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, CNRS, Toulouse INP, UT3, Toulouse, France

Hélène Fargier
IRIT, CNRS, Toulouse, France

Érik Martin-Dorel
IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, 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 [35]. 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 casted into games of complete information, i.e., without any uncertainty. We thus embed this class of games into classical game theory, enabling 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

Digital Object Identifier 10.4230/LIPIcs.ITP.2023.25

Related Version *Paper-and-pencil proof article*: `doi:10.1016/j.ijar.2022.09.010` [35]

Supplementary Material *Software (Formal proofs repo)*: `https://github.com/pPomCo/belgames`
archived at `swh:1:dir:5566e90ea5b3121a0b4f989a7584a251995c297a`

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

1 Introduction

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 [16]. 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 [15, 30]. 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 [13, 35], 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 *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 [35], 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: Kaliszyk et al. [33], formalizing foundations of decision making, using Isabelle/HOL; Vestergaard [40] then Le Roux [36], formalizing Kuhn's existence of a Nash equilibrium in finite games in extensive form, using Coq; Lescanne et al. [25], studying rationality of infinite games in extensive form, using Coq; Martin-Dorel et al. [27], 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. [23], 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 [24]. Furthermore, game theory is an important topic in economics: Lange et al. [21], proposing guidelines for formal reasoning; Kaliszyk et al. [20], formalizing microeconomic foundations, using Isabelle/HOL, and Echenim et al. [12], formalizing the Binomial Pricing Model, using Isabelle/HOL. 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. 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. Throughout the paper, we interleave mathematical and formal statements as needed to ensure our formally verified results are well-surveyable: the definitions and results are given first in mathematical syntax, then in Coq syntax when required (i.e., we include the Coq statements of the main theorems and the definitions they depend on, but not those of intermediate lemmas).

2 Motivating Example: the Murder of Mr. Jones

▶ **Example 1** (Inspired by *the Murder of Mr. Jones* [39]). Player 1 and Player 2 have to choose a partner which can be 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 their interest, choosing 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); choosing the culprit yields no payoff (\$0k). Moreover, Player 1 is investigating P and will know whether he is guilty before making the decision. Similarly, Player 2 will know whether R is guilty.

The Bayesian approach claims that any knowledge shall be described by a single subjective probability; it is not well-suited here. Indeed, assume Player 1 learns that Peter is innocent. It should not impact the evidence of 50% chance per sex, so the probability of guilt should become 1/2 for Quentin and 1/2 for Rose. However, in a purely Bayesian view, a prior probability must be made explicit, e.g., by equiprobability assumption: 1/4 for Peter, 1/4 for Quentin, and 1/2 for Rose. After conditioning, the posterior probability would not give 50% chance per sex anymore: equiprobability and conditioning "given Peter is innocent" yields 1/3 for Quentin and 2/3 for Rose. Learning that Peter is innocent would increase the odds against Rose! 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, 37] and the upper-lower probability theory (ULP) [7, 41]. 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 ω^*, and three functions $m : 2^\Omega \to [0,1]$, $\mathrm{Bel} : 2^\Omega \to [0,1]$, and $\mathrm{Pl} : 2^\Omega \to [0,1]$, which all map subsets of Ω 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 $\omega^* \in A$, but that does not support smaller claims such as $\omega^* \in B \subset A$. 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 Pr^* which is bounded by Bel and Pl: $\forall A, \mathrm{Bel}(A) \le \mathrm{Pr}^*(A) \le \mathrm{Pl}(A)$; 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 $(\mathrm{Pr}_x)_x$ which satisfy $\mathrm{Pr}_x(\{P,Q\}) = \mathrm{Pr}_x(\{R\}) = 1/2$ (see Table 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 $(\mathrm{Pr}_x)_{x\in[0,\,0.5]}$.

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

In this work, we follow the DS approach and formalize notions in terms of the function m. We chose to use the Coq proof assistant, along with the SSReflect tactic langage and the MathComp library [26]. This combination offers several features that contribute to facilitate the formalization: dependent types (making it possible to easily grasp usual definitions in game theory, and account for the variability of actions spaces w.r.t. individual agents), reflection prodicates (to easily go back and forth between decidable Boolean predicates and their propositional counterpart), packed classes and structure inference (making it possible to get formal statements as concise and legible as "LaTeX" ones) as well as the availability of comprehensive theories of finite sets or functions with finite support, endowed with decidable equality, and big operators such as $\sum$ or $\prod$.[1] Regarding automation (which is sometimes a criterium to choose a particular theorem prover over another): the formal verification of our results especially involves proofs by rewriting, very often under the binder of a big operator: the `under` tactic was instrumental to this aim [28]. Besides using this tactic, the formalization did not highlight a particular need to automate specific fragments or recurring proof goals.

▶ **Definition 2** (Frame of discernment)**.** *A* frame of discernment *is a finite set Ω, representing the possible states of the world. One of them is* the actual state of the world ω^*.

[1] MathComp notations: `{set X}` denotes finite sets over `(X : finType)`, `A:&:B` $= A \cap B$, `A:|:B` $= A \cup B$, `~:A` $= A^c$, `set0` $= \emptyset$; `{ffun X -> Y}` denotes the type of finite support functions from `(X : finType)` to `(Y : Type)`; and for any `(T : X -> Type)`, `{ffun 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 3** (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 Ω is endowed with MathComp's finite type structure, that is, `(W : finType)`. It makes it possible to use set operations and big operators. The carrier of the functions m, Bel, and Pl is `(R : realFieldType)`: only field operations are needed for this work.

▶ **Definition 4** (Basic probability assignment). *A basic probability assignment (bpa), a.k.a. mass function, is a set-function $m : 2^\Omega \to [0,1]$ such that:*

$$m(\emptyset) = 0 \quad \textit{and} \quad \sum\nolimits_{A\subseteq\Omega} m(A) = 1 \quad \textit{and} \quad \forall A \subseteq \Omega,\ m(A) \geq 0. \qquad \text{Formally:} \tag{1}$$

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

▶ **Definition 5** (Belief function, plausibility measure). *Given a bpa m over Ω, the associated belief function* $\mathrm{Bel} : 2^\Omega \to [0,1]$ *and plausibility measure* $\mathrm{Pl} : 2^\Omega \to [0,1]$ *are defined by:*

$$\mathrm{Bel}(A) = \sum\nolimits_{B\subseteq A} m(B) \quad \textit{and} \quad \mathrm{Pl}(A) = \sum\nolimits_{B\cap A\neq\emptyset} m(B).$$

▶ **Proposition 6** (Duality). *For any $A \subseteq \Omega$,* $\mathrm{Pl}(A) = 1 - \mathrm{Bel}(A^c)$ *and* $\mathrm{Bel}(A) = 1 - \mathrm{Pl}(A^C)$.

▶ **Proposition 7** (Super- and sub-additivity). Bel *is super-additive while* Pl *is sub-additive: for disjoint sets $A, B \subseteq \Omega$,* $\mathrm{Bel}(A \cup B) \geq \mathrm{Bel}(A) + \mathrm{Bel}(B)$ *and* $\mathrm{Pl}(A \cup B) \leq \mathrm{Pl}(A) + \mathrm{Pl}(B)$.

▶ **Proposition 8** (Bounds). *For any $A \subseteq \Omega$,* $0 = \mathrm{Bel}(\emptyset) = \mathrm{Pl}(\emptyset) \leq \mathrm{Bel}(A) \leq \mathrm{Pl}(A) \leq \mathrm{Bel}(\Omega) = \mathrm{Pl}(\Omega) = 1$.

▶ **Definition 9** (Focal elements, focal set). *Given a bpa m over Ω, 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 $\mathcal{S}_m$. In other words, $A \in \mathcal{S}_m$ iff $m(A) > 0$.*

▶ **Proposition 10** (Focal elements, focal set). *Given a bpa m over Ω, Definition 5 can straightforwardly be rephrased by rewriting the sums over the focal set:*

$$\mathrm{Bel}(A) = \sum\nolimits_{\substack{B\in\mathcal{S}_m\\ B\subseteq A}} m(B) \quad \textit{and} \quad \mathrm{Pl}(A) = \sum\nolimits_{\substack{B\in\mathcal{S}_m\\ B\cap A\neq\emptyset}} m(B).$$

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

▶ **Definition 11** (k-additivity). *For any bpa m, let $k = \max_{B\in\mathcal{S}_m} |B|$ be the maximal cardinality of its focal elements. Then, m is said to be k-additive.*

▶ **Definition 12** (Probability measure). *Given a bpa m over Ω, 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* `dist m` $: \Omega \to [0,1]$ defined by $x \mapsto m(\{\omega\})$.

```
Structure proba := {proba_val :> bpa; proba_ax : \max_(B in focalset proba_val) #|B| == 1}.
Definition dist (m : proba) := fun w => m [set w]. (*[set w] corresponds to {ω} *)
```

3.2 Conditioning in the Belief Function Theory

Conditioning is the operation that captures knowledge revision (fact learning) as well as focusing (hypothesis) [11, 9, 14]. 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 13** (Conditioning). *Given a bpa m, a predicate* $\texttt{revisable}_m : \Omega \to \{1, 0\}$*, and an event $C \subseteq \Omega$ such that* $\texttt{revisable}_m(C)$ *holds, a conditioning turns m into a bpa $m(\cdot \mid_{cond} C)$ such that the complement C^c of C is impossible, i.e.,* $\mathrm{Bel}(C^c \mid C) = 0$*. Formally:*

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

In other words, assuming m is `revisable` by the event C implies that if one learns that C holds, then one also learns that no evidence for the complement C^c 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 :> ∀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 $\mathrm{Bel}(\cdot \mid C)$ and $\mathrm{Pl}(\cdot \mid C)$ is the conditioning of those prior probabilities which assign the maximal probability to event C, that we now know for sure.

▶ **Definition 14** (Dempster's conditioning). *For any bpa m and any event C such that* $\mathrm{Pl}(C) \neq 0$*, Dempster's conditioning defines the bpa:* $m(A \mid_D C) = \sum_{B \cap C = A \neq \emptyset} m(B) / \mathrm{Pl}(C)$*. Formally:*

```
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 15** (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).

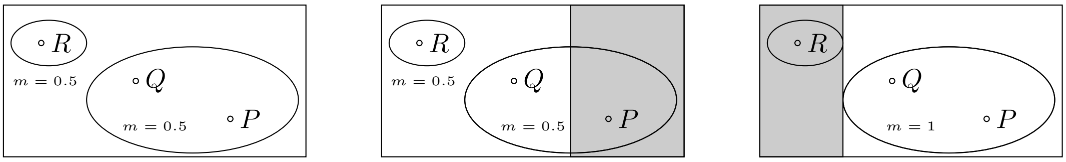

Figure 1 Prior (left) and posteriors given $\{Q, R\}$ (center) and given $\{P, Q\}$ (right). White and gray areas denote possible and impossible events – circles denote focal elements.

▶ **Proposition 16** (Dempster's conditioning, Pl). *For any bpa m and any event C such that $\mathrm{Pl}(C) \neq 0$, it holds that $\mathrm{Pl}(A \mid_D C) = \mathrm{Pl}(A \cap C)/\,\mathrm{Pl}(C)$.*

Two other rules have been proposed and called strong (resp. weak) conditioning [34]; the former, also known as geometrical conditioning [38], is another rule capturing knowledge revision; the latter is seldom used since it yields non-intuitive results (e.g., it may happen that $\mathrm{Bel}(C \mid C) < 1$). We also formalize these two rules below.

▶ **Definition 17** (Strong conditioning). *For any bpa m and any event C s.t. $\mathrm{Bel}(C) \neq 0$, the strong conditioning is defined by the bpa $m(A \mid_S C) = m(A)/\,\mathrm{Bel}(C)$ 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 18** (Strong conditioning, Bel). *For any bpa m and any event C such that $\mathrm{Bel}(C) \neq 0$, it holds that $\mathrm{Bel}(A \mid_S C) = \mathrm{Bel}(A \cap C)/\,\mathrm{Bel}(C)$.*

▶ **Definition 19** (Weak conditioning). *For any bpa m and any event C such that $Pl(C) \neq 0$, the weak conditioning is defined by the bpa: $m(A \mid_W C) = m(A)/Pl(B)$ 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} =>
  if A :&: C != set0 then m A / Pl m C else 0].
Program Definition Weak_cond m C (Hrev : Weak_revisable m C) : bpa :=
  {| bpa_val := Weak_fun m C ; bpa_ax := _ |}.
Program Definition Weak_conditioning : conditioning :=
  {| cond_val := Weak_cond ; cond_ax := _ |}.
```

▶ **Proposition 20** (Weak conditioning, Bel). *For any bpa m and any event C such that $\mathrm{Pl}(C) \neq 0$, it holds that* $\mathrm{Bel}(A \mid_W C) = (\mathrm{Bel}(A) - \mathrm{Bel}(A \setminus C))/\,\mathrm{Pl}(C)$.

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 ω^*. The agent's preferences on outcomes (which are left implicit here) are expressed by a real-valued utility function $u : A \times \Omega \to \mathbb{R}$: $u(a, \omega) > u(a', \omega')$ 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 $u_a : \Omega \to \mathbb{R}$ denote the partial application of u: u_a provides the utility of a depending on the state of the world ω.

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 [35], 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 $\varphi^{\text{XEU}}_{u_a}$ function (resp. $\varphi^{\text{CEU}}_{u_a}$, $\varphi^{\text{JEU}}_{u_a}$, and $\varphi^{\text{TBEU}}_{u_a}$) over the powerset 2^Ω. Those $\varphi^{\text{XEU}}_{u_a}$ functions are themselves parameterized 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 21** (Generalized expected utility)**.** *For any bpa m, any utility function $u : A \times \Omega \to \mathbb{R}$, and any $a \in A$, let us pose $u_a = \omega \mapsto u(a, \omega)$. Let $\varphi : (\Omega \to \mathbb{R}) \to \left(2^\Omega \to \mathbb{R}\right)$ be a parameter function. We then consider the following generalized expected utility of a:*

$$\text{XEU}(m)(\varphi(u_a)) = \sum\nolimits_{B \in \mathcal{S}_m} m(B) \times \varphi(u_a)(B). \quad \text{Formally:}$$

```
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 φ^{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 22** (Choquet expected utility)**.** *For any bpa m, any utility function $u : A \times \Omega \to \mathbb{R}$ and any action $a \in A$, the Choquet expected utility of $u_a : \Omega \to \mathbb{R}$ is:*

$$\text{CEU}(m)(u_a) = \sum\nolimits_{B \in \mathcal{S}_m} m(B) \times \min\nolimits_{\omega \in B} u_a(\omega) = \text{XEU}(m)(\varphi^{\text{CEU}}(u_a)),$$

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 φ^{CEU} and CEU are defined as follows:

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

Another rule, axiomatized by Jaffray [18, 19], 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 23** (Jaffray expected utility). *For any bpa m, any utility function $u : A \times \Omega \to \mathbb{R}$ and any action $a \in A$, the Jaffray expected utility of $u_a : \Omega \to \mathbb{R}$ is parameterized by a family of coefficients $\alpha_{(x_*,x^*)} \in [0,1]$ for each possible utility values $x_* \le x^*$. 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:*

$$\mathrm{JEU}^\alpha(m)(u_a) = \sum\nolimits_{B\in\mathcal{S}_m} m(B)\times\big(\alpha_{(B_*,B^*)} \times B_* + (1-\alpha_{(B_*,B^*)}) \times B^*\big) = \mathrm{XEU}(m)(\varphi^{\mathrm{JEU}^\alpha}(u_a)),$$

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

```
Definition fJEU (α : R -> R -> R) (u_a : W -> R) : {set W} -> R :=
  fun B => match minS u_a B, maxS u_a B with
  | Some rmin, Some rmax => let alp := α rmin rmax in alp * rmin + (1-alp) * rmax
  | _, _ => 0   end.
Definition JEU α (m : bpa) (u_a : W -> R) := XEU m (fJEU α u_a).
```

Finally, in the Transferable Belief Model [39], the decision rule is made by recovering a "pignistic" probability distribution[2] BetP 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. $\mathrm{BetP} : \omega \mapsto \sum_{\substack{B\in\mathcal{S}_m\\ \omega\in B}} m(B)/|B|$, that we show to be equivalent to the following definition.

▶ **Definition 24** (Transferable Belief Model expected utility). *For any bpa m, any utility function $u : A \times \Omega \to \mathbb{R}$ and any action $a \in A$, the TBEU of $u_a : \Omega \to \mathbb{R}$ is defined by:*

$$\mathrm{TBEU}(m)(u_a) = \sum\nolimits_{B\in\mathcal{S}_m} m(B) \times \sum\nolimits_{\omega\in B} u_a(\omega) \;/\; |B| = \mathrm{XEU}(m)(\varphi^{\mathrm{TBEU}}(u_a))$$

with $\varphi^{\mathrm{TBEU}}(u_a)(B) = \sum_{\omega\in B} u_a(\omega)/|B|$. *Formally:*

```
Definition fTBEU (u_a : W -> R) := fun B => \sum_(w in B) u_a w / #|B|%:R.
Definition TBEU (m : bpa) (u_a : W -> R) := XEU m (fTBEU u_a).
```

▶ **Proposition 25.** *CEU, JEU, and TBEU all generalize the expected utility criterion. For any probability distribution p, any utility function $u : A \times \Omega \to \mathbb{R}$ and any action $a \in A$,* $\mathrm{CEU}(p)(u_a) = \mathrm{JEU}^\alpha(p)(u_a) = \mathrm{TBEU}(p)(u_a) = \sum_{\omega\in\Omega} p(\omega) \times u_a(\omega)$.

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

4 Formalization of Several Classes of Games of Complete Information

Game theory is a subdomain of multi-agent decision making [29, 30]. 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] 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.

4.1 Games of Complete Information

▶ **Definition 26** (Game of complete information). *A CGame is a tuple $G = \big(I, (A_i, u_i)_{i \in I}\big)$ where I is a finite set of players; for each Player i, A_i is the set of their actions; $u_i : A \to \mathbb{R}$ is an utility function, assigning an utility value to each "action profile", i.e., a vector of actions, also called "pure strategy profile" $a = (a_1, \dots, a_n) \in A = A_1 \times \cdots \times A_n$. 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:

```
Definition cprofile (I : finType) (A : I -> eqType) := {ffun ∀i : I, A i}.
Definition cgame (I : finType) (A : I -> eqType) := cprofile A -> I -> R.
```

One of the most prominent solution concept in game theory is that of Nash equilibrium [31]:

▶ **Definition 27** (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* `change_strategy a a'_i` *in Coq). A profile a is a pure Nash equilibrium iff $\forall i, \forall a'_i,\ u_i(a) \not< u_i(a_i.a_{-i})$:*

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

▶ **Example 28.** 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. Here, both (Q_1, R_2) and (R_1, Q_2) are Nash equilibria.

Table 2 Utility functions of Example 28 (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.

	P_2	Q_2	R_2
P_1	$(0,0)$	$(0,3)$	$(0,3)$
Q_1	$(3,0)$	$(2,2)$	$\boxed{(3,3)}$
R_1	$(3,0)$	$\boxed{(3,3)}$	$(2,2)$

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

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

A mixed strategy profile ρ defines a probability over the set of pure strategy profiles, namely $p_\rho(a) = \prod_{i \in I} \rho_i(a_i)$. We package this data in a `proba` structure (Definition 12):

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

Last, the utility of a mixed strategy profile is the expected utility w.r.t. the probability over pure strategy profiles, and the notion of Nash 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 :=
  ∀i (si : proba R (A i)), ~ ms_util G mp i < ms_util G (change_strategy mp si) i.
```

A standard reduction [22, 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 $(N, (\mathcal{A}_i, u_i)_{i\in N})$, 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 ⇒ [eqType of proba R (A i)])
  := fun mp i ⇒ ms_util G mp i.
Theorem mixed_cgameE G mp i :  ms_utility G mp i = (mixed_cgame G) mp i.
Theorem 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 [32, 42], where the utility is not defined globally but locally (namely, split in several "local games"). This yields a hypergraph, where vertices denotes players and hyperedges denote local games. Formally, a hypergraphical game is a tuple $G = \big(I, E, (A_i)_{i\in I}, (u_i^e)_{e\in E, i\in e}\big)$, where I is the set of players, $E \subseteq 2^I$ is the set of local games (in any local game $e = \{a, b, c, \dots\}$, Players a, b, $c\dots$ are playing), A_i is the set of actions of Player i and $u_i^e : A_e \to \mathbb{R}$ 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 -> pred I).
```

For any local game `lg`, local profiles are profiles that involve only players which `plays\_in lg`:

```
Definition localprof (lg : localgame) :=
  {ffun ∀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: $u_i(a) = \sum_{\substack{e\in E\\ i\in e}} u_i^e(a_e)$, 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 : ∀lg, 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 ⇒ a (val i)] (exist _ i (tagged s)).
```

5 Bel Games

Harsanyi has proposed [15] 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 Θ_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.

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 were already defined in a possibilistic setting by Ben Amor et al. [3], which makes it possible to represent uncertainty using possibility theory [10]. We further extend this framework to Belief functions (encompassing both Bayesian games and possibilistic games) [13, 35].

▶ **Definition 29** (Bel game). *A Bel game* [35] *is defined by a tuple* $G = \big(I, (A_i, \Theta_i, u_i)_{i \in I}, m\big)$*:*
- *I is the finite set of players;*
- *A_i is the set of actions of Player i; Θ_i is the finite set of types of Player i;*
- *$u_i : A \times \Theta \to \mathbb{R}$ is the utility function of Player i; it depends on the joint action $(a_1, \dots, a_n) \in A := \prod_{i \in I} A_i$ and on the type configuration $(\theta_1, \dots, \theta_n) \in \Theta := \prod_{i \in I} \Theta_i$;*
- *$m : 2^\Theta \to [0, 1]$ 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 : finType) (A : I -> eqType) (T : I -> finType) :=
  (bpa R (cprofile T) * (cprofile A -> cprofile T -> player -> R)).
```

▶ **Example 30** (Bel game). We now are able to express Example 1 with a Bel game $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$. The set of players is $I = \{1, 2\}$, 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 $\Theta_1 = \{\mathtt{P}, \bar{\mathtt{P}}\}$. Similarly, Player 2 will learn either that R is the murderer ($\omega^* \in \{R\}$) or that she is not ($\omega^* \in \{P, Q\}$), so $\Theta_2 = \{\mathtt{R}, \bar{\mathtt{R}}\}$. The knowledge is expressed over $\Theta = \Theta_1 \times \Theta_2$: since $(\mathtt{P}, \bar{\mathtt{R}}) \equiv P$, $(\bar{\mathtt{P}}, \bar{\mathtt{R}}) \equiv Q$, $(\bar{\mathtt{P}}, \mathtt{R}) \equiv R$ and $(\mathtt{P}, \mathtt{R})$ is impossible, the knowledge is $m\big(\{(\mathtt{P}, \bar{\mathtt{R}}), (\bar{\mathtt{P}}, \bar{\mathtt{R}})\}\big) = m\big(\{(\bar{\mathtt{P}}, \mathtt{R})\}\big) = 0.5$. Finally, utility functions are given in Table 3.

Table 3 Utility functions of Example 30 for $\theta = (\mathtt{P}, \bar{\mathtt{R}})$ (left, P is the murderer), $\theta = (\bar{\mathtt{P}}, \bar{\mathtt{R}})$ (center, Q is the murderer) and $\theta = (\bar{\mathtt{P}}, \mathtt{R})$ (right, R is the murderer). Configuration $\theta = (\mathtt{P}, \mathtt{R})$ can't occur.

	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	$(0,3)$	$(0,3)$	$(0,0)$

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 θ_i, Player i will play $\sigma_i(\theta_i) \in A_i$. Next, a strategy profile $\sigma = (\sigma_1, \dots, \sigma_n)$ is a vector of such functions:

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

```
Definition iprofile I A T := cprofile (fun i ⇒ [eqType of {ffun T i -> A i}]).
```

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

```
Definition proj_iprofile I A T (p : iprofile A T) : cprofile A :=
  fun theta ⇒ [ffun i ⇒ p i (theta i)].
```

In the following, we denote by $u_{i,\sigma} : \Theta \to \mathbb{R}$ the function mapping states of the world ω to the corresponding utility of σ 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 σ for Player i with type θ_i is the expected utility w.r.t. the conditioned probability distribution "given θ_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 σ for Agent i with type θ_i is $\sum_{B\subseteq\Omega} m(B \mid_D \theta_i) \times \min_{\theta'\in B} u_i(\sigma^{\theta'}, \theta')$. Doing so, we need to ensure that conditioning is meaningful and technically possible, i.e., that the bpa is `revisable` given any type of any player. For the sake of readability, we now introduce two shorthands: `Tn`, representing the set Θ gathering all type configurations; and `event_ti` $:= \theta_i \mapsto \{\theta' \in \Theta \mid \theta'_i = \theta_i\}$:

```
Notation Tn := [finType of {dffun ∀i : 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
  := [∀i : player, [∀ti : T i, revisable cond G.1 (event_ti ti)]].
```

▶ **Definition 31** (Utility in a Bel game). *For any Bel game $G = \big(I, (A_i, \Theta_i, u_i)_{i\in I}, m\big)$, any conditioning* `cond` *for which G is proper and any* XEU *parameter $\varphi^{\mathrm{XEU}} : (\Theta \to \mathbb{R}) \to 2^\Theta \to \mathbb{R}$, the utility of the pure strategy profile σ for Player i having type $\theta_i \in \Theta_i$, is the integration of $u_{i,\sigma} = \theta \mapsto u_i(\sigma^\theta, \theta)$, i.e.,* $\mathrm{XEU}\,\big(m(\cdot \mid_{cond} \theta_i)\big)\big(\varphi^{\mathrm{XEU}}(u_{i,\sigma})\big)$.

```
Definition belgame_utility A T (G : belgame A T) (cond: conditioning R Tn)
  fXEU (HG : proper_belgame G cond) (p : iprofile A T) (i : player) (ti : T i) : R
  := let kn := cond G.1 (event_ti ti) (is_revisable HG ti) in
     XEU kn (fXEU (fun t ⇒ G.2 (proj_iprofile p t) t i)).
```

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) :=
  ∀i : I, ∀ti : T i, ∀ai : A i,
  ~ (belgame_utility u H p ti < belgame_utility u H (change_istrategy p ti ai) ti).
```

▶ **Example 32** (Utility of a strategy). Let $\sigma = (\sigma_1, \sigma_2)$ be defined by $\sigma_1(\mathtt{P}) = Q_1$, $\sigma_1(\bar{\mathtt{P}}) = P_1$, $\sigma_2(\mathtt{R}) = Q_2, \sigma_1(\bar{\mathtt{R}}) = R_2$. σ 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 σ for Player 1 with type $\bar{\mathtt{P}}$ is the integration of $\varphi^{\mathrm{CEU}}(u_{i,\sigma})$ w.r.t. the posterior bpa $m(\cdot \mid \bar{\mathtt{P}})$. Recall Example 15,

the posterior bpa "given $\bar{\mathtt{P}}$" has two focal elements: $\{Q\}$ and $\{R\}$, both with mass 1/2. Considering type configurations, those focal elements are $\{(\bar{\mathtt{P}}, \bar{\mathtt{R}})\}$ and $\{(\bar{\mathtt{P}}, \mathtt{R})\}$.

$$\begin{aligned}\mathrm{XEU}\left(m(\cdot \mid_D \bar{\mathtt{P}})\right)(\varphi^{\mathrm{CEU}}(u_{1,\sigma})) &= \sum_{B \in \mathcal{S}_{m(\cdot \mid_D \bar{\mathtt{P}})}} m(B \mid_D \bar{\mathtt{P}}) \times \min_{\theta \in B} u_1(\sigma^\theta, \theta) \\ &= 0.5 \times u_1((P_1, Q_2), (\bar{\mathtt{P}}, \mathtt{R})) + 0.5 \times u_1((P_1, R_2), (\bar{\mathtt{P}}, \bar{\mathtt{R}})) = 3.\end{aligned}$$

One may check that for every player and type, σ's CEU equals 3, the best possible score. Since no player, whatever is their type, has incentive to deviate, σ 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) [17]. 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 [13, 35]. We also extend the TBM transform to any conditioning (in the general case, a slight change is necessary, but for Dempster's and Strong conditioning the original statement holds, and so does the complexity of the transform). 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 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 [35]).

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

The three transforms all follow the same approach: starting from a Bel game G, they build the equivalent hypergraphical game $\tilde{G}$, with pairs (i, θ_i) 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 (i, θ_i) plays in a local game `lg` iff the type θ_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 (i, θ_i), 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 (σ `: iprofile A T)` in G, that is, $\sigma : \prod_{i \in I}(\Theta_i \to A_i)$, is flattened to $\tilde{\sigma}$ `: cprofile (fun i_ti : {i : I & T i} => A (val i))` in $\tilde{G}$, that is, $\tilde{\sigma} : \prod_{(i,\theta_i) \in I \times \Theta_i} A_i$. 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 flatten I (T : I -> finType) A (sigma : iprofile A T) :=
  [ffun i_ti => sigma (projT1 i_ti) (projT2 i_ti)].
```

6.1 The Direct Transform

The direct transform applies Dempster's conditioning on-the-fly (so this is the only possible conditioning). 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, θ_i) plays in B iff θ_i is possible in B, that is, if $\exists\theta' \in B,\ \theta_i = \theta'_i$. Its local utility in 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 33** (Direct transform of a Bel game). *The direct transform of a Bel game* $G = (I, (A_i, \Theta_i, u_i)_{i\in I}, m)$ *is the hypergraphical game* $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{(i,\theta_i)})_{(i,\theta_i)\in\tilde{I}}, (\tilde{u}^e_{(i,\theta_i)})_{e\in\tilde{E},(i,\theta_i)\in e})$*:*

- $\tilde{I} = \{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i\}$, $\tilde{E} = (e_B)_{B\subseteq\mathcal{S}_m}$, $e_B = \{(i, \theta_i) \mid \theta \in B, i \in I\}$, $\tilde{A}_{(i,\theta_i)} = A_i$,
- *for each* $e_B \in \tilde{E}$, $(i, \theta_i) \in e_B$ *and* $\tilde{\sigma} \in \tilde{A}$, *let us pose* $\tilde{v}^{\tilde{\sigma}}_i(\theta) = u_i(\tilde{\sigma}^\theta, \theta)$ *in:*
  $\tilde{u}^{e_B}_{(i,\theta_i)}(\tilde{\sigma}_{e_B}) = m(B) \times \left(\varphi^{\mathrm{XEU}}(\tilde{v}^{\tilde{\sigma}}_i)(B \cap \{\theta' \mid \theta'_i = \theta_i\})\right) / \mathrm{Pl}(\{\theta' \mid \theta'_i = \theta_i\})$.

Formally, let G be a proper Bel game w.r.t. Dempster's conditioning and `fXEU` a φ function:

```
Variable (proper_G : proper_belgame G (Dempster_conditioning R Tn))
         (fXEU : {ffun Tn -> R} -> {ffun {set Tn} -> 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 (i, θ_i) plays in the local game e_B iff θ_i is possible in B:

```
Definition HRdirect_plays_in (lg : HRdirect_localgame) (i_ti : HR_player) : bool
  := [∃t : 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 θ, the `cprofile` $(p_{(1,\theta_1)}, \ldots, p_{(n,\theta_n)})$. 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 $B \cap \{\theta' \in \Theta \mid \theta'_i = \theta_i\}$ $= B\cap$(`event_ti` θ_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 : ∀lg, 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 34** (Correctness of the direct transform). *For any proper Bel game G, Player i with type θ_i, XEU function φ^{XEU}, and profile σ, we have* $\mathrm{XEU}\left(m(\cdot \mid_D \theta_i)\right)\left(\varphi^{\mathrm{XEU}}(u_{i,\sigma})\right) = \tilde{u}_{(i,\theta_i)}(\texttt{flatten}(\sigma))$. *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 θ_i" are pre-computed, let $\mathcal{S}^*$ be the union of all posterior focal sets (i.e., the set of all possible focal elements given any θ_i). Each $B \in \mathcal{S}^*$ leads to a local game. As in the direct transform, a vertex (i, θ_i) plays in e_B if θ_i is possible in B. Its utility in e_B is the part of XEU computed over the posterior focal element B. Note that (i, θ_i)'s local utility in B may be 0, if B is not focal in the posterior "given θ_i". Formally speaking:

▶ **Definition 35** (Conditioned transform). *The conditioned transform of a Bel game $G = \big(I, (A_i, \Theta_i, u_i)_{i\in I}, m\big)$ is the hypergraphical game $\tilde{G} = \big(\tilde{I}, \tilde{E}, (\tilde{A}_{(i,\theta_i)})_{(i,\theta_i)\in\tilde{I}}, (\tilde{u}^e_{(i,\theta_i)})_{e\in\tilde{E},(i,\theta_i)\in e}\big)$:*

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

Formally, let `fXEU` be any φ^{XEU}, `cond` be 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 : ∀lg, HRcond_localprof lg -> HRcond_localplayer lg -> R
  := fun lg p x => let (i_ti, Hi_ti) := x in let (i, ti) := i_ti in
     let kn := cond G.1 (event_ti ti) (is_revisable proper_G ti) in
     kn lg * fXEU [ffun t => G.2 (HRcond_mkprofile Hi_ti p t) t i] lg.
Definition HRcond : cgame R HR_action := hg_game HRcond_u.
```

▶ **Theorem 36** (Correctness of the conditioned transform). *For any proper Bel game G, Player i with type θ_i, conditioning c, XEU function φ^{XEU}, profile σ:* $\mathrm{XEU}\big(m(\cdot \mid_c \theta_i)\big)\big(\varphi^{\mathrm{XEU}}(u_{i,\sigma})\big) = \tilde{u}_{(i,\theta_i)}(\texttt{flatten}(\sigma))$. *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 [39], 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 24). Here, we benefit from BetP's 1-additivity to produce a low-complexity hypergraph: local games correspond to single states of the world $\theta \in \Theta$. In this work, we generalize the TBM transform to any conditioning, refining the statement defining local games; we show that for Dempster's and the strong conditioning, the original statement suffices, unlike for the weak conditioning.

▶ **Definition 37** (TBM transform). *Let $G = \big(I, (A_i, \Theta_i, u_i)_{i\in I}, m\big)$ be a Bel game; it is TBM-transformed into the hypergraphical game $\tilde{G} = \big(\tilde{I}, \tilde{E}, (\tilde{A}_{(i,\theta_i)})_{(i,\theta_i)\in\tilde{I}}, (\tilde{u}^e_{(i,\theta_i)})_{e\in\tilde{E},(i,\theta_i)\in e}\big)$ s.t.:*

- $\tilde{I} = \big\{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i\big\}$, $\tilde{A}_{(i,\theta_i)} = A_i$, $\tilde{E} = (e_\theta)_{\theta\in\Theta}$,
- $e_\theta = \{(i, \theta'_i) \mid \theta'_i = \theta_i \lor (\exists B \in \mathcal{S}_{m(\cdot|\theta'_i)}, \theta \in B \land \exists \theta'' \in B, \theta'_i = \theta''_i)\}$,
- $\tilde{u}^{e_\theta}_{(i,\theta_i)}(\tilde{\sigma}_e) = \mathrm{BetP}_{(i,\theta_i)}(\theta) \times u_i(\sigma^\theta, \theta)$.

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 (i, θ_i) plays in $e_{\theta'}$ if $\theta_i = \theta'_i$ (the original statement, sufficient for Dempster's and the strong conditioning) or if there is a focal element B which contains both θ' and any θ'' such that $\theta_i = \theta''_i$ (necessary for the weak conditioning):

```
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 | [∃B, [&& B \in focalset (m_ti i_ti),
    lg \in B & [∃t, (t \in B) && (t (projT1 i_ti) == projT2 i_ti)]]]].
```

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

```
Definition HRTBM_u : ∀lg, 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 38** (Correctness of the TBM transform). *For any proper Bel game G, Player i with type θ_i, conditioning c, and profile σ,* $\mathrm{TBEU}\left(m(\cdot \mid_c \theta_i)\right)\left(\varphi^{\mathrm{TBEU}}(u_{i,\sigma})\right) = \tilde{u}_{(i,\theta_i)}(\texttt{flatten}(\sigma))$*. 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 Concluding remarks

In this paper, a 2.5 k LOC Coq/SSReflect formalization of Bel games has been presented. It gathers a theory for Dempster-Shafer belief functions (∼1 k LOC) as well as a generic class of games of incomplete information, built upon the former. 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 transforms casting such incomplete games into standard complete-information games [35] have been formalized, one of them being further generalized. We have formally verified that these transforms preserve equilibria. Thus, Bel games are solvable 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 pre- and post-conditions), 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. doi:10.1007/s10817-019-09538-8.

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. doi: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. *Possibility Theory: An Approach to Computerized Processing of Uncertainty.* Plenum Press, 1988.

11 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.

12 Mnacho Echenim and Nicolas Peltier. The binomial pricing model in finance: A formalization in Isabelle. In *CADE*, volume 10395 of *LNCS*, pages 546–562. Springer, 2017.

13 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 *LNCS*, pages 328–341. Springer, 2021. doi:10.1007/978-3-030-86772-0_24.

14 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.

15 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.

16 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.

17 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.

18 Jean-Yves Jaffray. Linear Utility Theory for Belief Functions. *Operations Research Letters*, 8(2):107–112, 1989.

19 Jean-Yves Jaffray. Linear Utility Theory and Belief Functions: a Discussion. In *Progress in decision, utility and risk theory*, pages 221–229. Springer, 1991.

20 Cezary Kaliszyk and Julian Parsert. Formal microeconomic foundations and the first welfare theorem. In *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2018*, pages 91–101. ACM, 2018.

21 Christoph Lange, Colin Rowat, and Manfred Kerber. The formare project - formal mathematical reasoning in economics. In *MKM/Calculemus/DML*, volume 7961 of *LNCS*, pages 330–334. Springer, 2013.
22 Rida Laraki, Jérôme Renault, and Sylvain Sorin. *Mathematical foundations of game theory*. Springer, 2019.
23 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. doi:10.4204/EPTCS.256.4.
24 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.
25 Pierre Lescanne and Matthieu Perrinel. "backward" coinduction, Nash equilibrium and the rationality of escalation. *Acta Informatica*, 49(3):117–137, 2012.
26 Assia Mahboubi and Enrico Tassi. Mathematical Components, 2022. URL: https://doi.org/10.5281/zenodo.7118596.
27 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.
28 Érik Martin-Dorel and Enrico Tassi. SSReflect in Coq 8.10. In *The Coq Workshop 2019*, Portland State University, OR, USA, 2019. URL: https://staff.aist.go.jp/reynald.affeldt/coq2019/coqws2019-martindorel-tassi.pdf.
29 Oskar Morgenstern and John Von Neumann. *Theory of Games and Economic Behavior*. Princeton University Press, 1953.
30 Roger B Myerson. *Game Theory*. Harvard university press, 2013.
31 John Nash. Non-Cooperative Games. *Annals of Mathematics*, pages 286–295, 1951.
32 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.
33 Julian Parsert and Cezary Kaliszyk. Towards formal foundations for game theory. In *Interactive Theorem Proving - 9th International Conference ITP 2018*, volume 10895 of *LNCS*, pages 495–503. Springer, 2018.
34 Bernard Planchet. Credibility and Conditioning. *Journal of Theoretical Probability*, 2(3):289–299, 1989.
35 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. doi:10.1016/j.ijar.2022.09.010.
36 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 *LNCS*, pages 293–309. Springer, 2009. doi:10.1007/978-3-642-03359-9_21.
37 Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
38 Philippe Smets. Jeffrey's Rule of Conditioning Generalized to Belief Functions. In *Uncertainty in artificial intelligence*, pages 500–505. Elsevier, 1993.
39 Philippe Smets and Robert Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66(2):191–234, 1994.
40 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.
41 Peter Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman & Hall, 1991.
42 Elena Yanovskaya. Equilibrium Points in Polymatrix Games. *Lithuanian Mathematical Journal*, 8:381–384, 1968.