



HAL
open science

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

Pierre Pomeret-Coquot, H el ene Fargier,  Erik Martin-Dorel

► **To cite this version:**

Pierre Pomeret-Coquot, H el ene Fargier,  Erik 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 destin ee au d ep ot et  a la diffusion de documents scientifiques de niveau recherche, publi es ou non,  emanant des  tablissements d'enseignement et de recherche franais ou  trangers, des laboratoires publics ou priv es.



Distributed under a Creative Commons Attribution 4.0 International License

1 **Bel-Games:**
2 **A Formal Theory of Games**
3 **of Incomplete Information**
4 **Based on Belief Functions**
5 **in the Coq Proof Assistant***

6 **Pierre Pomeret-Coquot** ✉ 
7 IRIT, Université de Toulouse III – Paul Sabatier, France

8 **Hélène Fargier** ✉ 
9 IRIT, CNRS, Toulouse, France

10 **Érik Martin-Dorel** ✉ 
11 IRIT, Université de Toulouse III – Paul Sabatier, France

12 — **Abstract** —

13 Decision theory and game theory are both interdisciplinary domains that focus on modelling
14 and analyzing decision-making processes. On the one hand, decision theory aims to account
15 for the possible behaviors of an agent with respect to an uncertain situation. It thus provides
16 several frameworks to describe the decision-making processes in this context, including that of
17 belief functions. On the other hand, game theory focuses on multi-agent decisions, typically with
18 probabilistic uncertainty (if any), hence the so-called class of Bayesian games. In this paper, we
19 use the Coq/SSReflect proof assistant to formally prove the results we obtained in [29]. First, we
20 formalize a general theory of belief functions with finite support, and structures and solutions
21 concepts from game theory. On top of that, we extend Bayesian games to the theory of belief
22 functions, so that we obtain a more expressive class of games we refer to as Bel games; it makes it
23 possible to better capture human behaviors with respect to lack of information. Next, we provide
24 three different proofs of an extended version of the so-called Howson–Rosenthal’s theorem, showing
25 that Bel games can be turned into games of complete information, i.e., without any uncertainty.
26 Doing so, we embed this class of games into classical game theory and thus enable the use of existing
27 algorithms.

28 **2012 ACM Subject Classification** Theory of computation → Logic and verification; Theory of
29 computation → Type theory; Theory of computation → Higher order logic; Theory of computation
30 → Algorithmic game theory; Theory of computation → Solution concepts in game theory; Theory
31 of computation → Representations of games and their complexity

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

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

36 **Funding** *Pierre Pomeret-Coquot:* ANTI, funded by the French “Investing for the Future – PIA3”
37 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

38 **1** Introduction

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

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

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

70 **Related works.** Several formalization efforts have been carried out in game theory since
 71 2006, each focusing on a somewhat different fragment: Vestergaard [34] then Le Roux [30],
 72 formalizing Kuhn’s existence of a Nash equilibrium in finite games in extensive form, using
 73 Coq; Lescanne et al. [21], studying rationality of infinite games in extensive form, using Coq;
 74 Martin-Dorel et al. [23], studying the probability of existence of winning strategies in Boolean
 75 finite games, using Coq; Bagnall et al. [4], formalizing well-known results of algorithmic
 76 game theory, using Coq; Dittmann [8], proving the positional determinacy of parity games,
 77 using Isabelle/HOL; Le Roux et al. [19], proving that a determinacy assumption implies
 78 the existence of Nash equilibrium in 2-player games, using Coq and Isabelle/HOL; this
 79 result being combined with that of Dittmann, using Isabelle/HOL [20]. Game theory aside,
 80 numerous works have been carried out in proof assistants to formalize probability and/or
 81 measure theory. Regarding the Coq proof assistant, we can mention the recent works by
 82 Affeldt et al. (on information theory based on discrete probability theory [2]; and measure
 83 theory based on **MathComp** [1]) and Boldo et al. [5], focusing on Lebesgue’s integral theory.

84 **Paper outline.** We start by introducing a motivating example, for which the Bayesian

85 approach fails but the Dempster-Shafer approach succeeds. Then, we incrementally present
 86 the Coq formalization of Bel games, giving successively the main mathematical definitions
 87 and lemmas and their formal Coq counterpart, throughout the paper: Section 3 presents the
 88 Dempster-Shafer theory of belief functions; then Section 4 focuses on complete-information
 89 games (including hypergraphical games) while Section 5 deals with Bel games; then Section 6
 90 is devoted to our formalization of Howson-Rosenthal’s generalized theorem in the scope of
 91 n -player Bel games. Finally, Section 7 gives concluding remarks and perspectives.

92 **2 Motivating Example: the Murder of Mr. Jones**

93 ► **Example 1** (Inspired by the *Murder of Mr. Jones*, [33]). Two players, named Player 1 and
 94 Player 2, are independently looking for an association, with either Peter (P), Quentin (Q),
 95 or Rose (R). The point is that a crime has been committed, for which these three people
 96 only are suspected. Furthermore, a poor-quality surveillance video allows to estimate that
 97 there is a 50% chance that the culprit is a man (P or Q), and a 50% chance that it is a
 98 woman (R). As to the interest of the associations, making the deal with an innocent people
 99 leads to a payoff of \$6k, to be shared between the people making the deal (that is, \$2k or
 100 \$3k depending on if the players choose the same partner or not); associating with the culprit
 101 produces no payoff (\$0k). Moreover, Player 1 is investigating about P and will know whether
 102 he is guilty before making the decision. Similarly, Player 2 will know whether R is guilty.

103 The Bayesian approach (claiming any knowledge shall be described by a single subjective
 104 probability) is not well-suited here. Assume Player 1 learns that P is innocent. It should
 105 not impact the evidence of 50% chance per sex, so the probability of guilt should become
 106 $1/2$ for Q and $1/2$ for R . However, in a purely Bayesian view, a prior probability must be
 107 made explicit (e.g. by equiprobability assumption: $1/4$ for P , $1/4$ for Q and $1/2$ for R), so
 108 after conditioning the posterior probability would not give 50% chance per sex anymore:
 109 equiprobability and conditioning “given $\neg P$ ” yields $1/3$ for Q and $2/3$ for R : learning that
 110 P is innocent would increase the odds against R ! In the sequel, we will reuse this example
 111 to highlight how the framework of belief functions better captures uncertain knowledge.

112 **3 Formalization of Belief Functions for Mono-Agent Decision Making**

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

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

127 3.1 Belief functions

128 The theory of belief functions is a powerful toolset from decision theory and statistics. It
 129 encompasses two distinct approaches for reasoning under uncertainty: the Dempster-Shafer
 130 theory of evidence (DS) [7, 31] and the upper-lower probability theory (ULP) [7, 35]. Both
 131 approaches consider a finite set of possible “states of the world” $\Omega = \{\omega_1, \dots, \omega_n\}$, one of
 132 which being the actual state of the world ω^* , and three functions m , Bel and Pl : $2^\Omega \rightarrow [0, 1]$
 133 which all map subsets of Ω to real numbers. Those functions are deducible one from another.
 134 In the DS theory, the mass function m is the basic knowledge about the world ($m(A)$ is
 135 the part of belief supporting the evidence $\omega^* \in A$: it asserts that A holds but cannot be
 136 refined into smaller claims). The non-additive continuous measures Bel and Pl indicate how
 137 much a proposition is implied by (resp. is compatible with) the knowledge. By contrast, the
 138 ULP theory suppose that there is an unknown probability \Pr^* which is bounded by Bel and
 139 Pl: $\forall A, \text{Bel}(A) \leq \Pr^*(A) \leq \text{Pl}(A)$; then m is just a concise representation of the family of
 140 compatible probabilities.

141 Example 1 can be understood in both theories. In the DS theory, $m(\{P, Q\}) = m(\{R\}) =$
 142 $1/2$ directly encodes the given evidences. In the ULP theory, one rather considers the family
 143 of probabilities which satisfy $\Pr(\{P, Q\}) = \Pr(\{R\}) = 1/2$ (see Table 1).

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

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

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

150 **► Definition 1** (Frame of discernment). *A frame of discernment is a finite set Ω , representing*
 151 *the possible states of the world. One of them is the actual state of the world ω^* .*

152 **► Definition 2** (Events). *A set $A \subseteq \Omega$ is an event which represents the proposition “ $\omega^* \in A$ ”.*
 153 *All set functions we consider (m , Bel, Pl) map events to real numbers within $[0, 1]$.*

154 The set Ω and the carrier of m , Bel and Pl are formalized in MathComp by asserting:

155 **Variable** ($\Omega : \text{finType}$) ($\mathbb{R} : \text{realFieldType}$).

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

$$158 \quad m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \subseteq \Omega} m(A) = 1 \quad \text{and} \quad \forall A \subseteq \Omega, m(A) \geq 0. \quad \text{Formally:} \quad (1)$$

159

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

```

160
161 Definition bpa_axiom (m : {ffun {set W} → R}) :=
162   [ && m set0 == 0, \sum_A m A == 1 & [∀A, m A >= 0] ].
163
164 Structure bpa := { bpa_val :> {ffun {set W} → R} ; bpa_ax : bpa_axiom bpa_val }.

```

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

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

```

168
169 Definition Bel (m : bpa) : {set W} → R := fun A => \sum_(B | B \subset A) m B.
170
171 Definition Pl (m : bpa) : {set W} → R := fun A => \sum_(B | B :&: A != set0) m B.

```

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

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

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

```

177
178 Lemma BelE (m : bpa) (A : {set W}) : Bel m A = 1 - Pl m (~:A).
179
180 Lemma Bel_super m A B : [disjoint A & B] → Bel m (A :|: B) >= Bel m A + Bel m B.
181
182 Lemma Bel0 (m : bpa) : Bel m set0 = 0. (* and likewise for Pl *)

```

► **Definition 5** (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 . Formally:*

```

185
186 Definition focal_element (m : bpa) : pred {set W} := fun A => m A > 0.
187
188 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:*

$$\text{Bel}(A) = \sum_{\substack{B \in \mathcal{S}_m \\ B \subseteq A}} m(B) \quad \text{and} \quad \text{Pl}(A) = \sum_{\substack{B \in \mathcal{S}_m \\ B \cap A \neq \emptyset}} m(B). \quad \text{Formally:}$$

```

194
195 Lemma Bel_focalsetE m A : Bel m A = \sum_(B in focalset m | B \subset A) m B.
196
197 Lemma Pl_focalsetE m A : Pl m A = \sum_(B in focalset m | B :&: A != set0) m B.

```

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

► **Definition 6** (k -additivity). *Given a bpa m over Ω and $k \in \mathbb{N}^*$, we say that the bpa m is k -additive iff the cardinality of its focal elements is at most k . Formally:*

```

202
203 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 $\text{Bel} = \text{Pl}$ is a discrete probability measure, associated with the distribution $x \in \Omega \mapsto m(\{\omega\})$. Formally:*

```

208
209 Structure proba := { proba_val :> bpa ; proba_ax : k_additivity proba_val == 1 }.
210
211 Lemma PrE (m : proba) (A : {set W}) : Bel m A = Pl m A.
212
213 Definition dist (m : proba) := fun w => m [set w]. (* [set w] corresponds to {w} *)
214
215 Lemma Pr_distE (m : proba) A : Bel m A = \sum_(w in A) dist m w.

```

214 3.2 Conditioning in the Belief Function Theory

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

221 Before dealing with conditional events $(\cdot | C)$ —read “given C ”—a precondition happens
 222 to be necessary: on the technical side, it avoids division-by-zero, and on the semantics
 223 side, it means one cannot learn that an impossible event holds. Since the definition of
 224 this precondition is specific to each conditioning rule, we abstract it away in the form of a
 225 **revisable** predicate, which indicates whether an event can be assumed.

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

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

234 In other words, assuming the event C is revisable implies that if one learns that C holds,
 235 then one also learns that no evidence for the complement can hold. Next, we formalize
 236 a **conditioning** structure that encapsulates the **revisable** predicate, the conditioning
 237 algorithm itself—which turns a **revisable** prior in its posterior “given C ”—and a proof of
 238 the **conditioning_axiom**:

```
239 Structure conditioning := { revisable : bpa → pred {set W} ;
240                             cond_val :> ∀ m C, revisable m C → bpa ;
241                             cond_ax : conditioning_axiom cond_val }.
```

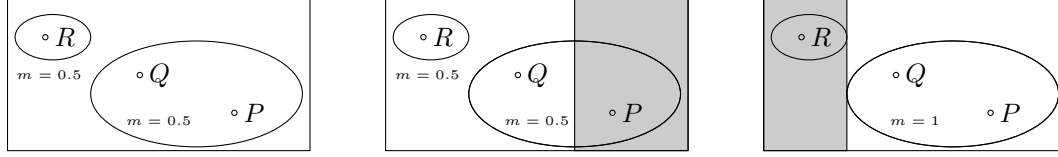
244 The most common conditioning is the so-called Dempster’s conditioning [7], which captures
 245 knowledge revision (i.e., fact learning). In the DS framework, it is understood as a transfer
 246 of parts of beliefs: learning that C holds, $m(B)$ is transferred to $B \cap C$ if it is not empty, or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

252 ► **Definition 8** (Dempster’s conditioning). *For any bpa m and any event C such that $\text{Pl}(C) \neq 0$,*
 253 *Dempster’s conditioning defines the bpa: $m(A |_D C) = \sum_{B \cap C = A \neq \emptyset} m(B) / \text{Pl}(C)$.*

```
254 Definition Dempster_revisable m C := Pl m C != 0.
255 Definition Dempster_fun (m : bpa) (C : {set W}) := [ffun A : {set W} =>
256   if A == set0 then 0
257   else \sum_(B : {set W} | (B \in focalset m) && (B :&: C == A)) m B / Pl m C].
258 Program Definition Dempster_cond m C (Hrev : Dempster_revisable m C) : bpa :=
259   { | bpa_val := Dempster_fun m C ; bpa_ax := _ | }.
260 Program Definition Dempster_conditioning : conditioning :=
261   { | cond_val := Dempster_cond ; cond_ax := _ | }.
```

264 ► **Example 2** (Knowledge revision, follow-up of Example 1/Table 1). Dempster’s conditioning is
 265 the conditioning approach fitting our example (see [9] for details). Suppose e.g. the murderer

266 is Q ; Player 1 learns $\omega^* \notin \{P\}$, i.e., $\omega^* \in \{Q, R\}$. From this viewpoint, the evidence
 267 concerning men now only concerns Q : the knowledge becomes $m(\{Q\}) = m(\{R\}) = 0.5$
 268 (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).



269 **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.

270 **Proposition 6** (Dempster's conditioning, Pl). *For any bpa m and any event C such that*
 271 $\text{Pl}(C) \neq 0$, *we have $\text{Pl}(A \mid_D C) = \text{Pl}(A \cap C) / \text{Pl}(C)$. Formally:*

272 **Lemma** Dempster_condE m C (Hrev : revisable m C) :
 273 $\forall A, \text{Pl} (\text{Dempster_conditioning } m \ C \ \text{Hrev}) \ A = \text{Pl } m \ (A \ \&: \ C) / \text{Pl } m \ C.$
 274
 275

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

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

282 **Definition** Strong_revisable m C := Bel m C != 0.
 283 **Definition** Strong_fun (m : bpa) (C : {set W}) := [ffun A : {set W} =>
 284 if (A != set0) && (A \subset C) then m A / Bel m C else 0].
 285 **Program Definition** Strong_cond m C (Hrev : Strong_revisable m C) : bpa :=
 286 { | bpa_val := Strong_fun m C ; bpa_ax := _ | }.
 287 **Program Definition** Strong_conditioning : conditioning :=
 288 { | cond_val := Strong_cond ; cond_ax := _ | }.
 289
 290

291 **Proposition 7** (Strong conditioning, Bel). *For any bpa m and any event C such that*
 292 $\text{Bel}(C) \neq 0$, *we have $\text{Bel}(A \mid_S C) = \text{Bel}(A \cap C) / \text{Bel}(C)$. Formally:*

293 **Lemma** Strong_condE m C (Hrev : revisable m C) :
 294 $\forall A, \text{Bel} (\text{Strong_conditioning } m \ C \ \text{Hrev}) \ A = \text{Bel } m \ (A \ \&: \ C) / \text{Bel } m \ C.$
 295
 296

297 **Definition 10** (Weak conditioning). *For any bpa m and any event C such that $\text{Pl}(C) \neq 0$,*
 298 *the weak conditioning is defined by the bpa: $m(A \mid_W C) = m(A) / \text{Pl}(C)$ if $A \cap C \neq \emptyset$,*
 299 *0 otherwise. Formally:*

300 **Definition** Weak_revisable m C := Pl m C != 0.
 301 **Definition** Weak_fun (m : bpa) (C : {set W}) := [ffun A : {set W} =>
 302 if A \&: C != set0 then m A / Pl m C else 0].
 303 **Program Definition** Weak_cond m C (Hrev : Weak_revisable m C) : bpa :=
 304 { | bpa_val := Weak_fun m C ; bpa_ax := _ | }.
 305 **Program Definition** Weak_conditioning : conditioning :=
 306 { | cond_val := Weak_cond ; cond_ax := _ | }.
 307
 308

309 **Proposition 8** (Weak conditioning, Bel). *For any bpa m and any event C such that*
 310 $\text{Pl}(C) \neq 0$, *we have $\text{Bel}(A \mid_W C) = (\text{Bel}(A) - \text{Bel}(A \setminus C)) / \text{Pl}(C)$. Formally:*

311 **Lemma** Weak_condE m C (Hrev : revisable m C) :
 312 $\forall A, \text{Bel} (\text{Weak_conditioning } m \ C \ \text{Hrev}) \ A = (\text{Bel } m \ A - \text{Bel } m \ (A \ \setminus: \ C)) / \text{Pl } m \ C.$
 313

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 ω^* . 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 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 [29], we analyzed three standard functions that generalize expected utility. They provide real values, and thus lead to completely ordered preferences over actions (since every two actions are directly comparable from their score). We denoted them CEU, JEU, and TBEU, respectively, for Choquet-, Jaffray-, and Transferable Belief-Expected Utility. We have shown they are all expressible as the integration of a particular $\varphi_{u_a}^{\text{XEU}}$ function (resp. $\varphi_{u_a}^{\text{CEU}}$, $\varphi_{u_a}^{\text{JEU}}$, and $\varphi_{u_a}^{\text{TBEU}}$) over the powerset 2^Ω . Those $\varphi_{u_a}^{\text{XEU}}$ functions 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 :*

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

```

341 Definition XEU (m : bpa) (phi_u_a : {ffun {set W} -> R}) : R :=
342   \sum (B in focalset m) m B * phi_u_a B.
343

```

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 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:*

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

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

```

357 Definition fCEU (u_a : W -> R) : {set W} -> R :=
358   fun B => match minS u_a B with Some r => r | None => 0 end.
359 Definition CEU (m : bpa) (u_a : W -> R) := XEU m (fCEU u_a).
360

```

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.

365 ▶ **Definition 13** (Jaffray expected utility). *For any bpa m , any utility function $u : A \times \Omega \rightarrow \mathbb{R}$*
 366 *and any action $a \in A$, the Jaffray expected utility of $u_a : \Omega \rightarrow \mathbb{R}$ is parameterized by a family*
 367 *of coefficients $\alpha_{(x_*, x^*)} \in [0, 1]$ for each possible utility values $x_* \leq x^*$. For any $B \neq \emptyset$, let us*
 368 *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:*

$$369 \quad \text{JEU}^\alpha(m)(u_a) = \sum_{B \in \mathcal{S}_m} m(B) \times (\alpha_{(B_*, B^*)} \times B_* + (1 - \alpha_{(B_*, B^*)}) \times B^*) = \text{XEU}(m)(\varphi^{\text{JEU}^\alpha}(u_a)),$$

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

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

378 Last, in the Transferable Belief Model [33], the decision rule is made by recovering a
 379 “pignistic” probability distribution² BetP (namely, a probability that serves only for the
 380 choice) at the very moment where the decision is made (so, the equiprobability assumption is
 381 made, but after conditionings, if any). The score of an action is then the expected utility w.r.t.
 382 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.

383 ▶ **Definition 14** (Transferable Belief Model expected utility). *For any bpa m , any utility*
 384 *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:*

$$385 \quad \text{TBEU}(m)(u_a) = \sum_{B \in \mathcal{S}_m} m(B) \times \sum_{\omega \in B} u_a(\omega) / |B| = \text{XEU}(m)(\varphi^{\text{TBEU}}(u_a))$$

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

```
387 Definition fTBEU (u_a : W → R) := fun B => \sum_w (w in B) u_a w / #|B|:R.
388 Definition TBEU (m : bpa) (u_a : W → R) := XEU m (fTBEU u_a).
```

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

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

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

401 4 Formalization of Several Classes of Games of Complete Information

402 Game theory is a subdomain of multi-agent decision making [24, 25]. In this paper, we focus
 403 on simultaneous games, in which *players* make their choice (called *action* or *pure strategy*)
 404 without knowing others’ choices in advance; the outcome of an action depends on the choices
 405 of other agents. A typical problem amounts to identifying which actions are relevant from
 406 the viewpoint of a player, assuming others don’t cooperate but strive to increase their own
 407 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 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” $a = (a_1, \dots, a_n) \in A = A_1 \times \dots \times A_n$. 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 [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 `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 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.

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

■ **Table 2** Utility functions of Example 3 (it is known that P is the murderer). The pair $(u_1(a_1.a_2), u_2(a_1.a_2))$ 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 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 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.
```

456 A standard reduction [18, Def. 4.6.1] amounts to viewing a mixed equilibrium of a game
 457 $(N, (A_i, u_i)_{i \in N})$ as a pure equilibrium in the mixed extension $(N, (\mathcal{A}_i, u_i)_{i \in N})$, where \mathcal{A}_i is
 458 the set of mixed strategies over A_i . Formally:

```
459 Definition mixed_cgame (G:cgame R A) : cgame R (fun i => [eqType of proba R (A i)])
460 := fun mp i => ms_util G mp i.
461
462 Lemma mixed_cgameE G mp i : ms_utility G mp i = (mixed_cgame G) mp i.
463 Lemma ms_NashE (G : cgame R A) (mp : mixed_cprofile) :
464 ms_Nash_equilibrium G mp <=> Nash_equilibrium (mixed_cgame G) mp.
```

466 4.2 Hypergraphical Games

467 Some games of complete information can be expressed succinctly as hypergraphical games
 468 [27, 36], where the utility is not defined globally but locally (namely, split in several “local
 469 games”). This yields a hypergraph, where vertices denotes players and hyperedges denote
 470 local games. Formally, a hypergraphical game is a tuple $G = (I, E, (A_i)_{i \in I}, (u_i^e)_{e \in E, i \in e})$,
 471 where I is the set of players, $E \subseteq 2^I$ is the set of local games (in any local game $e =$
 472 $\{a, b, c, \dots\}$, Players a, b, c, \dots are playing), A_i is the set of actions of Player i and $u_i^e : A_e \rightarrow \mathbb{R}$
 473 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
 474 related to e 's players). A hypergraphical game with 2-player local games is called a polymatrix.

475 In our formalization, local games are indexed by the finite type (`localgame : finType`);
 476 players playing a local game (`lg : localgame`) are those who verify the Boolean predicate
 477 (`plays_in lg`); `plays_in` thus formalizes E as a family of sets of players:

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

481 For any local game `lg`, local profiles are profiles that involve only players which `plays_in lg`:

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

486 In hypergraphical games, every player chooses one action, and plays it in every local game
 487 they are involved in. The global utility of a player is the sum of the locally obtained
 488 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
 489 hypergraphical game is a CGame that is specified by its local utility functions:

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

495 5 Bel Games

496 Harsanyi has proposed [13] a model for decision-making situations where players may have
 497 some uncertainty about other players, their actions, their utility functions, or more generally
 498 about any parameter of the game. To model such situations, the partially known parameters
 499 are expressed by so-called types:³ each Player i has a set of possible types Θ_i . Each type
 500 $\theta_i \in \Theta_i$ represents a possible parameter describing Player i 's characteristics and knowledge.
 501 Every Player i knows (or learns) their own type $\theta_i \in \Theta_i$ at the time of choosing an action. It
 502 may or may not be correlated with other players' types, so it is possible to model players
 503 that are not aware of other players' type as well as players with some knowledge about them.

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

504 Harsanyi defined the model of games of incomplete information where players' knowledge
 505 is given by a subjective probability and preferences agree with the expected utility: the
 506 so-called class of Bayesian games. In this setting, a probability measure expresses the
 507 knowledge on type configurations (the frame of discernment being the cartesian product of
 508 all players' types, that is, $\Omega = \prod_{i \in N} \Theta_i = \Theta$). Games of incomplete information already
 509 were extended to a possibilistic setting by Ben Amor et al. [3], and we extend it further to
 510 Belief functions (so generalizing both Bayesian games and possibilistic games) [11, 29].

511 **► Definition 10** (Bel game). *A Bel game [29] is defined by a tuple $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$:*
 512 **■** *I is the finite set of players;*
 513 **■** *A_i is the set of actions of Player i ; Θ_i is the finite set of types of Player i ;*
 514 **■** *$u_i : A \rightarrow \mathbb{R}$ is the utility function of Player i ; it depends on the joint action*
 515 *$(a_1, \dots, a_n) \in A := \prod_{i \in I} A_i$ and on the type configuration $(\theta_1, \dots, \theta_n) \in \Theta := \prod_{i \in I} \Theta_i$;*
 516 **■** *$m : 2^\Theta \rightarrow [0, 1]$ is a bpa which describes the prior knowledge.*

517 Formally speaking, a Bel game is fully defined by two elements: the bpa (prior knowledge)
 518 and the utility functions (the players' preferences). This pair is parameterized by three types
 519 I , the players; A , the family of actions $(A_i)_i$; and T , the family of types $(\Theta_i)_i$:

```
520 Definition belgame (I : finType) (A : I → eqType) (T : I → finType) :=
521 (bpa R (cprofile T) * (cprofile A → cprofile T → player → R)).
```

524 **► Example 4** (Bel game). We now are able to endow Example 1 with a Bel game $G =$
 525 $(I, (A_i, \Theta_i, u_i)_{i \in I}, m)$. The set of players is $\{1, 2\}$, their action sets are $A_i = \{P_i, Q_i, R_i\}$.
 526 Player 1 will learn either that P is the murderer ($\omega^* \in \{P\}$) or that he is not ($\omega^* \in \{Q, R\}$):
 527 Player 1's type set is $\Theta_1 = \{P, \bar{P}\}$. Similarly, Player 2 will learn either that R is the murderer
 528 ($\omega^* \in \{R\}$) or that she is not ($\omega^* \in \{P, Q\}$), so $\Theta_2 = \{R, \bar{R}\}$. The knowledge is expressed over
 529 $\Theta = \Theta_1 \times \Theta_2$: since $(P, \bar{R}) \equiv P$, $(\bar{P}, \bar{R}) \equiv Q$, $(\bar{P}, R) \equiv R$ and (P, R) is impossible, the knowledge
 is $m(\{(P, \bar{R}), (\bar{P}, \bar{R})\}) = m(\{(\bar{P}, R)\}) = 0.5$. Finally, utility functions are given in Table 3.

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

530 **■ Table 3** Utility functions of Example 4 for $\theta = (P, \bar{R})$ (left, P is the murderer), $\theta = (\bar{P}, \bar{R})$ (center,
 531 Q is the murderer) and $\theta = (\bar{P}, R)$ (right, R is the murderer). Configuration $\theta = (P, R)$ can't occur.

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

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

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

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

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

543 In *Bayesian games*, the global utility of a strategy profile σ for Player i with type θ_i is the
 544 expected utility w.r.t. the conditioned probability distribution “given θ_i ”. In *Bel games*, both
 545 expectation and conditioning have to be made explicit, to properly model agents' preferences
 546 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, that is, the bpa is **revisable** given any type of any player. For the sake of readability, we now introduce two shorthands: **Tn**, representing the set Θ gathering all type configurations; and **event_ti** := $\theta_i \mapsto \{\theta' \in \Theta \mid \theta'_i = \theta_i\}$:

```

549 Notation Tn := [finType of {dffun  $\forall i : I, T i$  }].
550 Definition event_ti i (ti : T i) := [set t : Tn | t i == ti].
551
552
553
554
555
556
557

```

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

```

558 Definition proper_belgame A T (G : belgame A T) (cond : conditioning R Tn) : bool
559 := [ $\forall i : \text{player}, [\forall ti : T i, \text{revisable cond G.1 (event\_ti ti)}]]$ .
560
561
562

```

► **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 **cond** for which G is proper and any XEU parameter $\varphi^{\text{XEU}} : (\Theta \rightarrow \mathbb{R}) \rightarrow 2^\Theta \rightarrow \mathbb{R}$, 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., $\text{XEU}(m(\cdot \mid_{\text{cond}} \theta_i))(\varphi^{\text{XEU}}(u_{i,\sigma}))$.*

```

563 Definition belgame_utility A T (G : belgame A T) (cond : conditioning R Tn)
564 fXEU (HG : proper_belgame G cond) (p : iprofile A T) (i : player) (ti : T i) : R
565 := let kn := cond G.1 (event_ti ti) (is_revisable HG ti) in
566 XEU kn (fXEU (fun t => G.2 (proj_iprofile p t) t i)).
567
568
569
570
571
572

```

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:

```

573 Definition BelG_Nash_equilibrium A T (G : belgame A T) (cond : conditioning R Tn)
574 fXEU (H : proper_belgame G cond) (p : iprofile A T) :=
575  $\forall i : I, \forall ti : T i, \forall ai : A i,$ 
576  $\sim (\text{belgame\_utility } u \text{ H } p \text{ ti} < \text{belgame\_utility } u \text{ H } (\text{change\_istrategy } p \text{ ti } ai) \text{ ti})$ .
577
578
579
580

```

► **Example 5** (Utility of a strategy). Let $\sigma = (\sigma_1, \sigma_2)$ be defined by $\sigma_1(P) = Q_1, \sigma_1(\bar{P}) = P_1, \sigma_2(R) = Q_2, \sigma_2(\bar{R}) = R_2$. 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{P} is the integration of $\varphi^{\text{CEU}}(u_{i,\sigma})$ 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 $\{(\bar{P}, R)\}$.

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

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

In the following, we formally define and prove correct three Howson–Rosenthal-like transforms that we have devised in previous work [11, 29]. All these transforms cast n -player Bel games into hypergraphical games; the games so obtained all have the same utility values,

though different hypergraphs. These transforms can be applied safely, depending on the conditioning and on the decision rule (cf. Table 4): Dempster’s conditioning is hard-coded into the Direct transform while the TBM transform’s low complexity comes from properties of the distribution BetP considered by the TBEU.

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

Table 4 Transforms, conditioning and XEU they are suited for, and their worst-case complexity w.r.t. the k -additivity of the bpa and the size of the input Bel game (taken from [29]).

The three transforms all follow the same approach: starting from a Bel game G , they build the equivalent hypergraphical game \tilde{G} , with pairs (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 $(G : \text{belgame } A \ T)$ 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 :

```

615 Definition HR_player : finType := [finType of {i : I & T i}].
616 Definition HR_action (i_ti : HR_player) : eqType := A (projT1 i_ti).
617

```

Strategy profiles of G and of \tilde{G} are in one-to-one correspondance. 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 $\tilde{\sigma} : \text{cprofile } (\text{fun } i_ti : \{i : I \ \& \ T \ i\} \Rightarrow A \ (\text{val } i))$ 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:

```

624 Definition flatten I (T : I -> finType) A (sigma : iprofile A T) :=
625   [ffun i_ti => sigma (projT1 i_ti) (projT2 i_ti)].
626

```

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, 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 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 $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{(i, \theta_i)})_{(i, \theta_i) \in \tilde{I}}, (\tilde{u}_{(i, \theta_i)}^e)_{e \in \tilde{E}, (i, \theta_i) \in e})$:*

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

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

```

640 Variable (proper_G : proper_belgame G (Dempster_conditioning R Tn))
641   (fXEU : {ffun Tn -> R} -> {ffun {set Tn} -> R}).
642

```

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

```

645 Definition HRdirect_localgame := [finType of {set Tn}].
646

```

A vertex (i, θ_i) plays in the local game e_B iff θ_i is possible in B :

```

649
650 Definition HRdirect_plays_in (lg : HRdirect_localgame) (i_ti : HR_player) : bool
651 := [∃ t : Tn, [∧ t \in lg & t (projT1 i_ti) == projT2 i_ti]].
652

```

Then, local utility functions are given by a function which constructs from a local profile p and a type configuration θ , the cprofile $(p_{(1,\theta_1)}, \dots, p_{(n,\theta_n)})$. This function has type:

```

653
654 Definition HRdirect_mkprofile lg i_ti (Hi_ti : HRdirect_plays_in lg i_ti)
655 (p : HRdirect_localprof lg) (t : Tn) : profile.
656

```

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 $B \cap \{\theta' \in \Theta \mid \theta'_i = \theta_i\} = B \cap (\text{event_ti } \theta_i)$ so the local utility amounts to an on-the-fly Dempster's conditioning. The resulting HG game is finally built from local utility functions:

```

657
658 Definition HRdirect_u : ∀ lg, HRdirect_localprof lg → HRdirect_localplayer lg → R
659 := fun lg p x ⇒ let (i_ti, Hi_ti) := x in let (i, ti) := i_ti in
660   G.1 lg * fXEU [ffun t ⇒ G.2 (HRdirect_mkprofile Hi_ti p t) t i]
661   (lg &: (event_ti ti)) / Pl G.1 (event_ti ti).
662
663 Definition HRdirect : cgame R HR_action := hg_game HRdirect_u.
664

```

► **Theorem 12** (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 $\text{XEU}(m(\cdot \mid_D \theta_i))(\varphi^{\text{XEU}}(u_{i,\sigma})) = \tilde{u}_{(i,\theta_i)}(\text{flatten}(\sigma))$. Thence, Nash equilibria of G and \tilde{G} are in one-to-one correspondence:*

```

673
674 Theorem HRdirect_correct (i : I) (ti : T i) (p : iprofile A T) :
675   belgame_utility fXEU properG p ti = HRdirect (flatten p) (existT _ i ti).
676
677 Theorem HRdirect_eqNash (p : iprofile A T) :
678   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 18** (Conditioned transform). *The conditioned transform of a Bel game $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$ is the hypergraphical game $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{(i,\theta_i)})_{(i,\theta_i) \in \tilde{I}}, (\tilde{u}_{(i,\theta_i)}^e)_{e \in \tilde{E}, (i,\theta_i) \in e})$:*

- $\tilde{I} = \{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i\}$, $\tilde{E} = (e_B)_{B \in \mathcal{S}^*}$, $e_B = \{(i, \theta_i) \mid \theta \in B, i \in I\}$, $\tilde{A}_{(i,\theta_i)} = A_i$,
- for each $e_B \in \tilde{E}$, $(i, \theta_i) \in e_B$ and $\tilde{\sigma} \in \tilde{A}$, let us pose $\tilde{v}_i^{\tilde{\sigma}}(\theta) = u_i(\tilde{\sigma}^\theta, \theta)$ in:

$$\tilde{u}_{(i,\theta_i)}^{e_B}(\tilde{\sigma}_{e_B}) = m(B \mid \theta_i) \times f_{\tilde{v}_i^{\tilde{\sigma}}}^{\text{XEU}}(B).$$

Formally, let fXEU be any φ^{XEU} , cond any conditioning, and G be proper w.r.t. cond:

```

692
693 Variables (fXEU: (Tn → R) → {set Tn} → R)
694 (cond : conditioning R Tn) (proper_G : proper_belgame G cond).
695

```

After similar definitions for HRcond_localgame and HRcond_plays_in, we define:

```

696
697 Definition HRcond_u : ∀ lg, HRcond_localprof lg → HRcond_localplayer lg → R
698 := fun lg p x ⇒ let (i_ti, Hi_ti) := x in let (i, ti) := i_ti in
699   let kn := cond G.1 (event_ti ti) (is_revisable proper_G ti) in
700   kn lg * fXEU [ffun t ⇒ G.2 (HRcond_mkprofile Hi_ti p t) t i] lg.
701
702 Definition HRcond : cgame R HR_action := hg_game HRcond_u.
703

```


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

```
707
708 Theorem HRcond_correct (i : I) (ti : T i) (p : iprofile A T) :
709   belgame_utility fXEU proper_G p ti = HRcond (flatten p) (existT _ i ti).
710 Theorem HRcond_eqNash (p : iprofile A T),
711   BelG_Nash_equilibrium fXEU proper_G p <=> Nash_equilibrium (HRcond) (flatten p).
712
```

713 6.3 The TBM Transform

714 The TBM transform is designed for the Transferable Belief Model [33], in which knowledge
 715 is first revised using Dempster’s conditioning, then decision is eventually made w.r.t. a
 716 probability distribution BetP which is deduced from the bpa m (Definition 14). Here we
 717 benefit from BetP’s 1-additivity to produce a low-complexity hypergraph: local games
 718 correspond to single states of the world, that is, they all involve only n players.

719 ► **Definition 19** (TBM transform). *Let $G = (I, (A_i, \Theta_i, u_i)_{i \in I}, m)$ be a Bel game; it is TBM-*
 720 *transformed into the hypergraphical game $\tilde{G} = (\tilde{I}, \tilde{E}, (\tilde{A}_{(i,\theta_i)})_{(i,\theta_i) \in \tilde{I}}, (\tilde{u}_{(i,\theta_i)}^e)_{e \in \tilde{E}, (i,\theta_i) \in e})$ s.t.:*
 721 $\tilde{I} = \{(i, \theta_i) \mid i \in I, \theta_i \in \Theta_i\}$, $\tilde{E} = (\text{Players}(\theta))_{\theta \in \Theta}$, $\tilde{A}_{(i,\theta_i)} = A_i$,
 722 $\tilde{u}_{(i,\theta_i)}^e(\tilde{\sigma}_e) = \text{BetP}_{(i,\theta_i)}(\theta) \times u_i(\sigma^\theta, \theta)$.

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

```
725
726 Variables (cond : conditioning R Tn) (proper_G : proper_belgame cond).
727 Definition HRTBM_localgame : finType := Tn.
728 Definition HRTBM_plays_in : HRTBM_localgame -> pred HR_player :=
729   fun lg i_ti => lg (projT1 i_ti) == projT2 i_ti.
730
```

731 Local utilities are computed w.r.t. the “pignistic” distribution BetP:

```
732
733 Definition HRTBM_u : ∀ lg, HRTBM_localprof lg -> HRTBM_localplayer lg -> R :=
734   fun lg p x => let (i_ti, _) := x in let (i, ti) := i_ti in
735     let betp := BetP (cond G.1 (event_ti ti) (is_revisable proper_G ti)) in
736     dist betp lg * G.2 (HRTBM_mkprofile p) lg i.
737 Definition HRTBM : cgame R HR_action := hg_game HRTBM_u.
738
```

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

```
742
743 Theorem HRTBM_correct (i : I) (ti : T i) (p : iprofile A T) :
744   belgame_utility fTBEU proper_G p ti = HRTBM (flatten p) (existT _ i ti).
745 Theorem HRTBM_eqNash (p : iprofile A T),
746   BelG_Nash_equilibrium fTBEU proper_G p <=> Nash_equilibrium HRTBM (flatten p).
747
```

748 7 Conclusion and Perspectives

749 In this paper, a Coq/SSReflect formalization of Bel games has been presented. It provides a
 750 generic class of games of incomplete information, built upon the Dempster-Shafer theory of
 751 belief functions. This framework makes it possible to capture (lack of) knowledge better than
 752 usual game models based on probability. Following Howson’s and Rosenthal’s approach, three
 753 different algorithms transforming such incomplete games into standard complete-information
 754 games have been formalized, and the fact that these transforms preserve equilibria (as long

755 as we comply with Table 4) has been formally verified. Thus, Bel games can ultimately be
 756 solved using state-of-the-art, effective algorithms for complete games.

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

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

771 ——— References ———

- 772 1 Reynald Affeldt and Cyril Cohen. Measure construction by extension in dependent type theory
 773 with application to integration, 2022. URL: <https://arxiv.org/abs/2209.02345>.
- 774 2 Reynald Affeldt, Jacques Garrigue, and Takafumi Saikawa. A library for formalization of
 775 linear error-correcting codes. *Journal of Automated Reasoning*, 64(6):1123–1164, 2020.
- 776 3 Nahla Ben Amor, Hélène Fargier, Régis Sabbadin, and Meriem Trabelsi. Possibilistic Games
 777 with Incomplete Information. In *Proceedings of the Twenty-Eighth International Joint Confer-*
 778 *ence on Artificial Intelligence*, pages 1544–1550. ijcai.org, 2019.
- 779 4 Alexander Bagnall, Samuel Merten, and Gordon Stewart. A Library for Algorithmic Game
 780 Theory in SSReflect/Coq. *Journal of Formalized Reasoning*, 10(1):67–95, 2017.
- 781 5 Sylvie Boldo, François Clément, Florian Faissole, Vincent Martin, and Micaela Mayero. A
 782 Coq formalization of Lebesgue integration of nonnegative functions. *Journal of Automated*
 783 *Reasoning*, 66(2):175–213, 2022. URL: <https://doi.org/10.1007/s10817-021-09612-0>.
- 784 6 The Coq Development Team. The Coq Proof Assistant, 2022. URL: [https://doi.org/10.](https://doi.org/10.5281/zenodo.1003420)
 785 [5281/zenodo.1003420](https://doi.org/10.5281/zenodo.1003420).
- 786 7 Arthur P. Dempster. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The*
 787 *Annals of Mathematical Statistics*, 38:325–339, 1967.
- 788 8 Christoph Dittmann. Positional determinacy of parity games. Available at [https://www.](https://www.isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf)
 789 [isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf](https://www.isa-afp.org/browser_info/devel/AFP/Parity_Game/outline.pdf), 2016.
- 790 9 Didier Dubois and Thierry Denoeux. Conditioning in Dempster-Shafer Theory: Prediction vs.
 791 Revision. In *Belief Functions: Theory and Applications - Proceedings of the 2nd International*
 792 *Conference on Belief Functions*, pages 385–392. Springer, 2012.
- 793 10 Didier Dubois and Henri Prade. Focusing vs. belief revision: A fundamental distinction when
 794 dealing with generic knowledge. In *Qualitative and quantitative practical reasoning*, pages
 795 96–107. Springer, 1997.
- 796 11 Hélène Fargier, Érik Martin-Dorel, and Pierre Pomeret-Coquot. Games of incomplete infor-
 797 mation: A framework based on belief functions. In *Symbolic and Quantitative Approaches to*
 798 *Reasoning with Uncertainty - 16th European Conference*, volume 12897 of *Lecture Notes in*
 799 *Computer Science*, pages 328–341. Springer, 2021.
- 800 12 Ruobin Gong and Xiao-Li Meng. Judicious judgment meets unsettling updating: Dilation,
 801 sure loss and simpson’s paradox. *Statistical Science*, 36(2):169–190, 2021.
- 802 13 John C Harsanyi. Games with Incomplete Information Played by “Bayesian” Players, I–III.
 803 Part I. The Basic Model. *Management Science*, 14(3):159–182, 1967.

- 804 **14** Johannes Hölzl and Armin Heller. Three chapters of measure theory in Isabelle/HOL. In
805 *Interactive Theorem Proving*, pages 135–151, Berlin, Heidelberg, 2011. Springer Berlin Heidel-
806 berg.
- 807 **15** Joseph T Howson Jr and Robert W Rosenthal. Bayesian Equilibria of Finite Two-Person
808 Games with Incomplete Information. *Management Science*, 21(3):313–315, 1974.
- 809 **16** Jean-Yves Jaffray. Linear Utility Theory for Belief Functions. *Operations Research Letters*,
810 8(2):107–112, 1989.
- 811 **17** Jean-Yves Jaffray. Linear Utility Theory and Belief Functions: a Discussion. In *Progress in*
812 *decision, utility and risk theory*, pages 221–229. Springer, 1991.
- 813 **18** Rida Laraki, Jérôme Renault, and Sylvain Sorin. *Mathematical foundations of game theory*.
814 Springer, 2019.
- 815 **19** Stéphane Le Roux, Érik Martin-Dorel, and Jan-Georg Smaus. An Existence Theorem of Nash
816 Equilibrium in Coq and Isabelle. In *Proceedings Eighth International Symposium on Games,*
817 *Automata, Logics and Formal Verification*, volume 256 of *Electronic Proceedings in Theoretical*
818 *Computer Science*, pages 46–60, 2017.
- 819 **20** Stéphane Le Roux, Érik Martin-Dorel, and Jan-Georg Smaus. Existence of Nash equilibria
820 in preference priority games proven in Isabelle. In *Kurt Gödel Day and Czech Gathering of*
821 *Logicians*, 2021.
- 822 **21** Pierre Lescanne and Matthieu Perrinel. "backward" coinduction, Nash equilibrium and the
823 rationality of escalation. *Acta Informatica*, 49(3):117–137, 2012.
- 824 **22** Assia Mahboubi and Enrico Tassi. *Mathematical Components*. Zenodo, January 2021.
- 825 **23** Érik Martin-Dorel and Sergei Soloviev. A Formal Study of Boolean Games with Random
826 Formulas as Payoff Functions. In *22nd International Conference on Types for Proofs and*
827 *Programs, TYPES 2016*, volume 97 of *Leibniz International Proceedings in Informatics*, pages
828 14:1–14:22, 2016.
- 829 **24** Oskar Morgenstern and John Von Neumann. *Theory of Games and Economic Behavior*.
830 Princeton University Press, 1953.
- 831 **25** Roger B Myerson. *Game Theory*. Harvard university press, 2013.
- 832 **26** John Nash. Non-Cooperative Games. *Annals of Mathematics*, pages 286–295, 1951.
- 833 **27** Christos H. Papadimitriou and Tim Roughgarden. Computing Correlated Equilibria in
834 Multi-Player Games. *Journal of the Association for Computing Machinery*, 55(3):1–29, 2008.
- 835 **28** Bernard Planchet. Credibility and Conditioning. *Journal of Theoretical Probability*, 2(3):289–
836 299, 1989.
- 837 **29** Pierre Pomeret-Coquot, Hélène Fargier, and Érik Martin-Dorel. Games of incomplete informa-
838 tion: A framework based on belief functions. *International Journal of Approximate Reasoning*,
839 151:182–204, 2022.
- 840 **30** Stéphane Le Roux. Acyclic preferences and existence of sequential Nash equilibria: A formal and
841 constructive equivalence. In *Proc. Theorem Proving in Higher Order Logics, 22nd International*
842 *Conference*, volume 5674 of *Lecture Notes in Computer Science*, pages 293–309. Springer, 2009.
- 843 **31** Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.
- 844 **32** Philippe Smets. Jeffrey’s Rule of Conditioning Generalized to Belief Functions. In *Uncertainty*
845 *in artificial intelligence*, pages 500–505. Elsevier, 1993.
- 846 **33** Philippe Smets and Robert Kennes. The Transferable Belief Model. *Artificial Intelligence*,
847 66(2):191–234, 1994.
- 848 **34** René Vestergaard. A constructive approach to sequential Nash equilibria. *Information. Pro-*
849 *cessing Letters*, 97(2):46–51, 2006. doi:10.1016/j.ipl.2005.09.010.
- 850 **35** Peter Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman & Hall, 1991.
- 851 **36** Elena Yanovskaya. Equilibrium Points in Polymatrix Games. *Lithuanian Mathematical*
852 *Journal*, 8:381–384, 1968.