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.

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 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 → Logic and verification; Theory of
computation → Type theory; Theory of computation → Higher order logic; 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, Coq Proof Assistant,
SSReflect Proof Language, MathComp Library

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°ANR-19-PI3A-0004.

*Copyright © Pierre Pomeret-Coquot and Hélène Fargier and Érik Martin-Dorel;
Licensed under Creative Commons License CC-BY 4.0
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 Dempster-Shafer theory of belief functions. This framework naturally encompass Bayesian games, as belief functions generalize probability measures. Also, we generalized the Houwen-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; Lescan 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, [33]). 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 $6k, to be shared between the people making the deal (that is, $2k or $3k depending on if the players choose the same partner or not); associating with the culprit produces no payoff ($0k). 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” \( \Omega = \{ \omega_1, \ldots, \omega_n \} \), one of which being the actual state of the world \( \omega^* \), and three functions \( m, \text{Bel} \) and \( \text{Pl} : 2^{\Omega} \to [0,1] \) 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 \( \text{Bel} \) and \( \text{Pl} \) 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 \( \Pr^* \) which is bounded by \( \text{Bel} \) and \( \text{Pl} : \forall A, \text{Bel}(A) \leq \Pr^*(A) \leq \text{Pl}(A) \); 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 \( \Pr(\{P, Q\}) = \Pr(\{R\}) = 1/2 \) (see Table 1).

<table>
<thead>
<tr>
<th>( A \subseteq \Omega )</th>
<th>( \emptyset )</th>
<th>{P}</th>
<th>{Q}</th>
<th>{P, Q}</th>
<th>{P, R}</th>
<th>{Q, R}</th>
<th>{P, Q, R}</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m(A) )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \text{Bel}(A) )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>( \text{Pl}(A) )</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( \Pr_5(A) )</td>
<td>0</td>
<td>( x )</td>
<td>0.5 - ( x )</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5 + ( x )</td>
<td>1 - ( x )</td>
</tr>
</tbody>
</table>

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 (\( \Pr_5 \)).

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.\(^1\)

\(^1\) MathComp notations: \( \{ \text{set } X \} \) denotes finite sets over \((X : \text{finType})\), \( A : \&: B = A \cap B, A : \|: B = A \cup B, \rightarrow A = A, \text{set0} = \emptyset \); \( \{ \text{fun } X \to Y \} \) denotes the type of finite support functions from \((X : \text{finType})\) to \((Y : \text{Type})\); and for any \((T : \text{Type})\), \( \{ \text{fun forall } x : X, T x \} \) denotes the type of finite support functions with a dependently-typed codomain, mapping any \((x : X)\) to an element of \((T x)\).
Definition bpa_axiom (m : {ffun {set W} → R}) :=
{∀ & m set0 == 0, ∑ m _ A m A == i & [∀ A, m A >= 0]}. Structure bpa := { bpa_val := {ffun {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 :&: A != set0) m B.

Lemma Bel_super m A B : {disjoint A & B} → Bel m (A :&: B) >= Bel m A + Bel m B.
Lemma Bel0 (m : bpa) : Bel m set0 = 0. (* and likewise for Pl *)

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 m over Ω, Definition 4 can straightforwardly be rephrased by rewriting the sums over the focal set:
Bel m (A) = \sum_ _ (B \subset A) m B and Pl m (A) = \sum_ _ (B \subset A \neq ∅) m B. Formally:
Lemma Bel_focalsetE m A := \sum_ _ (B in focalset m \subset A) m B.
Lemma Pl_focalsetE m A := \sum_ _ (B in focalset m \subset A :&: A != set0) m B.

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

Definition k_additivity m := \max_ _ (B in focalset m) #|B|.

Proposition 5 (Probability measure). Given a bpa m over Ω, if m is 1-additive, i.e. if all focal elements are singletons, then Bel = Pl is a discrete probability measure, associated with the distribution x ∈ Ω ↦ m(\{ω\}). Formally:
Structure proba := { proba_val := bpa ; proba_ax := k_additivity proba_val == 1 }. Lemma PrE (m : proba) (A : {set W}) : Bel m A = Pl m A.
Definition dist (m : proba) := fun w ⇒ 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 $(\cdot | C)$—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 $\text{revisable}$ predicate, which indicates whether an event can be assumed.

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

\[
\text{Definition } \text{conditioning}_\text{axiom} : (\text{revisable} : \text{bpa} \rightarrow \text{pred } \{\text{set } W\}) \rightarrow (\text{cond} : \forall m C, \text{revisable} m C \rightarrow \text{bpa}) := \\
\forall m C (Hrev : \text{revisable} m C), \text{Bel}(\text{cond} m C Hrev) (\cdot | C) = 0.
\]

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 $\text{conditioning}$ structure that encapsulates the $\text{revisable}$ predicate, the conditioning algorithm itself—which turns a $\text{revisable}$ prior in its posterior “given $C$”—and a proof of the $\text{conditioning}_\text{axiom}$:

Structure $\text{conditioning} := \{ \text{revisable} : \text{bpa} \rightarrow \text{pred } \{\text{set } W\} ; \\
\text{cond} _\text{val} : \forall m C, \text{revisable} m C \rightarrow \text{bpa} ; \\
\text{cond} _\text{ax} : \text{conditioning}_\text{axiom} \text{cond} _\text{val} \}.$

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 $\text{Bel}(\cdot | C)$ and $\text{Pl}(\cdot | C)$ 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 $\text{Pl}(C) \neq 0$, Dempster’s conditioning defines the bpa: $m(A | D C) = \sum_{B \cap C = A \neq \emptyset} m(B) / \text{Pl}(C)$.

Structure $\text{Dempster}_\text{revisable} m C := \text{Pl} m C \neq 0.$

Definition $\text{Dempster}_{\text{fun}} (m : \text{bpa}) (C : \{\text{set } W\}) := \{\text{ffun } A : \{\text{set } W\} \Rightarrow \\
\text{if } A == \text{set0 } \text{then } 0 \\
\text{else } \sum_{\{B : \{\text{set } W\} | \{B \notin \text{focalset } m\} \&\& \{B : \& C == \texttt{A} \}} m B / \text{Pl} m C\}.

Program Definition $\text{Dempster}_\text{cond} m C (Hrev : \text{Dempster}_\text{revisable} m C) : \text{bpa} := \\
\{\mid \text{bpa} _\text{val} := \text{Dempster}_\text{fun} m C ; \text{bpa} _\text{ax} := \_ \}.$

Program Definition $\text{Dempster}_\text{conditioning} : \text{conditioning} := \\
\{\mid \text{cond} _\text{val} := \text{Dempster}_\text{cond} ; \text{cond} _\text{ax} := \_ \}.$

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).

![Diagram of prior and posterior distributions](image)

**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, \( \Pi \)). For any bpa \( m \) and any event \( C \) such that \( \Pi(C) \neq 0 \), we have \( \Pi(A \mid_D C) = \Pi(A \cap C) / \Pi(C) \). Formally:

\[
\Pi(A \mid_D C) = \frac{\Pi(A \cap C)}{\Pi(C)}.
\]

**Lemma** Dempster_cond \( m \subseteq C \) (Hrev : revisable \( m \subseteq C \)):

\[ \forall A, \Pi(Dempster\_conditioning m \subseteq C \text{ Hrev}) A = \Pi(A \mid_D C) / \Pi(C). \]

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 \( Bel(C \mid C) < 1 \)). We also formalize these two rules below.

**Definition 9** (Strong conditioning). For any bpa \( m \) and any event \( C \) such that \( Bel(C) \neq 0 \), the weak conditioning is defined by the bpa \( m(A \mid_S C) = m(A) / Bel(C) \) if \( A \subseteq C \), 0 otherwise.

\[
Bel(A \mid_S C) = \frac{m(A)}{Bel(C)} 
\]

**Definition** Strong_revisable \( m \subseteq C \) := Bel \( m \subseteq C \neq 0 \).

**Definition** Strong_fun \( (m : \text{bpa}) (C : \text{set W}) := \{ \text{fun A : set W} \Rightarrow 
\text{if A != set0 then } m(A) / Bel(C) \text{ else } 0 \}. \)

**Program Definition** Strong_cond \( m \subseteq C \text{ Hrev}) : \text{bpa} := 
\{ \text{bpa_val := Strong_fun m \subseteq C ; bpa_ax := _1} \}.

**Program Definition** Strong_conditioning : conditioning := 
\{ \text{cond_val := Strong_cond ; cond_ax := _1} \}.

**Proposition 7** (Strong conditioning, \( Bel \)). For any bpa \( m \) and any event \( C \) such that \( Bel(C) \neq 0 \), we have \( Bel(A \mid_S C) = Bel(A \cap C) / Bel(C) \). Formally:

\[
Bel(A \mid_S C) = \frac{Bel(A \cap C)}{Bel(C)} 
\]

**Lemma** Strong_cond \( m \subseteq C \text{ Hrev}) A := Bel(A \mid_S C) / Bel(C) \).

**Definition** Weak_revisable \( m \subseteq C \) := P \( m \subseteq C \neq 0 \).

**Definition** Weak_fun \( (m : \text{bpa}) (C : \text{set W}) := \{ \text{fun A : set W} \Rightarrow 
\text{if A != set0 then } m(A) / (P m \subseteq C \neq 0) \}. \)

**Program Definition** Weak_cond \( m \subseteq C \text{ Hrev}) : \text{bpa} := 
\{ \text{bpa_val := Weak_fun m \subseteq C ; bpa_ax := _1} \}.

**Program Definition** Weak_conditioning : conditioning := 
\{ \text{cond_val := Weak_cond ; cond_ax := _1} \}.

**Proposition 8** (Weak conditioning, \( Bel \)). For any bpa \( m \) and any event \( C \) such that \( P(C) \neq 0 \), we have \( Bel(A \mid_W C) = (Bel(A) - Bel(A \setminus C)) / P(C) \). Formally:

\[
Bel(A \mid_W C) = \frac{Bel(A) - Bel(A \setminus C)}{P(C)} 
\]

**Lemma** Weak_cond \( m \subseteq C \text{ Hrev}) A := Bel(A \mid_W C) / P m \subseteq C. \)
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 outcomes 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 \( u : A \times \Omega \rightarrow \mathbb{R} \); \( u(a, \omega) > u(a', \omega') \) would mean the agent prefers the outcome of \( a \) when \( \omega^* = \omega \) to the outcome of \( a' \) when \( \omega^* = \omega' \). 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 \( u_a \succ u_{a'} \) would encode the fact the agent prefers \( a \) to \( a' \). 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 \( \varphi^{\text{CEU}} \) function (resp. \( \varphi^{\text{JEU}} \), \( \varphi^{\text{TBEU}} \)) over the powerset \( 2^\Omega \). Those \( \varphi^{\text{CEU}} \) functionals 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 \( \varphi : (\Omega \rightarrow \mathbb{R}) \rightarrow (2^\Omega \rightarrow \mathbb{R}) \) be a parameter function. We then consider the following generalized expected utility of \( a \):

\[
\text{XEU}(m)(\varphi(u_a)) = \sum_{B \in \text{focal}(m)} m(B) \times \varphi(u_a)(B).
\]

Let us review these \( \varphi^{\text{XEU}} \) 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:

\[
\text{CEU}(m)(u_a) = \sum_{B \in \text{focal}(m)} m(B) \times \min_{\omega \in B} u_a(\omega) = \text{XEU}(m)(\varphi^{\text{CEU}}(u_a)),
\]

with \( \varphi^{\text{CEU}}(u_a)(B) = \min_{\omega \in B} u_a(\omega) \).

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 \( \varphi^{\text{CEU}} \) and CEU are defined as follows:

**Definition** \( f\text{CEU} \) (\( u_a : \mathbb{W} \rightarrow \mathbb{R} \)):

\[
\text{fun} \ B \Rightarrow \text{match} \ \text{minS} \ u_a B \text{ with } \text{Some} \ r \Rightarrow r | \text{None} \Rightarrow 0 \ \text{end}.
\]

**Definition** CEU (\( m : \text{bpa} \)) (\( u_a : \mathbb{W} \rightarrow \mathbb{R} \)) := CEU \( m \) (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 $\alpha_{x, x^*} \in [0, 1]$ for each possible utility values $x, x^*$. 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:

\[
\text{JEU}^\alpha(m)(u_a) = \sum_{B \in S_m} m(B) \times (\alpha(B, B^*) \times B_* + (1 - \alpha(B, B^*)) \times B^*) = \text{XEU}(m)(\varphi_{\text{JEU}}^\alpha(u_a)),
\]

with $\varphi_{\text{JEU}}^\alpha(u_a)(B) = \alpha(B, B^*) \times B_* + (1 - \alpha(B, B^*)) \times B^*$. Formally:

| Definition fJEU $(\alpha : R \rightarrow R \rightarrow R) \rightarrow (\text{set } W \rightarrow R) : \{\text{set } W\} \rightarrow R :=$
| \begin{align*}
| & \text{fun } B \Rightarrow \text{match } \minS u_a B, \maxS u_a B \text{ with} \\
| & | \text{Some } \text{rmin}, \text{Some } \text{rmax} \Rightarrow \text{let } \alpha := \text{rmin} \times \text{rmax} \text{ in } \alpha \times \text{rmin} + (1 - \alpha) \times \text{rmax}
| \end{align*}
| Definition JEU $\alpha (m : \text{bpa}) (u_a : W \rightarrow R) := \text{XEU} m (fJEU \alpha \ u_a)$. |

Last, in the Transferable Belief Model [33], the decision rule is made by recovering a “pignistic” probability distribution\footnote{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.} 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 conditions, if any). The score of an action is then the expected utility w.r.t. BetP : $\omega \mapsto \sum_{B \in B_*} m(B) \times \text{len}(B)/|B|$, 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:

\[
\text{TBEU}(m)(u_a) = \sum_{B \in S_m} m(B) \times \sum_{\omega \in B} u_a(\omega) / |B| = \text{XEU}(m)(\varphi_{\text{TBEU}}(u_a))
\]

with $\varphi_{\text{TBEU}}(u_a)(B) = \sum_{\omega \in B} u_a(\omega)/|B|$. Formally:

| Definition fTBEU $(u_a : W \rightarrow R) \rightarrow (\text{set } W \rightarrow R) : \{\text{set } W\} \rightarrow R :=$
| \begin{align*}
| & \text{fun } B \Rightarrow \text{let } \text{alp} := \text{minS} u_a B, \text{maxS} u_a B \text{ with} \\
| & | \text{let } \omega := \text{alp} \text{ in } \omega
| \end{align*}
| Definition TBEU $m \rightarrow (\text{set } W \rightarrow R) : \{\text{set } W\} \rightarrow R := \text{XEU} m (fTBEU u_a)$. |

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 $(p : \text{proba}) u_a : \text{CEU} p u_a = \text{sum} \omega \text{ dist } p \omega \times u \omega$
| Lemma JEU_EU alpha $(p : \text{proba}) u_a : \text{JEU alpha} p u_a = \text{sum} \omega \text{ dist } p \omega \times u \omega$
| Lemma TBEU_EU $(p : \text{proba}) u_a : \text{TBEU} p u_a = \text{sum} \omega \text{ dist } p \omega \times u \omega$

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.
4.1 Games of Complete Information

Definition 15 (Game of complete information). A CGame is a tuple \( G = (I, (A_i, u_i)_{i \in I}) \) 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” \( a = (a_1, \ldots, a_n) \in A = A_1 \times \cdots \times A_n \). Player \( i \) prefers the outcome of profile \( a \) to that of \( a' \) 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, hence:

\[
\begin{aligned}
&\text{Definition cprofile (I : finType) (A : I \rightarrow eqType) := (ffun \forall i : I, A i).} \\
&\text{Definition cgame (I : finType) (A : I \rightarrow eqType) := cprofile A \rightarrow I \rightarrow \mathbb{R}.}
\end{aligned}
\]

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 \( j \neq i \), \( a'_i \) an action of Player \( i \), then \( a'_i a_{-i} \) denotes the profile \( a \) where the strategy of Player \( i \) has been switched to \( a'_i \) (called \text{change_strategy} \( a \ a'_i \) in Coq). A profile \( a \) is a pure Nash equilibrium iff \( \forall i, \forall a_i', a_i(a) \leq a_i(a_i, a_{-i}) \):

\[
\text{Definition change_strategy (p : cprofile A) (i : I) (a'_i : A i) : cprofile A}\\
\text{Definition Nash_equilibrium (G : cgame) (a : cprofile A) : bool :=}\\
\forall i : I, [\forall a'_i : A i, \neg (G a i < G (\text{change_strategy} a a'_i) i)].
\]

Example 3. Consider Example 1 anew; suppose one knows \( P \) is the murderer. The situation is captured by the CGame \( G = (I, (A_i, u_i)_{i \in I}) \) where \( I = \{1, 2\} \) is the set of players, \( A_i = \{P_i, Q_i, R_i\} \) the set of actions of Player \( i \) (choosing \( P, Q \) or \( R \)) and the \( u_i \)'s of Table 2.

\[
\begin{array}{ccc}
P_1 & Q_1 & R_1 \\
(0, 0) & (3, 0) & (3, 0) \\
(0, 3) & (2, 2) & (3, 3) \\
(3, 3) & (2, 2) & \\
\end{array}
\]

Table 2 Utility functions of Example 3 (it is known that \( P \) is the murderer). The pair \((u_1(a_1, a_2), u_2(a_1, a_2))\) is read at the intersection of line \( a_1 \) and column \( a_2 \).

Here, both \((Q_1, R_2)\) 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 \( \rho = (\rho_1, \ldots, \rho_n) \) is a vector of mixed strategies:

\[
\text{Definition mixed_cprofile := cprofile (fun i \Rightarrow [eqType of proba R (A i)]).}
\]

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

\[
\text{Definition mk_prod_proba (p : \forall i : X, proba R (A i)) : (ffun cprofile A \Rightarrow \prod prod_i dist (p i) (a i)).}\\
\text{Definition prod_proba (p : \forall i : I, proba R (A i)) (i0 : I) : proba R (cprofile A).}
\]

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 equilibrium extends straightforwardly:

\[
\text{Definition ms_util (G : cgame R A) (mp : mixed_cprofile) (i : I) : R :=}\\
\text{\sum_\langle p : cprofile A \rangle \text{dist (prod_proba mp witnessI mp) p} \ast (G \ p \ i).}\\
\text{Definition ms_Nash_equilibrium (G : cgame R A) (mp : mixed_cprofile) : Prop :=}\\
\forall i \ (\text{si : proba R (A i)}), \ 	ext{ms_util G mp i < ms_util G (\text{change_strategy mp si} i).}
\]
A standard reduction [18, Def. 4.6.1] amounts to viewing a mixed equilibrium of a game 
\((N, (A_i, u_i))_{i \in N}\) as a pure equilibrium in the mixed extension \((N, (A_i, u_i))_{i \in N}\), where \(A_i\) is the set of mixed strategies over \(A_i\). Formally:

```
Definition mixed_cgame (G : cgame R A) : cgame R (fun i => \[eqType of proba R (A_i)\]) :=
    fun mp i => ms_util G mp i.
Lemma mixed_cgameE G mp i : ms_utility G mp i = (mixed_cgame G) mp i.
```

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 \(G = (I, E, (A_e)_{i \in E, e \subseteq I}, (u_e)_e)\), 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), \(A\) is the set of actions of Player \(i\) and \(u_e : A_e \rightarrow R\) is the utility function of Player \(i\) in the local game \(e\) \((A_e = \prod_{i \in e} A_i\) 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 \rightarrow pred I).
```

For any local game \(lg\), local profiles are profiles that involve only players which \(plays_in lg\):

```
Definition localprof (lg : localgame) :=
    {ffun \forall 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: \(u_i(a) = \sum_{e \in E} u_e^i(a_e)\), 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 : \forall lg, localprof lg \rightarrow (i : I & plays_in lg i) \rightarrow R) : cgame :=
    fun a i \Rightarrow \sum_{s : \{lg : localgame | plays_in lg i\}} u (tag s) [ffun i \Rightarrow 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.

\(^3\) Thus, \textit{type} can refer to a type-theory concept or a game-theory one. Context will allow to disambiguate.
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, $\Omega = \prod_{i \in N} \Theta_i = \Theta$). 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].

**Definition 10 (Bel game).** A Bel game [29] is defined by a tuple $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$:
- $I$ 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 $i$;
- $u_i : A \to \Theta \to \mathbb{R}$ is the utility function of Player $i$; it depends on the joint action
$(a_1, \ldots, a_n) \in A := \prod_{i \in I} A_i$ and on the type configuration $(\theta_1, \ldots, \theta_n) \in \Theta := \prod_{i \in I} \Theta_i$;
- $m : 2^\Theta \to [0, 1]$ 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 $(A_i)_i$; and $T$, the family of types $(\Theta_i)_i$:

**Definition** belgame $(I : \text{finType}) (A : I \to \text{eqType}) (T : I \to \text{finType}) :=$

$\langle \text{bpa R (cprofile T) * (cprofile A \to cprofile T \to \text{player \to R})} \rangle$.

**Example 4 (Bel game).** We now are able to endow Example 1 with a Bel game $G =$
$(I, (A_i, \Theta_i, u_i)_{i \in I}, m)$. The set of players is $= \{1, 2\}$, their action sets are
$A_i = \{P_i, Q_i, R_i\}$.

Player 1 will learn either that $P$ is the murderer ($\omega^* \in \{P\}$) or that he is not ($\omega^* \in \{Q, R\}$);
Player 1’s type set is $\Theta_1 = \{\bar{P}, \bar{P}\}$. Similarly, Player 2 will learn either that $R$ is the murderer
($\omega^* \in \{R\}$) or that she is not ($\omega^* \in \{Q, P\}$), so $\Theta_2 = \{\bar{R}, \bar{R}\}$. The knowledge is expressed over
$\Theta = \Theta_1 \times \Theta_2$: since $(\bar{P}, \bar{R}) \equiv P$, $(\bar{P}, \bar{R}) \equiv Q$, $(\bar{P}, \bar{R}) \equiv R$ and $(\bar{P}, \bar{R})$ is impossible, the knowledge
is $m(\{(\bar{P}, \bar{R}), (\bar{P}, \bar{R})\}) = m(\{(P, R)\}) = 0.5$. Finally, utility functions are given in Table 3.

<table>
<thead>
<tr>
<th>$P_1$</th>
<th>$Q_1$</th>
<th>$R_1$</th>
<th>$P_2$</th>
<th>$Q_2$</th>
<th>$R_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0, 0)$</td>
<td>$(0, 3)$</td>
<td>$(0, 3)$</td>
<td>$(2, 2)$</td>
<td>$(3, 0)$</td>
<td>$(3, 3)$</td>
</tr>
<tr>
<td>$(3, 0)$</td>
<td>$(2, 2)$</td>
<td>$(3, 3)$</td>
<td>$(0, 3)$</td>
<td>$(0, 0)$</td>
<td>$(2, 2)$</td>
</tr>
<tr>
<td>$(3, 0)$</td>
<td>$(3, 3)$</td>
<td>$(2, 2)$</td>
<td>$(3, 3)$</td>
<td>$(0, 3)$</td>
<td>$(0, 3)$</td>
</tr>
</tbody>
</table>

**Table 3** Utility functions of Example 4 for $\theta = (P, R)$ (left, $P$ is the murderer), $\theta = (P, R)$ (center,
$Q$ is the murderer) and $\theta = (P, R)$ (right, $R$ is the murderer). Configuration $\theta = (P, R)$ can’t occur.

Since players know their own type before choosing their action, a pure strategy of Player $i$
becomes a function $\sigma_i : \Theta_i \to A_i$: having the type $\theta_i$, Player $i$ will play $\sigma_i(\theta_i) \in A_i$. Next, a
strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a vector of such functions:

**Definition** iprofile $I A T := \text{cprofile (fun i \Rightarrow [\text{eqType of \{\text{ffun T i \to A i\}}]})}$.

If the actual type configuration is $\theta = (\theta_1, \ldots, \theta_n)$, then for any strategy profile $\sigma$
by $\sigma^\theta = (\sigma_1(\theta_1), \ldots, \sigma_n(\theta_n)) \in A$ the action profile that will actually be played:

**Definition** proj_iprofile $I A T (p : \text{iprofile A T}) : \text{cprofile A} :=$

$\text{fun theta \Rightarrow [\text{ffun i \Rightarrow p i (theta i)]}}$.

In the following, we denote by $u_{i, \sigma} : \Theta \to \mathbb{R}$ the function mapping states of the world $\omega$
to the corresponding utility of $\sigma$ for Player $i$. It is defined by $u_{i, \sigma}(\theta) = u_i(\sigma^\theta, \theta)$.

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 $\sum_{B \in \Theta} m(B \mid \theta_i) \times \min_{\theta' \in \Theta} u_i(\sigma^\theta, \theta')$. 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: $\Theta$, representing the set $\Theta$ gathering all type configurations; and $\text{event}_{ti}$ := $\theta_i$ $\mapsto \{\theta' \in \Theta \mid \theta'_i = \theta_i\}$.

**Notation** $\Theta$ := $\{\text{intype of } \{\text{dfun } \forall \text{v i : I, T i}\}\}$.  

**Definition** $\text{event}_{ti}$ i := (ti : T i) := $\{\text{set } t : \Theta \mid t i := ti\}$.

A proper Bel game, in which conditioning is safe, shall satisfy the predicate:  

**Definition** proper_belgame A T (G : belgame A T) (cond : conditioning R Tn) := $\forall \theta_i \in \Theta_i$, $\forall i : I$, $T_i$.

\begin{itemize}
  \item **Definition 11** (Utility in a Bel game). For any Bel game $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$, any conditioning $\text{cond}$ for which $G$ is proper and any XEU parameter $\phi^{XEU} : (\Theta \rightarrow \mathbb{R}) \rightarrow 2^I \rightarrow \mathbb{R}$, the utility of the pure strategy profile $\sigma$ for Player $i$ having type $\theta_i \in \Theta_i$, is the integration of $u_i, \sigma$ := $\theta_i \mapsto u_i(\sigma^\theta, \theta)$, i.e., $XEU (m \cdot |_{\text{cond} \theta_i}) (\phi^{XEU}(u_i, \sigma))$.  

**Definition** belgame_utility A T (G : belgame A T) (cond : conditioning R Tn) := $\exists$ $\text{proper_belgame G}$ cond $\forall \theta_i \in \Theta_i$, $\forall i : I$, $T_i$.

\end{itemize}

Also, for Bel games, the definition of Nash equilibrium applies: an $\text{iprofile}$ is a Nash equilibrium iff no player, whatever is his type, has any incentive to deviate:

**Definition** BelG_Nash_equilibrium A T (G : belgame A T) (cond : conditioning R Tn) := $\forall \theta_i \in \Theta_i$, $\forall i : I$, $\forall \text{ai : A i}$,  
- $\text{belgame_utility u H p ti < belgame_utility u H (change_istrategy p ti ai) ti.}$

\begin{itemize}
  \item **Example 5** (Utility of a strategy). Let $\sigma = (\sigma_1, \sigma_2)$ be defined by $\sigma_1(P) = Q_1, \sigma_1(\bar{P}) = P_1$, $\sigma_2(R) = Q_2, \sigma_2(\bar{R}) = R_2$. $\sigma$ 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 $\bar{P}$ is the integration of $\phi^{CEU}(u_i, \sigma)$ w.r.t. the posterior bpa $m(\cdot \mid \bar{P})$. Recall Example 2, the posterior bpa “given $\bar{P}$” has two focal elements: $\{Q\}$ and $\{R\}$, both with mass 1/2. Considering type configurations, those focal elements are $\{(\bar{P}, \bar{R})\}$ and $\{(\bar{P}, R)\}$.  

$XEU (m \cdot |_{\bar{P}} \bar{P}) (\phi^{CEU}(u_i, \sigma)) = \sum_{B \in \mathcal{S}_{\text{node}(P)}} m(B \mid \bar{P}) \times \min_{\theta \in B} u_i(\sigma^\theta, \theta)$  

$= 0.5 \times u_1((P_1, Q_2), (\bar{P}, \bar{R})) + 0.5 \times u_1((P_1, R_2), (\bar{P}, R)) = 3$.  

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.

\section{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.

<table>
<thead>
<tr>
<th>Transform</th>
<th>Conditioning</th>
<th>XEU</th>
<th>Space</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct transform</td>
<td>Dempster’s c.</td>
<td>any</td>
<td>$O(k \times \text{Size}(G)^k)$</td>
<td>$O(k \times \text{Size}(G)^k)$</td>
</tr>
<tr>
<td>Conditioned transform</td>
<td>any</td>
<td>any</td>
<td>$O(k \times \text{Size}(G)^k)$</td>
<td>$O(k \times \text{Size}(G)^k)$</td>
</tr>
<tr>
<td>TBM transform</td>
<td>any</td>
<td>TBEU</td>
<td>$O(k \times \text{Size}(G))$</td>
<td>$O(\text{Size}(G))$</td>
</tr>
</tbody>
</table>

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 $\hat{G}$, with pairs $(i, \theta_i)$ as “abstract” players (i.e., $\hat{G}$’s vertices), denoting every type of every player of $G$. The local games correspond to focal elements, so Player $(i, \theta_i)$ plays in a local game $1g$ 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 $(G : \text{belgame} A T)$ be the input Bel game that has to be turned into a hypergraphical game named $\hat{G}$. $\hat{G}$’s players are pairs $(i, \theta_i)$, their action sets still are $A_i$:

**Definition** $HR_{\text{player}} : \text{finType} := \{\text{finType of i : I & T i}\}$.

**Definition** $HR_{\text{action}} (i_{\text{ti}} : HR_{\text{player}}) : \text{eqType} := A (\text{proj}T1 i_{\text{ti}})$.

Strategy profiles of $G$ and of $\hat{G}$ are in one-to-one correspondence. Every strategy profile $\sigma : \text{iprofile} A T$ of $G$, that is, $\sigma := \prod_{i \in I} (\Theta_i \rightarrow A_i)$, is flattened to $\hat{\sigma} : \text{iprofile} (\text{fun} i_{\text{ti}} : \{i : I & T i\} ⇒ A (\text{val} i))$ in $\hat{G}$, that is, $\hat{\sigma} := \prod_{(i, \theta_i) \in I \times \Theta_i} A_i$. E.g. in a 2-player game with 2 types per player, $\sigma = (\sigma_1, \sigma_2)$ is flattened to $(\sigma_1(\theta_1), \sigma_1(\theta_1'), \sigma_2(\theta_2), \sigma_2(\theta_2'))$.

This “dependent uncurrying” is performed by the following function:

**Definition** $\text{flatten} I (T : I ⇒ \text{finType}) A (\text{sigma} : \text{iprofile} A T) := [	ext{fun} i_{\text{ti}} ⇒ \text{sigma} (\text{proj}T1 i_{\text{ti}}) (\text{proj}T2 i_{\text{ti}})]$.

### 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 $(i, \theta_i)$ plays in $B$ iff $\theta_i$ is possible in $B$, that is, if $\exists \theta'$ in $B, \theta_i = \theta'_i$. Its local utility in $e_B$ is the “part of XEU” computed over $B'$, 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 $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$ is the hypergraphical game $\hat{G} = (\hat{I}, \hat{E}, (\hat{A}_i, \hat{\Theta}_i)_{i \in I}, (\hat{u}_i)_{(i, \theta_i) \in \hat{E}, (i, \theta_i) \in E})$:

- $\hat{I} = \{(i, \theta_i) | i \in I, \theta_i \in \Theta_i\}$,
- $\hat{E} = (e_B)_{B \subseteq S_m}$, $e_B = \{(i, \theta_i) | \theta \in B, i \in I\}$,
- $\hat{A}_i = A_i$,
- for each $e_B \in \hat{E}$, $(i, \theta_i) \in e_B$ and $\hat{\sigma} \in \hat{A}$ let us pose $\hat{u}_i(\hat{\sigma}) = u_i(\sigma^0, \theta)$ in:

\[
\hat{u}_i(\hat{\sigma})(\hat{\sigma}_{e_B}) = m(B) \times (\hat{\varphi}_{\text{XEU}} (\hat{u}_i)(B \cap \{(\theta' \mid \theta'_i = \theta_i)\}) / |B|)\{(\theta' \mid \theta_i = \theta_i)\}.
\]

Formally, let $G$ be a proper Bel game w.r.t. Dempster’s conditioning and $\varphi$ a function:

**Variable** (proper$_G : \text{proper} _{\text{belgame}} G \ (\text{Dempster} _{\text{conditioning} R Tn})$

(\text{fixEU} : (\text{fun} Tn \rightarrow R) \rightarrow (\text{fun} \{\text{set} Tn\} \rightarrow R))$

Then, let $\hat{G}$’s local games be indexed by focal elements, i.e., sets of type configurations:

**Definition** $HR_{\text{direct localgame}} := \{\text{finType of } \{\text{set} Tn\}\}$.

A vertex $(i, \theta_i)$ plays in the local game $e_B$ iff $\theta_i$ is possible in $B$:
The conditioned transform holds for any conditioning and element $B$. Note that Dempster’s conditioning transfers masses from $B$ to $B \cap \{\theta' \in \Theta | \theta'_i = \theta_i\} = B\cap(\theta(\text{event}_ti) \text{ if})$ so the local utility amounts to an on-the-fly Dempster’s conditioning. The resulting HG game is finally built from local utility functions:

$$\text{Definition HRdirect_u : } \forall lg \text{. HRdirect_localprof lg } \rightarrow \text{ HRdirect_localplayer lg } \rightarrow \text{ R}$$

$$:= \text{fun lg p x } \Rightarrow \text{ let (i,ti) := x in let (i,ti) := i,ti in G.1 lg \ast \text{ fXEU } [\text{f{fun t } \Rightarrow G.2 (HRdirect_mkprofile Hi_ti p t)} t i]}$$

$$\text{Definition HRdirect : cgame R HR_action := hg_game HRdirect_u.}$$

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 $S^*$ be the union of all posterior focal sets (i.e., the set of all possible focal elements given any $\theta_i$). Each $B \in S^*$ leads to a local game. As in the direct transform, a vertex $(i,\theta_i)$ plays in $e_B$ if $\theta_i$ is possible in $B$. Its utility in $e_B$ is part of XEU computed over the posterior focal element $B$. Note that $(i,\theta_i)$’s local utility in $B$ may be 0, if $B$ is not focal in the posterior “given $\theta_i$”. Formally speaking:

$$\text{Definition HRcond_localplayer : HRcond_localgame (i,ti) : HR_player}$$

$$:= \text{fun lg p x } \Rightarrow \text{ let (i,ti) := x in let (i,ti) := i,ti in G.1 lg \ast \text{ fXEU } [\text{f{fun t } \Rightarrow G.2 (HRcond_mkprofile Hi_ti p t)} t i]}$$

$$\text{Definition HRcond : cgame R HR_action := hg_game HRcond_u.}$$

After similar definitions for HRcond_localgame and HRcond_plays_in, we define:
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 $\text{BetP}$ which is deduced from the bpa $m$ (Definition 14). Here we benefit from $\text{BetP}$’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 $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$ be a Bel game; it is TBM-transformed into the hypergraphical game $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{i,\theta_i})_{(i,\theta_i) \in \tilde{I}}, (\tilde{u}_{(i,\theta_i)})_{(i,\theta_i) \in \tilde{I}}, (\tilde{u}_{(i,\theta_i)})_{(i,\theta_i) \in \tilde{I}})$ s.t.:  

$
\tilde{I} = \{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i, \tilde{E} = (\text{Players}(\theta))_{\theta \in \Theta}, \tilde{A}_{(i,\theta_i)} = A_i, 
\tilde{u}_{(i,\theta_i)}(\sigma_{(i,\theta_i)}) = \text{BetP}_{(i,\theta_i)}(\sigma) \times u_i(\sigma^i_{(i,\theta_i)} \theta_i) \}.
$

Formally, let $\text{cond}$ be a conditioning and $G$ be a proper Bel game w.r.t. $\text{cond}$; $G$’s local games are indexed by type configurations, and $(i, \theta_i)$ plays in $c_{\theta_i}$ iff $\theta_i = \theta_i'$.

Variables $(\text{cond} : \text{conditioning} \land \text{Tn}) \ (\text{proper}_G : \text{proper_belgame} \ \text{cond})$.

Definition HRTBM_localgame : finType $\Rightarrow$ Type := Type.

Definition HRTBM_plays_in : HRTBM_localgame $\Rightarrow$ pred HRTM_player :=

$\text{fun} \ lg \ i_{-}ti \Rightarrow \text{lg} (\text{projT1} i_{-}ti) \Rightarrow \text{projT2} i_{-}ti.$

Local utilities are computed w.r.t. the “pignistic” distribution $\text{BetP}$:

Definition HRTBM_u : $\forall$lg, HRTBM_localprof lg $\Rightarrow$ HRTBM_localplayer lg $\Rightarrow$ R :=

$\text{fun} \ lg \ p \ x \Rightarrow \text{let} \ (i_{-}ti, \_) := \text{lg in let} \ (1, \ ti) := i_{-}ti \ \text{in}$

$\text{let} \ \text{btep} := \text{BetP} \ \text{(cond} G.1 \ \text{(event} _{\text{ti}} ti) \ \text{(is_revisable} \ \text{proper}_G \ \text{ti)}) \ \text{in}$

$\text{dist} \ \text{btep} \ \text{lg} \star \ G.2 \ \text{(HRTBM_mkprofile} p) \ \text{lg} \ i$.

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 $\sigma$, $\text{TBEU} (m(\cdot | c_{\theta_i})) (\varphi^{\text{TBEU}}(u_i, \sigma)) = \tilde{u}_{i(\theta_i)}(\text{flatten} \sigma).$ 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) :

$\text{belgame.utility fTBEU proper}_G \ p \ ti = \text{HRTBM} \ \text{(flatten} p) \ \text{(existT} _{\text{ti}} ti)$.

Theorem HRTBM_eqNash (p : iprofile A T),

$\text{BelG_Nash.equilibrium fTBEU proper}_G \ p \ \Leftrightarrow \text{Nash.equilibrium} \ (\text{HRcond}) \ (\text{flatten} p).$
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 guarantees 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

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 Heidel-
berg.
15 Joseph T Howson Jr and Robert W Rosenthal. Bayesian Equilibria of Finite Two-Person
17 Jean-Yves Jaffray. Linear Utility Theory and Belief Functions: a Discussion. In Progress in
18 Rida Laraki, Jérôme Renault, and Sylvain Sorin. Mathematical foundations of game theory.
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
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
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
27 Christos H. Papadimitriou and Tim Roughgarden. Computing Correlated Equilibria in
29 Pierre Pomeret-Coquot, Hélène Fargier, and Érik Martin-Dorel. Games of incomplete informa-
tion: A framework based on belief functions. International Journal of Approximate Reasoning,
30 Stéphane Le Roux. Acyclic preferences and existence of sequential nash equilibria: A formal and
33 Philippe Smets and Robert Kennes. The Transferable Belief Model. Artificial Intelligence,
34 René Vestergaard. A constructive approach to sequential nash equilibria. Information Pro-
36 Elena Yanovskaya. Equilibrium Points in Polymatrix Games. Lithuanian Mathematical