$ set X$\}$ denotes finite sets over ( $\mathrm{X}: \mathrm{finType}$ ), $\mathrm{A}: \&: \mathrm{B}=A \cap B, \mathrm{~A}: \mid: \mathrm{B}=A \cup B$, $\sim: A=A^{c}$, set0 $=\emptyset ;\{f f u n X>Y\}$ denotes the type of finite support functions from (X : finType) to ( $\mathrm{Y}:$ Type); and for any ( $T: X \rightarrow$ Type), \{ffun forall $\mathrm{x}: \mathrm{X}, \mathrm{T} \mathrm{x}\}$ denotes the type of finite support functions with a dependently-typed codomain, mapping any ( $\mathrm{x}: \mathrm{X}$ ) to an element of ( T x ).

[^2]:    2 The names "pignistic" and BetP are references to classical Bayesian justification in decision theory, where both utilities and beliefs are elicited by considering limits of agent's agreement to a panel of bets.

[^3]:    3 Thus, type can refer to a type-theory concept or a game-theory one. Context will allow to disambiguate.

