UNIVERSIDAD ADOLFO IBAÑEZ

MASTER'S THESIS

The Generalized Causal-Effect Score in Databases

Autor:
Felipe Azúa Z.

Thesis Supervisor:
Leopoldo Bertossi D.
Universidad San Sebastián, Chile

Thesis Defense Committee:
Miguel Romero O. (PUC)
Gonzalo Ruz H. (UAI)

Thesis carried out in accordance with the requirements for the degree of
Master of Science in Data Science

of the

Faculty of Engineering and Science

July 24, 2024

UAI | FACULTAD DE INGENIERÍA Y CIENCIAS

UNIVERSIDAD ADOLFO IBAÑEZ

Abstract

Faculty of Engineering and Science

Master of Science in Data Science

The Generalized Causal-Effect Score in Databases

by Felipe Azúa Z.

The *Causal Effect* is a numerical measure used to uncover and analyze causal relationships between variables. In particular, it deals with interventions on variables and their effects in the broad areas of observational studies and structural causality. So far, only preliminary attempts have been made to use causal effect as an attribution score in data management, to measure the causal strength of tuples for query answering in databases. In this work, we introduce, generalize and investigate the so-called *Causal-Effect Score* as a numerical measure of explanatory relevance for database tuples in relation to queries posed to classical and probabilistic databases.

Keywords: Causality, Explainability, Probablistic Databases, Databases.

Acknowledgements: Felipe Azúa Z. has been supported by the SKEMA Business School Canada, and the Millennium Institute for Foundational Research on Data (IMFD, Chile). Leopoldo Bertossi D. has been financially supported by the SKEMA Business School Canada, the IMFD, and NSERC-DG 2023-04650. Part of this work was done while Felipe Azúa Z. was visiting the "Laboratory of Informatics, Modelling and Optimization of the Systems" (LIMOS), at U. Clermont-Ferrand, France.

Contents

Chapter 1

Introduction

Explanations belong to what users could expect from a data management system, for example, for the answers to a query, for a violation of an integrity constraint, etc. The same applies to results from automated systems as found in artificial intelligence (AI), and machine learning (ML). Explanations may come in different forms, in particular, as an *attribution score*, that is, a quantitative degree of relevance of a piece of data in a database (Bertossi, 2023d), or feature values in input entities in the case of AI and ML (Bertossi, 2023a).

Usually, these kinds of explanations are provided at a *local* level, e.g. for individual tuples in a database (DB) to which a query has been posed, or for features values for a particular entity under classification. This work concentrates on *local attribution scores* as explanations in data management.

Several scores have been proposed and investigated in data management. Some of them, such as *Responsibility* and *Causal-Effect*, are explicitly based on the notions of *counterfactual intervention* and *actual causality* (Halpern & Pearl, 2005; Halpern, 2015, 2016). Counterfactual intervention can be seen as hypothetical changes performed on a (causal) model, to identify other changes. By doing so, one can detect cause-effect relationships.

In DBs, *Responsibility* (Chockler & Halpern, 2004) has been applied to quantify the relevance of individual tuples, and attribute values in them, for a query result (Meliou, Gatterbauer, Moore, & Suciu, 2010; Meliou, Gatterbauer, Halpern, et al., 2010; Bertossi & Salimi, 2017b, 2017a; Bertossi, 2021). *Responsibility* has been extended as the Resp-score, and applied to provide explanations in ML (Bertossi, Li, Schleich, Suciu, & Vagena, 2020; Bertossi, 2023a, 2023c).

Another score that has been applied in data management is the *Shapley value* of *Coalition Game Theory* (Shapley, 1953; Roth, 1988), as a measure of contribution of individual players to a shared wealth or game function. In the case of DBs, the players are the tuples and the game function is the Boolean or aggregate query (Livshits, Bertossi, Kimelfeld, & Sebag, 2021b, 2021a; Deutch, Frost, Kimelfeld, & Monet, 2022; Bertossi, Kimelfeld, Livshits, & Monet, 2023). The Shapley value emerges as the only measure that satisfies a given set of desired properties (Roth, 1988). A related measure, also used in coalition game theory, is the *Banzhaf Power Index* (Banzhaf III, 1964), which has also been applied in data management, as the *Banzhaf Score* (Livshits et al., 2021b; Abramovich, Deutch, Frost, Kara, & Olteanu, 2024).

The *Causal-Effect Score* (CES) can be traced back to *causality in observational studies* (Rubin, 1974; Holland, 1986), where one usually cannot predefine and build control groups, but they have to be recovered from the available data (Gelman & Hill, 2007; Pearl, 2009; Roy & Salimi, 2023).

In (Salimi, Bertossi, Suciu, & Van den Broeck, 2016), a particular version of the CES was first used in data management as an alternative to the responsibility score, when the latter did not provide intuitive results. In order to use CES to assess the relevance of tuples in a database (DB) D for the result of a query, D is first transformed into a *tuple-independent probabilistic database* (TID) D^p (Suciu, Olteanu, Ré, & Koch, 2011) under the uniform distribution on tuples. Next, the CES is defined on the basis of D^p, and is computed by appealing to the *lineage* of the query (Suciu et al., 2011). This definition and construction can accommodate both *endogenous* and *exogenous* tuples, which also appear with actual causality, *Responsibility*, and the Shapley value in DBs (Meliou, Gatterbauer, Moore, & Suciu, 2010; Bertossi & Salimi, 2017b; Livshits et al., 2021b).

The application of the CES to aggregate queries was shown in (Salimi et al., 2016) only with an example, and a more complete investigation remains open. Actually, the CES seems to be much more appropriate for explaining results for aggregate queries than the responsibility score, which would only detect, under a counterfactual intervention, e.g. a tuple deletion, if there is a change or not of a numerical value, no matter how small (an issue also mentioned in (Meliou, Gatterbauer, Halpern, et al., 2010)). The CES would take, by definition, the amount of the change into consideration.

In our research, we propose and investigate an extension of the CES in DBs to the case where the probability distributions on tuples is arbitrary. Actually, it can be applied to general probabilistic DBs. This so-called *Generalized Causal-Effect Score* (GCES) is interesting in several situations, e.g. when: (a) One cannot assume independence or a uniform distribution on tuples, in particular *block-independent* PDBs (Suciu et al., 2011). (b) Additional domain semantics or domain knowledge is available and should be taken into account, e.g. integrity constraints on the underlying relational DB (Bertossi & Salimi, 2017a). (c) There are explicit correlations among database tuples (Sen & Deshpande, 2007; Kanagal & Deshpande, 2010). (d) There are quantitative or qualitative stochastic (in)dependencies among attributes provided by a Bayesian or causal network (Sen, Deshpande, & Getoor, 2009).

The application and analysis of attribution scores in the case of stochastic dependencies among variables (e.g. DB tuples, features in ML, etc.) is a relevant problem. In this case, each tuple can be seen as an instance of a joint distribution which is determined by the stochastic dependencies among attributes.[1]

Along this line, sometimes integrity or application constraints can be compiled into an updated joint distribution that may not be uniform anymore (Bertossi, 2023c; Cifuentes et al., 2024). It is worth mentioning that the behavior of the *Responsibility* score changes when it is defined and computed in the presence of integrity constraints (Bertossi & Salimi, 2017a). Furthermore, it could be the case that the initial DB is a probabilistic one at the very start; or one where different weights are assigned to its tuples.

In our work, we investigate the GCES in terms of its alignment with other scores, its general properties, its computational complexity. We also consider the case of

[1] A particular stochastic version of the Shapley value is *SHAP* (Lundberg & Lee, 2017), that is used in ML to quantify the relevance of a feature value for a classification outcome. Dependencies among feature values could be considered (Aas, Jullum, & Loland, 2021; Cifuentes et al., 2024).

aggregate queries. All the results can be applied to the original CES in (Salimi et al., 2016).

We also investigate some connections of CES and GCES with the other previously mentioned scores. More specifically, taking inspiration from the general properties of the Shapley value (Roth, 1988), we analyze the general properties of the CES, and compare them, in the context of DBs, with those of the Shapley and Banzhaf values, and *Responsibility*. Notice that an analysis of the properties of the latter (as applied to DBs) has not been carried out yet. Accordingly, we also make a contribution in this direction.

It is worth mentioning that, in (Livshits et al., 2021b), it is shown that the Banzhaf score and the CES as defined in (Salimi et al., 2016) coincide for Boolean and aggregation queries. This is used as a starting point for the analysis of general properties of the CES.

More specifically, our work makes the following main contributions:

A. From the Causal-Effect Score (CES) introduced in (Salimi et al., 2016) via an auxiliary uniform and tuple-independent probabilistic database (PDB), we give rise to the "Generalized Causal-Effect Score" (GCES), that can be applied to an arbitrary PDB. In doing so, we introduce a precise definition of a probability distribution modified by a counterfactual intervention.

   The PDB could we given right at the start, or could be auxiliary, created to define a non-uniform case of the CES applied to a relational DB. This latter case can be useful when there are stochastic or relational dependencies among database tuples (Suciu, 2020).

B. We motive and justify the use of the CES as an attribution score for query answering in DBs on the basis that it provides more intuitive results in comparison with, e.g. the Shapley Value and Responsibility. We show that the rankings provided by all these scores are not aligned.

C. We thoroughly investigate the alignment of CES with *Responsibility* and the Shapley value in terms of the query and the database at hand.

D. We investigate the *data complexity* of computing the GCES for an individual tuple given a fixed query and a tuple-independent PDB (TID). The complexity of this problem is shown to coincide with the complexity of computing the probability of a fixed query over a TID (Dalvi & Suciu, 2012). A particularly relevant case is that of the GCES on a non-uniform TID, which we simply call "Causal-Effect Score" (CES).

E. We also provide an axiomatic characterization of GCES. The axioms are inspired by those for the *Banzhaf Power Index* (BPI), which was shown to coincide with the CES when applied to DBs (Livshits et al., 2021b). We show that, for every fixed query and its associated game function, the GCES is the only real-valued function that satisfies the axioms. As a main difference with the classical setting of BPI, each subinstance of the database instance (i.e. subteam of players) may have a different weight.

This thesis is structured as follows. Chapter 2 establishes the problems to be studied. Chapter 3 provides background on databases, causality, and probabilistic databases; and describes related work. Chapter 4 establishes the research context, such as objectives and methodology. Chapter 5 introduces the Generalized Causal Effect Score (GCES). Chapter 6 recalls the *Responsibility* score, the *Banzhaf Power Index*, and the *Shapley value* as used in databases, and introduces some notation and basic results that facilitate their comparison. Chapter 7 investigates the alignment of the (non-generalized) Causal Effect Score with *Responsibilty* and the Shapley value depending on the query and the database at hand. Chapter 8 investigates the computational complexity of computing the GCES. Chapter 9 analyzes the general properties of the GCES, and the other scores, in the light of those of the Shapley value (Roth, 1988). We finalize with Chapter 10, drawing some conclusions and discussing open problems.

Chapter 2

Problem Definition

The goal of explanation scores in data management is to quantify the relevance of individual tuples in a particular instance. Known approaches, like Responsibility (Meliou, Gatterbauer, Moore, & Suciu, 2010), Shapley Value (Livshits et al., 2021b), and the Causal-Effect Score (CES) (Salimi et al., 2016), focus on providing local explanations for individual tuples. However, these scores assume independence and uniform distribution on them, which limits their applications to scenarios where there may exists some dependency or non-uniform distribution in the instances at hand.

This thesis addresses the problem of providing explanations in the context of probabilistic databases by introducing the Generalized Causal-Effect Score (GCES). GCES extends CES to accommodate arbitrary probability distributions over all possible subsets of the database instance at hand, namely, all *possible worlds*. This scores is more suited for context where there may exists some tuple dependency or no uniformity of the tuples.

Additionally, as previously presented, Shapley Value and Responsibility are also scores used in the context of non-probabilistic databases. It can be easily checked that the exact values provided by each are different. However, it could be the case that the *rankings* of all tuples by each of the scores are the same for some class of queries. This problem is addressed in this work, being refer as the *score alignment*.

Another problem addressed in this work is the computational complexity with respect to the size of the database (data complexity) associated with GCES and CES. In (Livshits et al., 2021b), the complexity of the latter was studied for a particular class of Boolean queries. Here, the complexity of both, GCES and CES is studied for a broader range of Boolean queries.

Lastly, this thesis tackles the problem of providing a set of properties for the GCES that uniquely defines it, that is, a set of properties that only the GCES satisfies. This is a common studied problem in the context of coalition game theory. For instance, there exists some set of properties that uniquely defines the Shapley Value (Shapley, 1953) and the Banzhaf Value (Banzhaf III, 1964), two well known values for this context.

As a summary, four problems are being studied in this thesis: (1) explanation score for probabilistic databases, (2) alignment of explanation scores for non-probabilistic databases, (3) complexity of the computation of GCES and CES, and (4) set of properties that defines the GCES.

Chapter 3

Background

3.1 Databases

Consider $\mathcal{S} = \langle R_1, ..., R_k \rangle$ be a database (DB) schema with R_i a relational predicate with arity r_i. $Dom(U)$ denotes the domain of an attribute U in a relation schema, i.e. the set of *constants* the attribute could potentially take. A relation is a finite extension for a relational predicate. It consists of *tuples* of constants, more precisely atoms or facts of the form $R(c_1, \ldots, c_n)$, where R is a relational predicate, and the c_i are constants. A database (instance) D is a collection of finite relations for the given schema. We denote with $Adom(D)$ the set of all constants appearing in tuples in tuples in D. Then, $Adom(D)$ is a finite subset of the union of the attribute domains, and can go beyond the set of constants appearing in D. We commonly use *universal tuple identifiers* (tids), as in the first column of Table 3.1, and refer to whole tuples by using only their tids.

Queries are formulas written in the language of first-order predicate logic (FOPL), $L(\mathcal{S})$, associated to schema $\mathcal{S}$. A *conjunctive query* (CQ) is a formula $\mathcal{Q}(\bar{x})$ of this language of the form $\exists \bar{x}'(A_1 \wedge \cdots \wedge A_l)$, where: (a) $\bar{x}'$ is a string of existentially quantified variables. (b) $\bar{x}$ is a string of *free variables* in the formula (i.e. different from those in $\bar{x}'$). (c) each $A_1, \ldots, A_l$ are predicates from $\mathcal{S}$ instantiated with variables (in $\bar{x} \cup \bar{x}'$), or constants. The A_i are also called *atoms* of the query. A conjunctive query does not have *self-joins* or is *self-join-free* if no predicate in the conjunction appears more than once.

A Boolean query does not have free variables. In particular, we have *Boolean conjunctive queries* (BCQs). Accordingly, we will usually write a BCQ as $\mathcal{Q}$: $\bar{\exists}(A_1, \ldots, A_l)$, indicating the *existential closure* of the conjunction, i.e. existentially quantifying all variables. We will denote with $Var(\mathcal{Q})$ the set of all variables in $\mathcal{Q}$. Another interesting class of queries is that of the *unions of conjunctive queries* (UCQs), in particular, the *unions of Boolean conjunctive queries* (UBCQs).

If $\mathcal{Q}$ is a Boolean query, $D \models \mathcal{Q}$ denotes that DB D satisfies the query. If the query is *open*, i.e. with free variables, say $\mathcal{Q}(\bar{x})$, and $\bar{a}$ is a sequence of constants from the domain, $D \models \mathcal{Q}[\bar{a}]$ denotes satisfaction of the query with the variables replaced by the constants $\bar{a}$. In this case, $\bar{a}$ is an *answer* to the query. With $\mathcal{Q}[D]$ we denote the set of answers to the query from D. If the query is Boolean, then $\mathcal{Q}[D]$ is the empty set (when the query is false) or $\{yes\}$, when the query is true. (Later on we will sometimes write $\mathcal{Q}[D] = 0$ or $\mathcal{Q}[D] = 1$, depending of whether the query is false or true, resp.)

A query $\mathcal{Q}$ is *monotone* when, for every $D_1 \subseteq D_2$, it holds $\mathcal{Q}[D_1] \subseteq \mathcal{Q}[D_2]$. Conjunctive queries and unions thereof are monotone. In this work, we will consider by default only monotone queries, mostly CQs, and numerical aggregations on CQs. Furthermore, all the computational complexity considerations refer to *data complexity*, i.e. in terms of the size of the database at hand.

For a BCQ $\mathcal{Q}$ and a variable v in it, $Atoms(v)$ will denote the set of atoms in $\mathcal{Q}$ where v appears; and $Atoms(\mathcal{Q})$ denotes the set of all atoms in $\mathcal{Q}$.

A special sub-class of BCQs, namely *hierarchical queries* (Suciu et al., 2011), is now recalled. A BCQ query $\mathcal{Q}$ is *hierarchical* if, for any two (existential) variables x and y in it, one of the following holds: (a) $Atoms(x) \subseteq Atoms(y)$, (b) $Atoms(y) \subseteq Atoms(y)$ or (c) $Atoms(x) \cap Atoms(y) = \emptyset$. Otherwise, the query is called *non-hierarchical*. This classification has turned out to be relevant in different situations, most prominently, and for the first time, for query answering in tuple-independent probabilistic databases (TIDs). For queries without self-joins, hierarchical queries can be evaluated in polynomial time, but for non-hierarchical queries, the problem is #P-hard (Suciu et al., 2011).

3.2 Actual Causality and Responsibility in Databases

For causality purposes, some of the tuples in a DB D are considered to be *endogenous*. They can be subject to causal (more precisely, counterfactual) *interventions*, in this case, deletions, insertions, or value updates. The other tuples are *exogenous*, and are taken as given. They may participate in query answering, but they are not subject to interventions. D^{en}, D^{ex} denote the subinstances containing the endogenous, resp. exogenous tuples.

We usually want to explain *why* a given query becomes true or returns a particular answer. A tuple $\tau \in D^{en}$ is a *counterfactual cause* for a BCQ $\mathcal{Q}$ if: $D \models \mathcal{Q}$, and $(D \smallsetminus \{\tau\}) \not\models \mathcal{Q}$. A tuple $\tau \in D^{en}$ is an *actual cause* for a BCQ $\mathcal{Q}$ if there is $\Gamma \subseteq D^{en}$, such that: $D \models \mathcal{Q}$, $(D \smallsetminus \Gamma) \models \mathcal{Q}$, and $(D \smallsetminus (\Gamma \cup \{\tau\})) \not\models \mathcal{Q}$ (Meliou, Gatterbauer, Halpern, et al., 2010). The set Γ is called a *contingency set* for τ, denoted Γ_τ.

The *responsibility* of τ as an actual cause for $\mathcal{Q}$ is defined by $\rho(\tau) := \frac{1}{1+|\Gamma_\tau|}$, where $|\Gamma_\tau|$ is the cardinality of a minimum-size contingency set for τ (Chockler & Halpern, 2004; Meliou, Gatterbauer, Halpern, et al., 2010). When τ is not an actual cause, its responsibility is defined as 0. Notice that every counterfactual cause is an actual cause with an empty contingency set, in which case the responsibility takes the maximum value 1.

Responsibility is an *explanation (or attribution) score* that quantifies the causal contribution of a tuple to a query answer. The data complexity of responsibility in DBs has been investigated in (Meliou, Gatterbauer, Halpern, et al., 2010; Bertossi & Salimi, 2017b, 2017a). This score was adapted and used in (Bertossi, 2023c) to provide explanations for results from binary classifiers with binary features, and extended in (Bertossi et al., 2020) for possibly non-binary features (see also (Bertossi, 2023a)).

The *Causal-Effect Score* (CES) is another explanation score that has been applied in data management. It was used in (Salimi et al., 2016) as an alternative to responsibility when the latter does not provide the most intuitive results. This will be introduced in more detail in Chapter 5, and is the main subject of this work. The complexity of CES has been partially studied in (Livshits et al., 2021b).

R	A	B	C	P
τ_1	a_1	b_1	c_1	p_1
τ_2	a_2	b_2	c_2	p_2
τ_3	a_3	b_3	c_3	p_3
τ_4	a_4	b_4	c_4	p_4
τ_5	a_5	b_5	c_5	p_5

R	A	B	C	P
τ_1	a_1	b_1	c_1	p_1
τ_2	a_2	b_2	c_2	p_2
τ_3	a_3	b_3	c_3	p_3
τ_4	a_4	b_4	c_4	p_4
τ_5	a_5	b_5	c_5	p_5

TABLE 3.1: (a) TI PDB and (b) BI PDB

3.3 Probabilistic Databases

We recall here what we need about probabilistic databases (PDBs). See (Suciu et al., 2011) for a deeper treatment. We can conceive a PDB as a regular relational database where relations have an extra attribute accommodating probability values associated to the corresponding tuples. Table 3.1 shows two PDBs. For example, the first tuple in the relation R in Table 3.1(a) indicates that tuple τ_1 is in the DB with probability p_1.

The semantics of a PDB, D^p, is a *possible world semantics*, in this case a collection $\mathcal{W}$ of regular DBs, W, whose relations, R_W, do not have a probabilistic attribute, but a global associated probability, $p(R_W)$. Different semantics differ on how the instances D are built and how probabilities are assigned to their relations.

The most common case is that of a *tuple-independent* PDBs (TIDs). In Table 3.1(a), each tuple is in the DB independently from the other tuples, including tuples in other relations. Each tuple has a probability assigned. In a possible world $W \in \mathcal{W}$, the corresponding (non-probabilistic) relation R_W will contain only some of the tuples in R, and the probability associated to R_W is defined by: $p(R_W) := \Pi_{\tau_i \in R_W} p_i \times \Pi_{\tau_j \in (R \smallsetminus R_W)} (1 - p_j)$. For example, if in a possible world W relation R_W contains only tuples τ_1 and τ_3, it will have the probability $p_1 \times (1 - p_2) \times p_3 \times (1 - p_4) \times (1 - p_5)$. Tuple independence beyond the single-relation level leads to the overall probability assigned to a possible world: $p(W) := \Pi_{R_W} p(R_W)$.

A more general case is that of *block-independent* PDBs (BIDs). Tuples (of a same relation) are separated in blocks of mutually exclusive tuples. We will assume by default, unless otherwise stated, that, within a block, the tuples' probabilities add up to 1. Table 3.1(b) shows a BID with a block with 2 tuples, and next one with 3. The semantics is similar to the previous one, but now the blocks are independent. When a regular relation R_W is built from the probabilistic relation R, only one tuple is chosen from each block. The probability associated to R_W is defined by: $p(R_W) := \Pi_{\tau_i \in R_W} p_i$, i.e. the product of the probabilities of the tuples chosen from the blocks. For example, if tuple τ_2 is chosen from the first block, and τ_4 from the second, $p(R_W) = p_2 \times p_4$. Similar to the TID case, an overall probability can assigned to a possible world: $p(W) := \Pi_{R_W} p(R_W)$.[1]

Remark 1. *More generally, and for the purpose of this work, given a relational instance D, a PDB associated to D, usually denoted with D^p, can be identified with a discrete* probability space *$\langle \mathcal{W}(D), p^D \rangle$, where $\mathcal{W}(D)$ is a collection of* possible worlds *W that are subinstances of D, p^D is defined on $\mathcal{W}(D)$, and $\sum_{W \in \mathcal{W}(D)} p^D(W) = 1$. When D is clear from the context, we simply write $\mathcal{W}$ and p. Unless otherwise stated, if there is a partition (D^{en}, D^{ex}) of D, we assume that, for every $W \in \mathcal{W}$ with $D^{ex} \not\subseteq W$, it holds $p^D(W) = 0$.*

[1] We could relax the conditions by choosing *at most* one tuple per block, $\mathcal{B}$, and having the sum of probabilities $p_{\mathcal{B}}$ for $\mathcal{B}$ possibly smaller that 1. In case no tuple is chosen, $\mathcal{B}$ contributes with $(1 - p_{\mathcal{B}})$ to the product.

Since, D is finite, $\mathcal{W}(D)$ and the W are finite. ■

A query on a PDB becomes a random variable. If it is Boolean, it takes the values 0 or 1 on the outcomes $W \in \mathcal{W}$. There are several query-answering semantics that have been considered in the literature. We briefly mention those that become relevant in our work. Let $\mathcal{Q}$ be a query for schema $\mathcal{S}$.

(a) *Probabilistic Answer*: If $\mathcal{Q}$ is a BCQ, the probability of $\mathcal{Q}$ (being true) is $P(\mathcal{Q}) := \sum_{W\in\mathcal{W}:\, W\models\mathcal{Q}} p(W)$. If $\mathcal{Q}(\bar{x})$ is an open query, and $\bar{a}$, a sequence of constants, $P(\bar{a}) := P(\mathcal{Q}[\bar{a}])$. Under this semantics, each answer comes with a probability (of being an answer). Notice that for a Boolean query, $P(\mathcal{Q}) = \mathbb{E}(\mathcal{Q})$, which invites us to define, for an *aggregate query* $\mathcal{Q}$: $\mathcal{Q}[D] := \mathbb{E}(\mathcal{Q}) = \sum_{W\in\mathcal{W}} p(W) \times \mathcal{Q}[W]$.

(b) *Most Probable Answer*: A sequence of constants $\bar{a}^\star$ is a most probable answer to $\mathcal{Q}$ iff $\bar{a}^\star = \text{argmax}_{\bar{a}} \, \text{P}(\mathcal{Q}[\bar{a}])$, denoted $\bar{a}^\star = \mathit{MPA}(\mathcal{Q})$. In this case, we obtain all the answers to $\mathcal{Q}$ that have the same maximum probability. If the query is Boolean, 0 or 1 become MPAs.

(c) *Answers from Most Probable DB* (Gribkoff, Van den Broeck, & Suciu, 2014): $\mathit{Ans}_{\mathcal{W}}(\mathcal{Q}) := \{\bar{a} \mid W^\star \models \mathcal{Q}[\bar{a}]\}$, where $W^\star := \text{argmax}_{W\in\mathcal{W}} \, p(W)$. In this case, we obtain fully classical answers if there is a single most probable database (MPD). If there are several MPDs, we may choose those that are *certain*, that is in common to all the MPDs, or those that are *possible* if they come from some MPD.

3.4 Query Lineage

We introduce the notion of lineage (Suciu et al., 2011; Salimi et al., 2016) and its notation by means of an example.

Example 1. Consider the database instance D consisting only of endogenous tuples, i.e. $D^{ex} = \emptyset$, and the Boolean CQ $\mathcal{Q}\colon \exists x \exists y (S(x) \wedge R(x,y) \wedge S(y))$. $\mathcal{Q}$ is true in D: $D \models \mathcal{Q}$.

D

R
$\langle c,b\rangle$
$\langle a,d\rangle$
$\langle b,b\rangle$
$\langle e,f\rangle$

S
$\langle a\rangle$
$\langle b\rangle$
$\langle c\rangle$

The lineage of query $\mathcal{Q}$ on D is the propositional formula:

$$\mathcal{L}(\mathcal{Q},D) \;:=\; (X_{S(c)} \wedge X_{R(c,b)} \wedge X_{S(b)}) \vee (X_{S(b)} \wedge X_{R(b,b)}).$$

Here, each X_τ is a propositional variable associated to a ground atom τ of the first-order language associated to the database schema (that includes the finite attribute domains); i.e. a tuple that can be or not be in a database instance D. We could built a query lineage using all these atoms τ, even those not in the instance at hand D.

In this example, given that the query is monotone, we have kept only those atoms X_τ that are true in that $\tau \in D'$. So, it is the query lineage instantiated on D. ■

More generally, the query lineage $\mathcal{L}(\mathcal{Q},D)$ is a propositional formula for which D and each of its subinstances D' act as (or determine) a truth assignment: $\sigma^{D'}(X_\tau) = 1$ iff $\tau \in D'$. $\mathcal{L}(\mathcal{Q},D')$ is *true* (under assignment $\sigma^{D'}$) iff $D' \models \mathcal{Q}$. $\mathcal{L}(\mathcal{Q},D')$ is *true* under assignment $\sigma^{D'}$ iff $D' \models \mathcal{Q}$. With monotone queries we are only interested in subintances of D since only they may switch the query from true to false.

When D is partitioned into $D^{ex} \cup D^{en}$, we will consider only subintances D' of D that contain D^{ex}. Then, $\sigma^{D'}(X_\tau) = 1$, for every $\tau \in D^{ex}$.

Notice that the lineage of a Boolean CQ is always a propositional formula in MON-DNF, i.e. a formula in disjunctive normal form where all atoms appear positively.

3.5 Related Work

The use of the Shapley value in AI can be traced back at least to its application to measure the contribution of individual formulas to the inconsistency of a knowledge base (Hunter & Konieczny, 2010).

The Shapley value as an attribution or explanation score has been used, more recently, in machine learning (ML), most commonly as the *SHAP* score (Lundberg & Lee, 2017) (see (Bertossi, 2023b) for more results and references on *SHAP*). It quantifies the contribution of feature values in an input entity to an ML-based system. There is a vast literature on the use of (diverse variants of) *SHAP* in ML. There has been some criticism to the *SHAP* score, see, e.g. (Huang & Marques-Silva, 2024).

The use of the Shapley value and the Banzhaf-Power-Index (BPI) in data management started with the conference version of (Livshits et al., 2021b), where it was used to quantify the contribution of individual tuples to a query answer. This works has attracted the attention of several researchers and has become an active area of research. Along this line of research, in (Deutch et al., 2022), the efforts have been concentrated on computational aspects of the Shapley value as applied to query answering.

In (Arad, Deutch, & Frost, 2022), the research is about the computation of the *rankings* induced by the Shapley values rather than on the values themselves. In (Davidson et al., 2022) the interest is placed on the combination of *data provenance* and the Shapley values, for the computation of the latter. In (Abramovich et al., 2024) similar research has delved more deeply into experimental results around the application of the BPI to query answering.

In (Kara, Olteanu, & Suciu, 2024), the problem of computing the Shapley value of variables in Boolean circuits (but not in its *SHAP* version as in (Arenas, Barcelo, Bertossi, & Monet, 2023)) is connected with query evaluation in probabilistic databases, obtaining dichotomy results (similar to those in (Dalvi & Suciu, 2012)) for the complexity of Shapley value computation for query answering in classical databases.

Applications of the responsibility score in data management (DM) and ML have been less explored than those of the Shapley value. The main articles on the use of responsibility in DM are (Meliou, Gatterbauer, Halpern, et al., 2010; Bertossi & Salimi, 2017b, 2017a; Bertossi, 2021); and (Bertossi, 2023c; Bertossi et al., 2020) in ML. In all those papers, semantic and computational problems were addresses, but no analysis of its general properties has been done. Close to responsibility as used in DM, we find the notion of *resilience*, whose computational aspects have been investigated (see (Makhija & Gatterbauer, 2023) and references therein).

Despite its widespread and general use in causality, particularly in *observational studies*, the *causal effect*, as a measure of causal strength -and the best of our knowledge- had not been explored in DM apart until it was used in (Salimi et al., 2016), in its simplest form, as an alternative to responsibility. The connection with the BPI was first established in (Livshits et al., 2021b). Its general properties when used as attribution score for query answering in DBs have not been investigated yet.

Chapter 4

Research Context

4.1 Objectives

The general objective of this thesis is to develop and analyze the Generalized Causal-Effect Score (GCES) for probabilistic databases, and thus provide an explanation score that accounts for dependencies and non-uniform distributions. In particular, this research specific objectives are:

- Extend the Causal-Effect Score (CES) to accommodate arbitrary probability distributions over possible worlds in a database.
- Investigate the alignment of CES with other explanation scores, such as Shapley Value and Responsibility, in non-probabilistic databases.
- Analyze the computational complexity (w.r.t. the size of the database instance) of computing GCES and CES.
- Define a set of properties for GCES that uniquely defines it.

4.2 Methodology

The research methodology mainly consisted in reviewing the existing results (in conjunction with the methodology by which they were obtained) of the research topics, and then establish, by mathematical proofs, the results for each. In particular, the following methodology was used for each research topic: (a) Definition of GCES: The formal definition of this new score is obtained starting by first identifying that the CES induces a probability distribution on the set of *possible worlds* of the instance at hand, and then allowing the use of any arbitrary probability distribution on said set; (b) Alignment of scores: some of the proofs of these results are based on techniques used in (Livshits et al., 2021b) and (Dalvi & Suciu, 2007), which can be found in the respective chapter; (c) Complexity of GCES and CES: These results were obtained by applying one-to-one reductions (Karp reductions) between the problem of computing GCES and CES with a well studied problem in probabilistic databases: the query evaluation problem; and (d) Properties of GCES: based on the axiomatic characterization given for the Banzhaf Power Index in (Dubey & Shapley, 1979), a new set of properties is given for the GCES. The proof that shows the existence and uniqueness of the GCES is also inspired by techniques used in the cited work.

Chapter 5

The Causal-Effect Score in Databases

In the context of databases (DBs), the *Causal-Effect Score* of a tuple was introduced in (Salimi et al., 2016) as a measure of its contribution to the answer to a query. This quantity is defined through a the effect of a *counterfactual intervention* in a causal model, which involves the query and the DB.

5.1 Tuple-Interventions on a PDB

In the context of a database D and its subinstances, and for our purposes, *interventions* will be of the form $do(\tau \text{ in})$ and $do(\tau \text{ out})$, with the intuitive meaning that tuple τ is *made true*, i.e. it is inserted into the database at hand (if it is not already in it). Similarly, $do(\tau \text{ out})$ means that τ is *made false*, i.e. removed from the database. We will apply interventions only with endogenous tuples. In principle, τ could be outside D, but it is declared as potentially endogenous. Interventions are commonly applied in the context of a query, to detect if making tuples true or false affect the answer. They can also be applied with sets of tuples $\mathcal{T}$, e.g. $do(\mathcal{T} \text{ in})$.[1]

In the following we will have expressions of the form $P(\mathcal{Q} = 1 \mid do(\tau \text{ in}))$, where P is the distribution of a PDB associated to an instance D, and $\mathcal{Q}$ is a BCQ. Intuitively, it means "the probability of the query being true given that tuple τ is true". However, this notation, despite its omnipresence, may be misleading in that it suggests a conditional probability, which strictly speaking is not. More precisely, and as every conditional probability, that redefines a given probability distribution, $P(\cdot \mid do(\tau \text{ in}))$ can be seen as a modification of distribution the original P. Similar considerations apply to the notation $P(\cdot \mid do(\tau \text{ out}))$. Definition 1 makes all this precise, by building on Remark 1.

Definition 1. Given an instance D, an associated PDB $D^p = \langle \mathcal{W}(D), p^D \rangle$, and $\tau \in D$:
(a) $D^p(do(\tau \text{ in})) := \langle \mathcal{W}^{+\tau}, p^{+\tau} \rangle$, with $\mathcal{W}^{+\tau} := \{W \cup \{\tau\} \mid W \in \mathcal{W}\}$; and, for $W' \in \mathcal{W}^{\tau+}$, $p^{+\tau}(W') := \sum_{W \cup \{\tau\} = W'} p(W)$.
(b) $D^p(do(\tau \text{ out})) := \langle \mathcal{W}^{-\tau}, p^{-\tau} \rangle$, with $\mathcal{W}^{-\tau} := \{W \smallsetminus \{\tau\} \mid W \in \mathcal{W}\}$; and, for $W' \in \mathcal{W}^{\tau-}$, $p^{-\tau}(W') := \sum_{W \smallsetminus \{\tau\} = W'} p(W)$.

(c) For a Boolean query $\mathcal{Q}$:

[1] It is common to apply interventions on variables of a causal model (Pearl, 2009). In the case of databases, to the *lineage of the query* with the random propositional variables X_τ, which are made *true* or *false* via $do(X_\tau = 1)$ or $do(X_\tau = 0)$ (see Section 3.4).

$$\begin{aligned} P(\mathcal{Q}=1 \mid do(\tau \ \text{in})) &:= p^{+\tau}(\{W' \in \mathcal{W}^{+\tau} \mid W' \models \mathcal{Q}\}) \\ P(\mathcal{Q}=0 \mid do(\tau \ \text{in})) &:= p^{+\tau}(\{W' \in \mathcal{W}^{+\tau} \mid W' \not\models \mathcal{Q}\}) \\ P(\mathcal{Q}=1 \mid do(\tau \ \text{out})) &:= p^{+\tau}(\{W' \in \mathcal{W}^{-\tau} \mid W' \models \mathcal{Q}\}) \\ P(\mathcal{Q}=0 \mid do(\tau \ \text{out})) &:= p^{+\tau}(\{W' \in \mathcal{W}^{-\tau} \mid W' \not\models \mathcal{Q}\}). \end{aligned} \tag{5.1}$$

■

Remark 2. *(a) On the RHS of (5.1), it holds (and similarly for the other cases):*

$$p^{+\tau}(\{W' \in \mathcal{W}^{+\tau} \mid W' \models \mathcal{Q}\}) = \sum_{W \in \mathcal{W},\ W \cup \{\tau\} \models \mathcal{Q}} p^D(W).$$

(b) In (5.1), P is the probability induced by $p^{+\tau}$ on the range of the query, in this case, $\{0,1\}$.

(c) Given a PDB D^p associated to an instance D, and a tuple τ, one can always compute the probability of τ being in D (seen through the PDB D^p), namely:

$$P(\tau) := \sum_{W \in \mathcal{W}(D), \tau \in W} p^D(W). \tag{5.2}$$

(d) It is easy to verify that, for an endogenous tuple τ:

$$\begin{aligned} P(\tau \mid do(\tau \ \text{in})) &:= \sum_{W' \in \mathcal{W}^{+\tau}, \tau \in W'} p^{+\tau}(W') = 1; \ \textit{and} \\ P(\tau \mid do(\tau \ \text{out})) &:= \sum_{W' \in \mathcal{W}^{-\tau}, \tau \in W'} p^{-\tau}(W') = 0, \end{aligned}$$

which captures the original intuition.

(e) For a TID D and two tuples $\tau, \tau' \in D^{en}$, the following holds:

$$P(\tau' \mid do(\tau \ \text{in})) = P(\tau' \mid do(\tau \ \text{out})) = p^D(\{\tau'\})$$

That is, an intervention $do(\tau \ \text{in})$ ($do(\tau \ \text{out})$, resp.) in a TID translates in changing the probability of τ to 1 (0, resp.) and leave all other probabilities unchanged. ■

The same definitions can be applied to a scalar aggregate query $\mathcal{Q}$, which also becomes a random variable over a PDB. Using (slightly extended) Datalog notation, they are of the form: $Ans^{\mathcal{Q}}(aggr(x)) \leftarrow Body(x, \bar{x})$, with $Body(x, \bar{x})$ a conjunction of atoms, $x \notin \bar{x}$, and *aggr* an aggregation function, such as *sum*, *max*, etc. Here, $Ans^{\mathcal{Q}}$ is an auxiliary answer-collecting predicate.[2]

5.2 The Generalized Causal Effect Score

It is not clear how to sensibly apply the responsibility score to aggregate queries (Meliou, Gatterbauer, Halpern, et al., 2010, sec. 3.3). In contrast to Boolean queries, where one switches the value of the query to 1 or 0, with an aggregate query we have the issue of deciding what is a reasonable margin of change in the aggregated amount. If we blindly apply the definition of the responsibility score (see Section 3.1), we would only check if there is a counterfactual change in the aggregated value, no matter how small, and every tuple would have the same responsibility.

[2]All this applies "componentwise" to aggregate queries with group-by, but we do not treat them.

E	A	B
τ_1	a	b
τ_2	a	c
τ_3	c	b
τ_4	a	d
τ_5	d	e
τ_6	e	b

S	A	C
τ_7	a	1
τ_8	a	2
τ_9	b	0
τ_{10}	a	3
τ_{11}	b	1
τ_{12}	b	10

TABLE 5.1: A DB D with two relations.

Example 2. Consider the DB D in Table 5.1, and the scalar aggregate query $\mathcal{Q}$ defined by: $Ans^{\mathcal{Q}}(sum(y)) \leftarrow S(x,y)$. Its answer is 17. When $Dom(B) \subseteq \mathbb{R}^+$, this is a monotone query.

If the interventions are tuple deletions, all the tuples but τ_9 would have the same responsibility of $\frac{1}{6}$. Tuple τ_9 would have responsibility 0 since it does not contribute to the answer. However, if interventions were updates of attribute values, specially numerical ones, even τ_9 could have a non-zero resposibility. ■

CES was first applied to DBs in (Salimi et al., 2016), showing that it provides more intuitive and sensible results than the Responsibility Score (see Example 3). Furthermore, the use of CES for aggregate queries makes perfect sense. We will retake the responsibility score and the CES in Chapter 6.

In (Salimi et al., 2016), in order to apply the CES, the original DB was converted into a uniform TID. However, the CES can be generalized by consider any PDB associated to the relational instance D at hand, as in Remark 1. Accordingly and for the following, we start considering a general PDB $D^p = \langle \mathcal{W}(D), p^D \rangle$.

Definition 2. Let D be a relational instance with PDB $D^p = \langle \mathcal{W}(D), p^D \rangle$, and $\mathcal{Q}$ a Boolean or scalar aggregate query. The *generalized causal effect score* (GCES) of $\mathcal{T} \subseteq D^{en}$ on $\mathcal{Q}$ is:

$$CE^p(D, \mathcal{Q}, \mathcal{T}) := \mathbb{E}(\mathcal{Q} \mid do(\mathcal{T}\ \mathsf{in})) - \mathbb{E}(\mathcal{Q} \mid do(\mathcal{T}\ \mathsf{out})). \quad (5.3)$$

■

Notice that, for a Boolean query, from (5.3) it holds:

$$CE^p(D, \mathcal{Q}, \mathcal{T}) = P(\mathcal{Q} = 1 \mid do(\mathcal{T}\ \mathsf{in})) - P(\mathcal{Q} = 1 \mid do(\mathcal{T}\ \mathsf{out})).$$

Remark 3. Special cases and notation. *We denote with $CE^p(D, \mathcal{Q}, \tau)$ the GCES, for a single tuple τ, and based on an arbitrary PDB D^p associated to an instance D. When the D^p corresponds to a TID associated to D, we write $CE^I(D, \mathcal{Q}, \tau)$. If, in addition, the tuples in D are* uniformly *assigned a probability $\frac{1}{2}$, we write $CE^{UI}(D, \mathcal{Q}, \tau)$, which is the particular case considered in (Salimi et al., 2016; Livshits et al., 2021b).* ■

Example 3. (ex. 2 cont.) Let D be a database instance with the relations E and S in Table 5.1 (used in (Salimi et al., 2016), see also (Bertossi, 2023d) for more details), with their tuples endogenous. Build a *uniform* TID by defining $p(\tau) := \frac{1}{2}$ for every tuple τ. Consider the Boolean query $\mathcal{Q}_1$ asking if there exists a path from a to b according to relation E. It can be expressed in Datalog (and as a disjunction of CQs for a fixed instance), which makes it monotone.

It holds: $CE^{UI}(D, \mathcal{Q}_1, \tau_1) = 0.65625$, $CE^{UI}(D, \mathcal{Q}_1, \tau_2) = CE^{UI}(D, \mathcal{Q}_1, \tau_3) = 0.21875$, and $CE^{UI}(D, \mathcal{Q}_1, \tau_4) = CE^{UI}(D, \mathcal{Q}_1, \tau_5) = CE^{UI}(D, \mathcal{Q}_1, \tau_6) = 0.09375$.

As noticed in (Salimi et al., 2016), these scores significantly differ from the responsibility scores, which are all $\frac{1}{3}$, despite the fact that they make the query true through paths of different lengths. We will retake this example in more detail in Example 6 in Section 6.

Consider now the aggregate query $\mathcal{Q}$ in Example 2. The CES for a tuple $\tau \in S$ is computed using Definition 2. We need to compute the expected value of the query when intervening the tuple τ. Denote $\tau[C]$ the restriction of a tuple τ to attribute C; in this case, the numerical value. Consider that, for any $\tau' \in D$ with $\tau \neq \tau'$, it holds the probability of τ' is the same when intervening τ, i.e. $P(\tau' \mid do(\tau \text{ in})) = P(\tau' \mid do(\tau \text{ out})) = p^D(\{\tau'\})$, and for τ, it holds $P(\tau \mid do(\tau \text{ in})) = 1$ and $P(\tau \mid do(\tau \text{ out})) = 0$. Now, and since the aggregate query $\mathcal{Q}$ is just adding the value of the attribute C from each tuple, the CES can be computed as follows:

$$\begin{aligned}
CE^{UI}(D, \mathcal{Q}, \tau_7) &= \mathbb{E}(\mathcal{Q} \mid do(\tau_7 \text{ in})) - \mathbb{E}(\mathcal{Q} \mid do(\tau_7 \text{ out})) \\
&= \sum_{\tau' \in S} \tau'[C] \times P(\tau' \mid do(\tau_7 \text{ in})) - \sum_{\tau' \in S} \tau'[C] \times P(\tau' \mid do(\tau_7 \text{ out})) \\
&= \left(\sum_{\tau' \in S \setminus \{\tau_7\}} \tau'[C] \times p^D(\{\tau'\}) \right) + \tau_7[C] \times 1 \\
&\qquad - \left[\left(\sum_{\tau' \in S \setminus \{\tau_7\}} \tau'[C] \times p^D(\{\tau'\}) \right) + \tau_7[C] \times 0 \right] \\
&= \tau_7[C] = 1.
\end{aligned}$$

This result is inline with the intuition: the average expected contribution of tuple τ_7 for a query that is adding up all tuples in the relation would be the attribute value of the tuple itself, $\tau_7[C]$. ■

In (Salimi et al., 2016) its was established that, for a Boolean monotone query $\mathcal{Q}$, a database D, and $\tau \in D^{en}$, an endogenous tuple, τ is an *actual cause* of $\mathcal{Q}$ in D iff $CE^{UI}(D, \mathcal{Q}, \tau) > 0$. We will now illustrate the definition and computation of the GCES.

Example 4. (ex. 3 cont.) Consider query $\mathcal{Q}_1$ on relation E. All tuples in the database instance are considered to be endogenous. In order to create a PDB D^p, let's assign the following probabilities to the possible worlds (restricted to relation E): For $W_1 = \{\tau_1, \tau_3, \tau_4, \tau_6\}$, $p(W_1) := 0.20$; for $W_2 = \{\tau_1, \tau_2, \tau_3\}$, $p(W_2) := 0.25$; for $W_3 = \{\tau_2, \tau_3, \tau_6\}$, $p(W_3) := 0.15$; for $W_4 = \{\tau_2, \tau_6\}$, $p(W_4) := 0.40$; and for any other $W \subseteq E$, $p(W) := 0$.

Notice that we are starting directly with a probability distribution over the possible worlds, as opposed to starting with probabilities assigned to individual tuples. As in Remark 2(c), we can compute their probabilities: For $\tau \in E$, $P(\tau) := P(\{W \in \mathcal{W} \mid \tau \in W\})$. We obtain: $P(\tau_1) = 0.2 + 0.25 = 0.45$, $P(\tau_2) = 0.8$, $P(\tau_3) = 0.8$, $P(\tau_4) = 0.2$, $P(\tau_5) = 0$, $P(\tau_6) = 0.75$. Now, if we compute the probabilities of the possible worlds using these tuple probabilities assuming independence, we obtain, e.g. for W_1: $P'(W_1) := 0.2 \times (1-0.8) \times 0.8 \times 0.2 \times (1-0) \times 0.75 = 0.00288$, which shows that D^p we started with is not a TID.

Now, consider the new probability distribution $P(\cdot \mid do(\tau_3 \text{ in}))$. In this case, $\mathcal{W}^{+\tau_3} = \{W_1', W_2', W_3'\}$, with $W_1' = W_1$, $W_2' = W_2$ and $W_3' = \{\tau_2, \tau_3, \tau_6\}$. Notice that the number of worlds with non-zero probability is smaller, because $W_3 \cup \{\tau_3\} = W_4 \cup \{\tau_3\} = W_3'$. Then, the probability of W_3' is given by $p^{+\tau_3}(W_3') = p(W_3) + p(W_4) = 0.55$. Something similar happens when we make τ_3 false, removing it from the worlds. Let us show how to compute the causal effect of tuple τ_3.

$$\begin{aligned}
CE^p(D, \mathcal{Q}_1, \tau_3) &:= \mathbb{E}(\mathcal{Q}_1 \mid do(\tau_3 \text{ in})) - \mathbb{E}(\mathcal{Q}_1 \mid do(\tau_3 \text{ out}))\\
&= P(\mathcal{Q}_1 = 1 \mid do(\tau_3 \text{ in})) - P(\mathcal{Q}_1 = 1 \mid do(\tau_3 \text{ out}))\\
&= \sum_{W \in \mathcal{W},\ W \cup \{\tau_3\} \models \mathcal{Q}} p(W) \ - \sum_{W \in \mathcal{W},\ W \smallsetminus \{\tau_3\} \models \mathcal{Q}} p(W).\\
&= (p(W_1) + p(W_2) + p(W_3) + p(W_4)) - (p(W_1) + p(W_2))\\
&= p(W_3) + p(W_4) = 0.55.
\end{aligned}$$

The GCES can also be computed using the new distributions $p^{+\tau_3}$ and $p^{-\tau_3}$ that are obtained through the interventions:

$$\begin{aligned}
CE^p(D, \mathcal{Q}_1, \tau_3) &= \mathbb{E}(\mathcal{Q}_1 \mid do(\tau_3 \text{ in})) - \mathbb{E}(\mathcal{Q}_1 \mid do(\tau_3 \text{ out}))\\
&= \sum_{W' \in \mathcal{W}^{+\tau_3},\ W' \models \mathcal{Q}_1} p^{+\tau_3}(W') \ - \sum_{W^* \in \mathcal{W}^{-\tau_3},\ W^* \models \mathcal{Q}_1} p^{-\tau_3}(W^*).\\
&= \left(p^{+\tau_3}(W_1') + p^{+\tau_3}(W_2') + p^{+\tau_3}(W_3')\right) - \left(p^{-\tau_3}(W_1^*) + p^{-\tau_3}(W_2^*)\right)\\
&= p^{+\tau_3}(W_3') = p(W_3) + p(W_4) = 0.55.
\end{aligned}$$

where $W_1^* = W_1, p^{-\tau_3}(W_1^*) = p(W_1)$ and $W_2^* = W_2 = p^{-\tau_3}(W_2^*) = p(W_2)$.

Notice the difference between $CE^{UI}(D, \mathcal{Q}_1, \tau_3)$ and $CE^p(D, \mathcal{Q}, \tau_3)$. ■

Remark 4. *(a) Definition 2 starts with a relational DB D, which is transformed into a PDB with the purpose of defining the GCES for D. However, nothing in that definition prevents us from starting right away with a PDB, for which the GCES could be defined in the same terms. This is despite the fact that QA semantics for PDBs are as described at the end of Section 3.3.*

(b) In this work, we only consider tuple-based interventions, as opposed to interventions on attribute values. Interventions of this latter kind were investigated in the context of actual causality and responsibility in (Bertossi, 2021). ■

Chapter 6

Revisiting Explanation Scores for Non-Probabilistic Databases

As already mentioned in Chapter 1, there are other attribution scores as explanations for query answering: *Responsibility* and the Shapley Value. We believe that the CES provides more intuitive explanations than the former, and we show an example later in this section.

However, a precise comparison has to be made, which we attempt in Chapter 7. With that goal, We start by recalling the definitions of *Responsibility* and Shapley Value for query answering, for which we first introduce some concepts and notation that will be useful for their definitions and the score comparisons. After that, we establish an interesting connection between the contingency sets for *Responsibility* and CES and Shapley Value (Corollary 6). Finally, we motivate the notion of *score alignment*, that we develop in Chapter 7.

In this section, unless otherwise stated, we consider only the uniform and independent case of CES, i.e. CE^{UI}, to which we simply refer with CES.

Definition 3. For an instance D, a monotone Boolean query $\mathcal{Q}$, assume $D^{ex} \subseteq W \subseteq D$.

(a) W is a *swinging set* for the tuple $\tau \in W$ and $\mathcal{Q}$ if $\mathcal{Q}[W] = 1$, and $\mathcal{Q}[W \setminus \{\tau\}] = 0$. $Swin(D, \mathcal{Q}, \tau)$ denotes the set of swinging sets of τ.

(b) W is a *minimal satisfiable set* if $\mathcal{Q}[W] = 1$ and $\mathcal{Q}[W \setminus \{\tau\}] = 0$ for every $\tau \in W$. $MSS(D, \mathcal{Q})$ denotes the set of all minimal satisfiability sets for D and $\mathcal{Q}$.

(c) W is a *minimal alternating set* if $\mathcal{Q}[D \setminus W] = 0$ and, for every $\tau \in W$, $\mathcal{Q}[D \setminus (W \setminus \{\tau\}))] = 1$. $MAS(D, \mathcal{Q})$ denotes the set of all minimal alternating sets for D and $\mathcal{Q}$. ■

Notice that if, for every $\tau \in W$, $W \in Swin(D, \mathcal{Q}, \tau)$, then $W \in MSS(D, \mathcal{Q})$. Sometimes, when the query or the database are clear from the context, we will simply omit the parameters, e.g. if the query is clear from context we will write $Swin(D, \tau)$, $MSS(D)$ and $MAS(D)$, respectively.

Example 5. (ex. 2 cont.) Consider the database with relation E given in Table 5.1; and the query $\mathcal{Q}_1$ asking if there is a path from a to b. $W = \{\tau_2, \tau_4, \tau_5\}$ is a *swinging set* for τ_6. Moreover, for τ_6: $Swin(D, \mathcal{Q}, \tau_6) = \{\{\tau_4, \tau_5\}, \{\tau_2, \tau_4, \tau_5\}, \{\tau_3, \tau_4, \tau_5\}\}$.

Now, $W' = \{\tau_4, \tau_5, \tau_6\}$ is a *minimal satisfiability set* since if any of the tuples are removed, the query is false. The set of all *minimal satisfiability sets* is $MSS(D, \mathcal{Q}) =$

$\{\{\tau_1\}, \{\tau_2, \tau_3\}, W'\}$, with each world in it built with the tuples that form a path from a to b.

It is easy to check that a *minimal alternating set* is $W^\star = \{\tau_1, \tau_2, \tau_4\}$. The set of all *minimal alternating sets* is $MAS(D, \mathcal{Q}_1) = \{\{\tau_1, \tau_2, \tau_4\}, \{\tau_1, \tau_3, \tau_4\}, \{\tau_1, \tau_2, \tau_5\}, \{\tau_1, \tau_3, \tau_5\}, \{\tau_1, \tau_2, \tau_6\}, \{\tau_1, \tau_3, \tau_6\}\}$. ■

Responsibility. The Responsibility Score was introduced in Section 3.2. Here we introduce useful notation to be be used in this Section.

For a database D, a query boolean $\mathcal{Q}$, and a tuple $\tau \in D$: (a) $Cont(D, \mathcal{Q}, \tau)$ denotes the set of all contingency sets $\Gamma \subseteq D^{en} \setminus \{\tau\}$ for τ. (b) τ is an *actual cause* when $Cont(D, \mathcal{Q}) \neq \emptyset$. and (c) $\rho(D, \mathcal{Q}, \tau) := \frac{1}{1+\min|\Gamma|}$ is the responsibility of τ for the query answer $\mathcal{Q}[D]$, with $\min|\Gamma|$ taken over all contingency set for τ, i.e. $\Gamma \in Cont(D, \mathcal{Q}, \tau)$.

We will omit some of the parameters when are clear from context, for example, if the query is clear from context, we will simply write $Cont(D, \tau)$ and $\rho(D, \tau)$.

The following proposition helps to explain the non-informative result obtained in Example 3 (that will be revisited in Example 6).

Proposition 1. Let D be a database instance and $\mathcal{Q}$ a Boolean monotone query. Let $MSS(D, \mathcal{Q})$ be the collection of minimal satisfiability sets (c.f. Definition 3). If, for every W and $W' \in MSS(D, \mathcal{Q})$, $W \cap W' = \emptyset$, then, the *Responsibility* of a tuple $\tau \in D^{en}$ is:

$$\rho(D, \mathcal{Q}, \tau) = \begin{cases} \dfrac{1}{|MSS(D, \mathcal{Q})|} & \text{, if } \tau \in W \text{ for some } W \in MSS(D, \mathcal{Q}) \\ 0 & \text{, otherwise} \end{cases}$$

Proof. Note that, if there is no $W \in MSS(D, \mathcal{Q})$ such $\tau \in W$, no contingency set exists for τ, and therefore $\rho(D, \mathcal{Q}, \tau) = 0$.

Now, consider $\tau \in W$, for some $W \in MSS(D, \mathcal{Q})$. Let $\Gamma^\star(\tau)$ denote the collection of set Γ that selects one endogenous tuple from each $W' \in MSS(D, \mathcal{Q}) \setminus W$. The cardinality of any of those sets is: $|MSS(D, \mathcal{Q})| - 1$.

It is clear that any set $\Gamma \in \Gamma^\star(\tau)$ is a contingency for the tuple τ, and if any tuple is removed from Γ, then it would not be a contingency of τ. It follows that any of such sets has the minimal cardinality, since for any other contingency $\Gamma' \notin \Gamma^\star(\tau)$, Γ' is a superset of some contingency in $\Gamma^\star(\tau)$. Therefore, $\rho(D, \mathcal{Q}, \tau) = \frac{1}{|MSS(D,\mathcal{Q})|}$. □

The following proposition about responsibility will be used in Proposition 11 from Chapter 7, when comparing CES with *Responsibility* for a particular BCQ.

Proposition 2. Let $\mathcal{Q}$ be a BCQ, and D, D' two instances (for the same schema) for which $MSS(D \cup D', \mathcal{Q}) = MSS(D, \mathcal{Q}) \cup MSS(D', \mathcal{Q})$. For every tuple $\tau \in D$, it holds:

$$\rho(D \cup D', \mathcal{Q}, \tau) = \frac{1}{1 + \min_{\Gamma \in Cont(D,\mathcal{Q},\tau)} |\Gamma| + \min_{\Gamma \in MAS(D',\mathcal{Q})} |\Gamma|}.$$

Proof. Consider a tuple $\tau \in D$. Its *Responsibility* for the answer of $\mathcal{Q}$ in $D \cup D'$ is:

$$\rho(D \cup D', \mathcal{Q}, \tau) = \frac{1}{1 + \min_{\Gamma \in Cont(D \cup D', \mathcal{Q}, \tau)} |\Gamma|}$$

Note that each $\Gamma \in Cont(D \cup D', \mathcal{Q}, \tau)$ can be written as $\Gamma = \Gamma_D \cup \Gamma_{D'}$, where $\Gamma \in Cont(D, \mathcal{Q}, \tau)$, $\Gamma_{D'} \subseteq D'^{en}$ (being D'^{en} the endogenous tuples from D') and $\Gamma_D \cap \Gamma_{D'} = \emptyset$. Now, since the minimum size of Γ is required, Γ_D and $\Gamma_{D'}$ have to have the minimal possible cardinality. This is achieved by minimizing both sets: (a) for Γ_D, $\min_{\Gamma \in Cont(D,\mathcal{Q},\tau)} |\Gamma|$, and (b) for $\Gamma_{D'}$ we have that it must satisfy $\mathcal{Q}[D' \smallsetminus \Gamma_{D'}] = 0$, since $\tau \in D$. Therefore $\Gamma_{D'}$ must belong to $MAS(D', \mathcal{Q})$. It follows that

$$\min_{\Gamma \in Cont(D \cup D', \mathcal{Q}, \tau)} |\Gamma| = \min_{\Gamma_D \in Cont(D, \mathcal{Q}, \tau)} |\Gamma_D| + \min_{\Gamma_{D'} \in MAS(D', \mathcal{Q})} |\Gamma_{D'}|,$$

which, by replacing in the previous expression, is equivalent to the proposition. □

Shapley Value and Banzhaf Power-Index. The Shapley Value (Shapley, 1953) is well-known in *Coalition Game Theory* as a measure of the contribution of individual players to a shared wealth or game function. In (Livshits et al., 2021b), it was adapted and used in databases, to quantify the contribution of tuples to a query answer. For that application, it was defined as follows:

$$Shapley(D, \mathcal{Q}, \tau) = \sum_{S \subseteq (D^{en} \smallsetminus \{\tau\})} \frac{|S|! \cdot (|D^{en}| - |S| - 1)!}{|D^{en}|!} \cdot \Delta(\mathcal{Q}, S, \tau), \quad (6.1)$$

where

$$\Delta(\mathcal{Q}, S, \tau) \;:=\; \mathcal{Q}[S \cup D^{ex} \cup \{\tau\}] - \mathcal{Q}[S \cup D^{ex}]. \quad (6.2)$$

Here, $\mathcal{Q}[S]$ is 0 or 1 when $\mathcal{Q}$ is Boolean and false, resp. true in S. $\mathcal{Q}$ can also be a numerical aggregation, and $\mathcal{Q}[S]$ is its numerical value. The *Bahnzhaf Power-Index* (BPI) (Banzhaf III, 1964) is defined, in the case of databases and queries, by:

$$BPI(D, \mathcal{Q}, \tau) \;:=\; \sum_{S \subseteq D^{en} \smallsetminus \{\tau\}} \frac{1}{2^{|D^{en}|-1}} \cdot \Delta(\mathcal{Q}, S, \tau), \quad (6.3)$$

is also popular in game theory. It is defined similarly to the Shapley value, but without considering, for subinstances S of $D^{en} \smallsetminus \{\tau\}$, the numbers of permutations of S and $D^{en} \smallsetminus (S \cup \{\tau\})$. For the BPI, only the number of subsinstances S matter.

In (Livshits et al., 2021b), it was shown that the CES coincides with the *Banzhaf Power-Index* (BPI). Accordingly:

$$CE^{UI}(D, \mathcal{Q}, \tau) \;=\; \sum_{S \subseteq D^{en} \smallsetminus \{\tau\}} \frac{1}{2^{|D^{en}|-1}} \cdot \Delta(\mathcal{Q}, S, \tau), \quad (6.4)$$

This fact will be important when comparing the CES to the Shapley Value. It is worth mentioning that the definition of the Shapley value guarantees that *it is the only measure* of contribution that satisfies certain desirable properties (Shapley, 1953; Roth, 1988). Although the definition of the BPI may look more intuitive than that of the Shapley value (considering all permutations of a subteam and its complement may not look immediately obvious), the BPI is bound to miss some of the properties of the Shapley value. Actually, that of *Efficiency*, as will see in more detail in Chapter 9.

The following proposition will be used in Chapter 7.

Proposition 3.[1] Let $\mathcal{Q}$ be a monotone Boolean query, D and D' two database instances such: (a) $D \cap D' = \emptyset$ and (b) the set of minimal satisfiable sets (MSS, see Def. 3) of their union, that is, $D \cup D'$, is:

$$MSS(D \cup D', \mathcal{Q}) = MSS(D, \mathcal{Q}) \cup MSS(D', \mathcal{Q}),$$

Then, for given endogenous tuples $\tau \in D$ and $\tau' \in D'$, it holds:

$$CE^{UI}(D \cup D', \mathcal{Q}, \tau) = CE^{UI}(D, \mathcal{Q}, \tau) \cdot (1 - P_{D^{p'}}(\mathcal{Q}))\text{ , and}$$
$$CE^{UI}(D \cup D', \mathcal{Q}, \tau') = CE^{UI}(D', \mathcal{Q}, \tau') \cdot (1 - P_{D^{p}}(\mathcal{Q}))$$

Where D^p and $D^{p'}$ denote the TID instances from D and D' with uniform distribution, and $P_{D^p}(\mathcal{Q})$ and $P_{D^{p'}}(\mathcal{Q})$ denotes the probability of the query $\mathcal{Q}$ of being true when evaluated over each.

Proof. Let $\mathcal{Q}$ be a Boolean monotone query and D and D' denote two (non-probabilistic) instances such $D \cap D' = \emptyset$ and $MSS(D \cup D', \mathcal{Q}) = MSS(D, \mathcal{Q}) \cup MSS(D', \mathcal{Q})$.

Now, we compute CES for an endogenous tuple $\tau \in D$ for the query $\mathcal{Q}$ in $D^\star$ according to Eq. 5.3:

$$\begin{aligned} CE^{UI}(D \cup D', \mathcal{Q}, \tau) &= \mathbb{E}(\mathcal{Q} \mid do(\tau\ \textsf{in})) - \mathbb{E}(\mathcal{Q} \mid do(\tau\ \textsf{out})) \\ &= P_{D \cup D'}(\mathcal{Q} = 1 \mid do(\tau\ \textsf{in})) - P_{D \cup D'}(\mathcal{Q} = 1 \mid do(\tau\ \textsf{out})) \end{aligned}$$

Here, $P_{D \cup D'}(Q = 1)$ denotes the probability of $\mathcal{Q}$ being true in the TID instance built from the the (non-probabilistic) instance $D \cup D'$ and the uniform distribution. Note that, since $\tau \in D$ and the instances D and D' satisfies conditions (a) and (b), then

$$\begin{aligned} P_{D \cup D'}(Q = 1 \mid do(\tau\ \textsf{in})) &= P_D(\mathcal{Q} = 1 \mid do(\tau\ \textsf{in})) \cdot P_{D'}(Q = 1 \mid do(\tau\ \textsf{in})) \\ &\quad + P_D(\mathcal{Q} = 0 \mid do(\tau\ \textsf{in})) \cdot P_{D'}(Q = 1 \mid do(\tau\ \textsf{in})) \\ &\quad + P_D(\mathcal{Q} = 1 \mid do(\tau\ \textsf{in})) \cdot P_{D'}(Q = 0 \mid do(\tau\ \textsf{in})) \\ &= P_D(\mathcal{Q} = 1 \mid do(\tau\ \textsf{in})) \cdot (1 - P_{D'}(Q = 1 \mid do(\tau\ \textsf{in}))) \\ &= P_D(\mathcal{Q} = 1 \mid do(\tau\ \textsf{in})) \cdot (1 - P_{D'}(Q = 1)) \end{aligned}$$

This also applies for $P_{D \cup D'}(Q = 1 \mid do(\tau\ \textsf{out}))$. Then, by factoring the terms, we obtain the desired result. This process is analogous for an endogenous tuple $\tau' \in D'$. □

Additionally, the following proposition relates $\Delta(\mathcal{Q}, S, \tau)$ in (6.2) with the the swinging worlds (see Def. 3); and swinging worlds with the contingencies of a tuple $\tau \in D^{en}$.

Proposition 4. Let D be a database instance with $\tau \in D^{en}$ and $\mathcal{Q}$ a Boolean monotone query.

(a) For $S \subseteq (D^{en} \smallsetminus \{\tau\})$: $\Delta(\mathcal{Q}, S, \tau) = 1$ iff $(S \cup \{\tau\} \cup D^{ex}) \in Swin(D, \mathcal{Q}, \tau)$.
(b) For every $D^{ex} \subseteq W \subseteq D$: $W \in Swin(D, \mathcal{Q}, \tau)$ iff $(D \smallsetminus W) \in Cont(D, \mathcal{Q}, \tau)$.

[1] This proposition can be easily generalized for a TID with an arbitrary probability distribution.

Proof. (a) Each swinging world $W \in Swin(D, \mathcal{Q}, \tau)$ satisfies $\mathcal{Q}[W] = 1$ and $\mathcal{Q}[W \setminus \{\tau\}] = 0$, and since $D^{ex} \subseteq W$, we can write $S = (W \setminus D^{ex}) \setminus \{\tau\}$, obtaining $\Delta(\mathcal{Q}, S, \tau) = 1$.

(b) ($\Rightarrow$) Assume that $W \in Swin(D, \mathcal{Q}, \tau)$. Then, it holds that $\mathcal{Q}[W] = 1$ and $\mathcal{Q}[W \setminus \{\tau\}] = 0$. Denote $\Gamma = D \setminus W$. Note that $W = D \setminus \Gamma$. Then, it holds that $\mathcal{Q}[D \setminus \Gamma] = 1$ and $Q[(D \setminus \Gamma) \setminus \{\tau\}] = 0$, which means that $\Gamma \in Cont(D, \mathcal{Q}, \tau)$.

($\Leftarrow$) Denote $\Gamma = D \setminus W$. Assume that $\Gamma \in Cont(D, \mathcal{Q}, \tau)$, which means that $Q[D \setminus \Gamma] = 1$ and $Q[(D \setminus \Gamma) \setminus \{\tau\}] = 0$. Note that $W = D \setminus \Gamma$. It follows that, $Q[W] = 1$ and $Q[W \setminus \{\tau\}] = 0$, which means that $W \in Swin(D, \mathcal{Q}, \tau)$. □

From this proposition, we immediately obtain: *For an instance D, a monotone Boolean query $\mathcal{Q}$, and $D^{ex} \subseteq W \subseteq D$, it holds:*

$$\begin{aligned} CE^{UI}(D, \mathcal{Q}, \tau) &= \frac{|Swin(D, \mathcal{Q}, \tau)|}{2^{|D^{en}|-1}} \\ Shapley(D, \mathcal{Q}, \tau) &= \sum_{W \in Swin(D,\mathcal{Q},\tau)} \frac{(|W| - |D^{ex}| - 1)! \cdot (|D| - |W|)!}{|D^{en}|!}. \end{aligned} \tag{6.5}$$

$$\begin{aligned} CE^{UI}(D, \mathcal{Q}, \tau) &= \frac{|Cont(D, \mathcal{Q}, \tau)|}{2^{|D^{en}|-1}} \\ Shapley(D, \mathcal{Q}, \tau) &= \sum_{\Gamma \in Cont(D,\mathcal{Q},\tau)} \frac{(|D^{en}| - |\Gamma| - 1)! \cdot |\Gamma|!}{|D^{en}|!}. \end{aligned} \tag{6.6}$$

■

A tuple $\tau \in D^{en} \subseteq D$ is called a *dummy tuple* for a BCQ $\mathcal{Q}$ if $\Delta(\mathcal{Q}, S, \tau) = 0$ for every $S \subseteq D^{en}$. We will often assume that no *dummy tuples* exist in the database at hand, since the following proposition holds.

Proposition 5. Let $\mathcal{Q}$ be a BCQ, and D an instance containing a dummy tuple τ_d for $\mathcal{Q}$. Then, for every tuple $\tau \in (D^{en} \setminus \{\tau_d\})$, it holds:

(a) $CE^{UI}(D, \mathcal{Q}, \tau) = CE^{UI}(D \setminus \{\tau_d\}, \mathcal{Q}, \tau)$.

(b) $\rho(D, \mathcal{Q}, \tau) = \rho(D \setminus \{\tau_d\}, \mathcal{Q}, \tau)$.

(c) $Shapley(D, \mathcal{Q}, \tau) = Shapley(D \setminus \{\tau_d\}, \mathcal{Q}, \tau)$.

Proof. For (a), recall that we can compute the CES for a given tuple according to Equation (6.4). Note that, for each $S \subseteq (D \setminus \{\tau_d\})$, $\Delta(\mathcal{Q}, S, \tau) = 1$ iff $\Delta(\mathcal{Q}, S \cup \tau_d, \tau) = 1$. Therefore, adding the tuple τ_d to $D \setminus \{\tau_d\}$, the number of $\Delta(\mathcal{Q}, S \cup \tau_d, \tau)$, which are equal to one, doubles. This change is countered by the increasing in endogenous tuples by one, leaving the CES unchanged for all endogenous tuples.
The proof for (b) is trivial, since, as the tuple τ_d does not change the value of the query, it does not belong to any contingency set of τ for the answer of $\mathcal{Q}$ and, therefore, the *Responsibility* is the same including or not said tuple.
For (c) we recall a property of Shapley Value in relation with dummy tuples: for a given BCQ $\mathcal{Q}$ and a instance D, if τ is a dummy tuple, then $Shapley(D, \mathcal{Q}, \tau) = 0$ (Shapley, 1953). It follows that, if we remove the tuples from the instance, the Shapley Value do not change. □

Tuples	CES / BPI	*Responsibility*	Shapley Value
τ_1	0.65625	1/3	0.5833
τ_2, τ_3	0.21875	1/3	0.1333
τ_4, τ_5, τ_6	0.09375	1/3	0.05

TABLE 6.1: CES, *Responsibility* and Shapley Value for each tuple in D.

Now, as discussed in Example 3, we can compare the (original, non-generalized version of the) CES with the last two scores we just introduced.

Example 6. (ex. 3 cont.) Consider the database instance D containing relation E in Table 5.1, with all tuples considered endogenous, and query $\mathcal{Q}_1$. The results of the computation of the CES, *Responsibility* and Shapley Value are shown in Table 6.1.

We can see, in this example, that the less informative score is *Responsibility*, which assigns 1/3 for all the tuples in D, despite the fact that the numbers of tuples of each path is different. Proposition 1 explains this result: (a) All paths are disjoint, and (b) all tuples belongs to one path. This makes *responsibility* the same for all tuples. If, for instance, one tuple belonged to two or more paths, then *Responsibility* would be different for some tuples.

Now, we can compute CES and the Shapley Value using the set of swiging worlds or the set of contingencies by using equations (6.5) and (6.6), respectively. For example, if we compute the swinging worlds and contingencies for τ_4, we obtain:

$$\begin{aligned} Swin(\mathcal{Q}_1, \tau_4) &= \{\{\tau_3, \tau_4, \tau_5, \tau_6\}, \{\tau_2, \tau_4, \tau_5, \tau_6\}, \{\tau_4, \tau_5, \tau_6\}\}, \text{ and} \\ Cont(\mathcal{Q}_1, \tau_4) &= \{\{\tau_1, \tau_2\}, \{\tau_1, \tau_3\}, \{\tau_1, \tau_2, \tau_3\}\}. \end{aligned}$$

Then, for CES and Shapley Value for τ_4 we obtain:

$$\begin{aligned} CE^{UI}(D, \mathcal{Q}_1, \tau_4) &= \frac{|Swin(\mathcal{Q}_1, \tau_4)|}{2^{|D^{en}|-1}} = \frac{|Cont(\mathcal{Q}_1, \tau_4)|}{2^{|D^{en}|-1}} = \frac{3}{32} = 0.09375 \\ Shapley(D, \mathcal{Q}_1, \tau_4) &= \sum_{W \in Swin(\mathcal{Q}_1, \tau_4)} \frac{(|W| - |D^{ex}| - 1)! \cdot (|D| - |W|)!}{|D^{en}|!} \\ &= \frac{(4-1)! \cdot (6-4)!}{6!} + \frac{(4-1)! \cdot (6-4)!}{6!} + \frac{(3-1)! \cdot (6-3)!}{6!} \\ &= 0.05 \end{aligned}$$

$$\begin{aligned} Shapley(D, \mathcal{Q}_1, \tau_4) &= \sum_{\Gamma \in Cont(\mathcal{Q}_1, \tau_4)} \frac{(|D^{en}| - |\Gamma| - 1)! \cdot |\Gamma|!}{|D^{en}|!} \\ &= \frac{(6-2-1)! \cdot (2)!}{6!} + \frac{(6-2-1)! \cdot (2)!}{6!} + \frac{(6-3-1)! \cdot (3)!}{6!} \\ &= 0.05. \end{aligned}$$

In this example, CES and the Shapley Value return different scores, but produce the same qualitative *rankings* for the tuples, i.e. they are equally ordered (according to their scores). We will see in Example 8, that this may not always be the case. ■

Chapter 7

Score Alignments for Non-Probabilistic Databases

Example 6 motivates deeper comparison of the attribution scores. In this section we address the problem of *alignment of scores*, but for the basic case of CES, CE^{UI}, which starts with a non-probabilistic DB. To compare the orders of the tuples induced by the scores, we first make precise the notion of *alignment*. Intuitively, the induced orders have to be mutually compatible.

Definition 4. (rankings and score alignment) Let $\mathcal{Q}$ be a boolean query, D a relational instance and $\tau, \tau' \in D^{en}$ two endogenous tuples.

(a) Given a numerical attribution score $\mathsf{sc}(D, \mathcal{Q}, \cdot)$, as a function of $\tau \in D^{en}$, its *ranking* is the total preorder relation on D^{en} defined by: $\tau \preceq^{\mathsf{sc}} \tau'$ iff $\mathsf{sc}(D, \mathcal{Q}, \tau) \leq \mathsf{sc}(D, \mathcal{Q}, \tau')$.

(b) Two numerical attribution scores $\mathsf{sc}_1, \mathsf{sc}_2$ are *aligned* if, for every $\tau, \tau' \in D^{en}$, $\tau \preceq^{\mathsf{sc}_1} \tau'$ iff $\tau \preceq^{\mathsf{sc}_2} \tau'$. ■

Notice that, according to Definition 4, if a score assigns the same value to all tuples in the database, it will be *aligned* with any other score. In Example 6, *Responsibility* is *aligned* with the CES and the Shapley Value. However, for monotone queries, these three scores may not be pairwise aligned. That is, for every pair of scores, there is a pair $(D, \mathcal{Q})$ for which the rankings are incompatible, as the counterexamples below show.

Example 7. (CES vs *Responsibility*) Consider the database $D^\star$ with the relations in Table 7.1 (a), and the BCQ

$$\mathcal{Q}_{\mathsf{RST}}: \ \exists x \exists y (R(x) \wedge S(x,y) \wedge T(y)). \tag{7.1}$$

Consider that all tuples in $D^\star$ are endogenous. Now, the CES, Responsibility and Shapley Value is computed for τ_3, τ_4 and τ_{11}. The results are shown in Table 7.1 (b).

From this, and denoting with $\preceq^\rho$ the order associated to *Responsibility*, the induced orders are: $\tau_4 \preceq^{CE^{UI}} \tau_3 \preceq^{CE^{UI}} \tau_{11}$ and $\tau_4 \preceq^\rho \tau_{11} \preceq^\rho \tau_3$. It follows that *Responsibility* is not *aligned* with the CES. ■

R	A
τ_1	a
τ_2	b
τ_3	e

S	A	B
τ_4	a	b
τ_5	a	c
τ_6	a	d
τ_7	b	b
τ_8	b	c
τ_9	b	d
τ_{10}	e	f

T	A
τ_{11}	b
τ_{12}	c
τ_{13}	d
τ_{14}	f

τ	*CES*	*Resp.*
τ_3	0.1292	1/3
τ_4	0.0829	1/5
τ_{11}	0.1868	1/4

TABLE 7.1: (a) Database $D^\star$ and (b) CES and *Responsibility* Value for $(D^\star, \mathcal{Q}_{\mathsf{RST}})$.

R	A	B
τ_1	a	c_1
τ_2	b	c_2
τ_3	b	c_3

S	A	B
τ_4	a	c_4
τ_5	a	c_5
τ_6	b	c_6
τ_7	b	c_7
τ_8	b	c_8
τ_9	b	c_9

τ	*CES*	*Resp.*	*Shapley*
τ_1	57/256	1/3	400/2520
τ_4	19/256	1/4	151/2520
τ_6	15/256	1/5	169/2520

TABLE 7.2: (a) Database $D^\star$ and (b) CES and Shapley Value for $(D^\star, \mathcal{Q}_{\mathsf{RS}})$.

Example 8. (CES and *Responsibility* vs Shapley Value) Consider the database $D^\star$ given by the relations in Table 7.2 (a), and the BCQ

$$\mathcal{Q}_{\mathsf{RS}}: \exists x \exists y (R(x,y) \land S(x,z)) \tag{7.2}$$

CES, *Responsibility* and Shapley are computed for τ_1,τ_4 and τ_6, as shown in Table 7.2 (b). It holds: $\tau_6 \preceq^{CE^{UI}} \tau_4 \preceq^{CE^{UI}} \tau_1$, $\tau_6 \preceq^{\rho} \tau_4 \preceq^{\rho} \tau_1$ and $\tau_4 \preceq^{Shapley} \tau_6 \preceq^{Shapley} \tau_1$. Then, the Shapley is not *aligned* with neither CES nor *Responsibility*. ■

It is worth mentioning that the database instances provided for each counter example are not the only databases where the pair of scores are not aligned for $\mathcal{Q}_{\mathsf{RST}}$.

In the following sections, we specify some syntactic classes of queries for which the scores are always aligned, and also, other classes for which the scores are not always aligned. We first compare CES with *Responsibility* in two scenarios: (a) instances with exogenous tuples, and (b) instances that do not have them. For case (a), we fully characterize the class of BCQs, that is, for every BCQ, we can determine if the scores are always aligned or not according to the query's syntax. For case (b), we exhibit two classes of queries for which the scores are always aligned. Finally, we compare both CES and *Responsibility* with the Shapley Value, exhibiting a class of queries for which the scores are not always aligned.

7.1 CES vs *Responsibility* Alignment

In the present section we treat the problem of alignment for CES and *Responsibility* in its more general way: the scores will be aligned for a query $\mathcal{Q}$ if, for all possible database instances D for $\mathcal{Q}$, the scores are aligned for $(D, \mathcal{Q})$. We will refer to "all possible database instances" for a given query $\mathcal{Q}$ as the collection of tuples built from the atoms in $\mathcal{Q}$, where each tuple can or not be exogenous. Actually, exogenous tuples are needed for most of the counter-examples that follow.

We start this section by providing some important notation, definitions and preliminary propositions about the syntactic structure of the query at hand. Next, we state the main result of this section, namely Theorem 1, which is proved by breaking it down in different cases.

Definition 5. Let $\mathcal{Q}$ be a BCQ. (a) Consider the undirected graph G whose nodes are the atoms of $\mathcal{Q}$, and edges are the pairs of atoms, (A_i, A_j), that have at least one variable in common. (a) The *components* of $\mathcal{Q}$ are the connected components of graph G. We will denote the components of $\mathcal{Q}$ by $C_1, \ldots, C_n$. $\mathcal{Q}_i$ will denote the subquery of $\mathcal{Q}$ formed by the (existentially quantified) conjunction of the atoms in C_i. (b) A set of variables $V \subseteq \mathit{Var}(\mathcal{Q})$ is *coincident* if all the sets $\mathit{Atoms}(v)$, for $v \in V$, are the same. The set of all coincident sets of variables is denoted by $\mathit{Coin}(\mathcal{Q})$. ■

Notice that: (a) $\mathit{Coin}(\mathcal{Q})$ is a partition of $\mathit{Var}(\mathcal{Q})$; and (b) If $\mathcal{Q}$ contains only trivial coincident sets, i.e. singletons, then $|\mathit{Coin}(\mathcal{Q})| = |\mathit{Var}(\mathcal{Q})|$ (see Example 9).

Example 9. For the query $\mathcal{Q}: \bar{\exists}(R(x,y) \wedge S(x) \wedge T(z,w) \wedge U(z))$, the undirected graph G has as nodes the elements of $\mathit{Atoms}(\mathcal{Q})$, and the edges are the pairs $(R(x,y), S(x))$ and $(T(z,w), U(w))$, which define also the two components: $C_1 = \{R(x,y), S(x)\}$ and $C_2 = \{T(z,w), U(w)\}$. The associated subqueries are $\mathcal{Q}_1: \exists x \exists y (R(x,y) \wedge S(x))$ and $\mathcal{Q}_2: \exists z \exists w (T(z,w) \wedge U(z))$.

$\mathit{Coin}(\mathcal{Q}) = \{\{x\}, \{y\}, \{z\}, \{w\}\}$ contains only trivial coincident sets if variables, they are singletons. However for the query $\mathcal{Q}: \bar{\exists}(R(x,y) \wedge T(x,y,z) \wedge U(z))$, $\mathit{Coin}(\mathcal{Q}) = \{\{x,y\}, \{z\}\}$, that contains a non-trivial coincident set of variables. ■

Notice that any two components of a query do not share any variable. Components can be identified as "independent" sub-queries.

Intuitively, the variables in a coincident set can be treated as if they were only one. More technically, this is done by mapping each unique combination of constants that occurs in the database in the positions of the coincident variables to a new fresh constant in a target database. The set of exogenous tuples is preserved, that is, a tuple in the target database is exogenous iff its corresponding tuple in the original database is exogenous. All this results in the same result for Boolean query evaluation. The following result formalizes this claim. For simplicity, only for two variables in a coincident set, but the result can be generalized iteratively to all the variables in a coincident set.

Proposition 6. (Coincident Variables) Let $\mathcal{Q}$ be a BCQ for a schema $\mathcal{S}$. Let $x, y \in \mathit{Var}(\mathcal{Q})$ be such that $\mathit{Atoms}(x) = \mathit{Atoms}(y)$. Consider the BCQ query $\mathcal{Q}'$ for a new schema $\mathcal{S}'$ obtained by decreasing the arity of the relational predicates of atoms in $\mathit{Atoms}(x)$ by 1, and replacing the occurrences of x, y by a single occurrence of a fresh variable v. It holds that, for every instance D for $\mathcal{S}$, one can build an instance D' for $\mathcal{S}'$ via a transformation f of tuples in D, such that, for each $S \subseteq D$: $\mathcal{Q}[S] = \mathcal{Q}'[f(S)]$. ■

Before proving the proposition, to convey the idea, we give an example.

Example 10. Consider the BCQ $\mathcal{Q}$: $\bar{\exists}(R(x,y) \wedge T(y,z,x) \wedge U(z))$, and the instance D in Table 7.3, with $D^{ex} = \{\tau_1, \tau_2\}$. Tuple τ_{10} is a *dummy tuple* (see Proposition 5).

R	A	B
$\underline{\tau_1}$	a	b
$\underline{\tau_2}$	a	c
τ_3	a	d
τ_4	b	b

S	A	B	C
τ_5	b	a	a
τ_6	b	b	a
τ_7	c	a	a
τ_8	d	a	a
τ_9	b	a	b
τ_{10}	c	a	c

T	A
τ_{11}	a
τ_{12}	b

TABLE 7.3: Instance D.

R'	A'
$\underline{\tau'_1}$	c_1
$\underline{\tau'_2}$	c_2
τ'_3	c_3
τ'_4	c_4

S'	A'	B'
τ'_5	c_1	a
τ'_6	c_1	b
τ'_7	c_2	a
τ'_8	c_3	a
τ'_9	c_4	a
τ'_{10}	c_5	a

T	A'
τ'_{11}	a
τ'_{12}	b

TABLE 7.4: Instance D'.

The new query is $\mathcal{Q}'\colon\ \bar{\exists}(R(v) \wedge T(v,z) \wedge U(z))$. Notice that it does not have non-trivial sets of coincident variables. Instance D' has to be build from D, and be compliant with the new schema. It is the instance in Table 7.4, which is obtained via the mapping $f\colon D \to D'$ defined by $f(\tau_i) := \tau'_i$. Moreover, the set of exogenous tuples of D' is $D'^{ex} = \{\tau'_1, \tau'_2\}$.

The transformation introduces fresh constants, the c_i (in the domain of the new schema). It holds: $\mathcal{Q}[S] = \mathcal{Q}'[S']$, for every $S \subseteq D$ and $S' = \{f(\tau) : \tau \in S\}$. ■

Remark 5. *As already pointed out, we can iteratively decrease any coincident set until its size is one, which motivates the following terminology and notation: Let $\mathcal{Q}$ be a BCQ with non-trivial set of coincident variables (i.e. its elements are not all singletons), and D a database instance. (a) The* reduced form *of $\mathcal{Q}$, denoted by $\mathcal{Q}^{red}$ is the result of iteratively decreasing the number of coincident variables, until making each $V \in Coin(\mathcal{Q})$ collapse into a singleton. $\mathcal{Q}^{red}$ does not have non-trivial sets of coincident variables. As a consequence, $Coin(\mathcal{Q}^{red}) = Var(\mathcal{Q}^{red})$. In Example 10, $\mathcal{Q}'$ is the* reduced form *of $\mathcal{Q}$. (b) Similarly, D^{red} denotes the* reduced form *of D, obtained at the end of the iterative application (composition), denoted f^{red}, of the one-step reduction transformation f.* ■

Notice that for Proposition 6 the self-join-free condition is needed. We will use Proposition 6 in Lemma 1.

Proof. (of Proposition 6) Let $\mathcal{Q}$, $\mathcal{Q}'$, D and D' as in Proposition 6. We refer to them as *original* and *target*, query or database, resp. Denote x, y the coincident variables and $v \notin Var(\mathcal{Q})$ the new variable.

Now D' is built, according to $\mathcal{Q}'$, in the following way: (1) First, add all tuples from D that are not from the relations of the atoms in $Atoms(x)$. (2) Then, for each atom $U \in Atoms(\mathcal{Q})$, introduce a fresh atom U', whose relational predicate's arity is that of the relational predicate of U minus 1, and it has the same variables as the atom U, but x and y, which are replaced by a single variable v; (3) Lastly, for each tuple τ from the relations of the atoms in $Atoms(\mathcal{Q})$, create the tuple τ' according to the following:

(a) Start from the new atom U' and replace each variable in it, but v, with the constant found in its position in the atom U from the tuple τ.

(b) Replace the variable v in U' with a unique fresh constant for each unique combination of constants of τ in the x and y's position in the atom U.

We exemplify step (2) and (3). Consider the atom $U(x,y,z,w)$ with four variables, being x and y two coincident variables, and consider the ground atom $U(a,b,c,d)$, being a, b, c and d constants. We first create the new atom $U'(v,z,w)$, being v a fresh variable. Note that the variables z and w are in both atoms, U and U', and x and y were replaced by v. After this, a new tuple is created, starting from the ground atom $U(a,b,c,d)$ by doing the following: (a) z and w are replaced by c and d (the constants in the positions of each variable in the atom $U(x,y,z,w)$, of the ground atom $U(a,b,c,d)$), and (b) v is replaced by a fresh unique constant c'. This result in a new ground atom $U'(c',c,d)$.

Now, consider the transformation $f : D \to D'$: For a tuple $\tau \in D$,

$$f(\tau) := \begin{cases} \tau' & \text{, if } \tau \text{ as a ground atom has the same predicate as some} U \in Atoms(x), \\ \tau & \text{, otherwise.} \end{cases}$$

Here, τ' is newly created tuple starting from τ.

By applying this transformation, any homomorphism that maps the atom U to the tuple τ has its corresponding unique homomorphism h' that maps from the new atom U' to the corresponding tuple $f(\tau)$.

We need to prove that $\mathcal{Q}[S] = \mathcal{Q}'[S']$ for every $S \subseteq D$ and $S' = \{f(\tau) : \tau \in S\}$. We do this by showing that every tuple τ is mapped to a unique tuple in τ', which is equivalent to showing that f is bijective.

Consider two tuples $\tau, \tau' \in D$. If (a,b) and (a',b') are pairs of constants in the positions of x and y for τ and τ' respectively, $f(\tau) = f(\tau')$ only if $\tau = \tau'$, which means that f is injective. This occurs even when τ is not from a relation appearing in $Atoms(x)$, because $f(\tau) = \tau$ in this case. Furthermore, by the way that D' is created, every tuple in it will have its correspondent tuple in D, implying that f is surjective. Then, f is a bijection.

Since $\mathcal{Q}[S] = \mathcal{Q}'[S']$ for every $S \subseteq D$, the bijection also holds with a non-empty set of exogenous tuples, since each subset of the original (and of the target) database now needs to contain D^{ex} (D'^{ex}, resp.). □

Now, we state the main result of this section.

Theorem 1. Let $\mathcal{Q}$ be a BCQ and $C_1, \ldots, C_n$, its components. Let $\mathcal{Q}_i$ denote the query built from the conjunction of all atoms in the component C_i. It holds:
(a) If $n = 1$ and $|Coin(\mathcal{Q})| = 1$, or if $n \geq 2$ and $|Atoms(\mathcal{Q}_i)| = 1$ for all $i = 1, \ldots, n$, then the CES and the *Responsibility* are aligned for $(D, \mathcal{Q})$ for every instance D with or without exogenous tuples.
(b) Otherwise, there is an instance D with a non-empty set of exogenous tuples, such that CES and *Responsibility* are not aligned for $(D, \mathcal{Q})$. ■

Before proving this result, we illustrate it with an example.

R_1	A	B
τ_1	a	a
τ_2	b	b

R_2	A	B	C
τ_3	a	a	a
τ_4	a	b	a
τ_5	b	a	b

R_2	A	B
τ_6	a	a
τ_7	b	b

τ	*CES*	*Resp.*
τ_3	0.09375	1/3
τ_5	0.03125	1/2

TABLE 7.5: (a) Database D. (b) CES and *Responsibility* for τ_3 and τ_5.

Example 11. Consider the BCQs:

$$\mathcal{Q} : \bar{\exists}(R_1(x,z) \wedge R_2(x,y,z) \wedge R_3(x,z)),$$
$$\mathcal{Q}' : \bar{\exists}(R_1(x,y) \wedge R_2(z) \wedge R_3(w,v)).$$

Query $\mathcal{Q}$ has one component, and query $\mathcal{Q}'$ has three components: $C'_1 = \{R_1(x,y)\}$, $C'_2 = \{R_2(z)\}$ and $C'_3 = \{R_3(w,v)\}$.

Since $\mathcal{Q}$ has only one component, the size of the set of coincident variables is $|Coin(\mathcal{Q})| = |\{\{x,z\},\{y\}\}| = 2$. By Theorem 1, there is a database D for which CES and *Responsibility* are not aligned for $(D,\mathcal{Q})$. Actually, in Proposition 8 we provide an algorithm to built such a database. For now, Table 7.5(a) shows the relations from D. Table 7.5(b) shows the CES and *Responsibility* scores for tuples τ_3 and τ_5: $\tau_5 \preceq^{CE^{UI}} \tau_3$ and $\tau_3 \preceq^{\rho} \tau_5$. The scores are not aligned for $(D,\mathcal{Q})$.

Since for $\mathcal{Q}'$ each component has only one atom, for every instance D, with or without exogenous tuples, CES and *Responsibility* are always aligned for $(D,\mathcal{Q})$. ■

Remark 6. *In order to prove Theorem 1, we proceed as follows:*

1. *We start by restricting queries to those with a single component. For instance, query* $\mathcal{Q}\!:\!\bar{\exists}(R(x,y) \wedge S(x,z))$ *satisfies this condition, but the query* $\mathcal{Q}'\!:\!\bar{\exists}(R(x,y) \wedge S(z))$ *not, since it has two components.*

   For this class of queries, Propositions 7 and Proposition 8 establish Theorem 1(a) and (b), resp., but assuming an additional condition of the queries.

   After that, with Lemma 1, that can be used to address the extra condition, we manage to establish Theorem 1 in its full generality (for single component queries).

2. *Next, we investigate score alignment for multi-component queries. For this case, Proposition 9 establishes Theorem 1(a); and Proposition 10, Theorem 1(b).* ■

Proposition 7. Consider the query $\mathcal{Q}_{R_n}\!:\!\exists x(R_1(x) \wedge R_2(x) \wedge \cdots \wedge R_n(x))$. It does not have a non-trivial coincident set of variables, and $n \geq 1$. CES and *Responsibility* are aligned for $(D,\mathcal{Q}_{R_n})$, for every instance D, with or without exogenous tuples.

Proof. Consider a database instance D without *dummy tuples*. Denote r_i the number of tuples of the relation R_i. The CES and *Responsiblity* for a tuple τ_i in relation R_i are:

$$CE^{UI}(D,\mathcal{Q},\tau_i) \;=\; \frac{\prod_{r_j=1,\ldots,n} (2^{r_j}-1)}{2^{|D|-1}} \times \frac{1}{(2^{r_i}-1)}, \tag{7.3}$$

and $\rho(D,\mathcal{Q},\tau) = \frac{1}{r_i}$. Notice that the first factor on the RHS of (7.3) does not depend on the tuple at hand. It follows that, for any two tuples τ_i and τ_j from relations R_i

R	A	B
τ_1	a	a
τ_2	a	b
τ_3	b	a

S	A
τ_4	a
τ_5	b

τ	*CES*	*Resp.*
τ_1	0.375	1/3
τ_3	0.125	1/2

TABLE 7.6: (a) Database D_2. (b) CES and *Responsibility* for τ_1 and τ_3.

and R_j resp., if $r_i < r_j$, it holds:

$$CE^{UI}(D, \mathcal{Q}, \tau_i) > CE^{UI}(D, \mathcal{Q}, \tau_j) \quad \text{and} \quad \rho(D, \mathcal{Q}, \tau_i) > \rho(D, \mathcal{Q}, \tau_j).$$

Whereas, if $r_i = r_j$, CES and *Responsibility* for each tuples are the same.

Now, consider a non-empty set of exogenous tuples. Notice that if there is an exogenous tuple in relation R_i, all endogenous tuples from that relation will have CES and *Responsibility* equal to 0. This occurs because there is no possible subset of tuples such that, removing an endogenous tuples from said relation, will produces a change in the query's answer. It follows that the scores will be also aligned for this case. □

Remark 7. *In order to establish negative, i.e. non-alignment, results, for a given class, $\mathcal{C}$, of queries, we will use on several occasions a technique based on the following ideas:*[1] *(a) Start with a concrete example consisting in a particular query $\mathcal{Q}$ and a particular instance D_0, for which the negative result holds. The particular query belongs to $\mathcal{C}$, and is, in some sense, "contained" in the other queries in the class. (b) Given an arbitrary query $\mathcal{Q}'$ in $\mathcal{C}$, query $\mathcal{Q}$ is reconstructed as a part of $\mathcal{Q}'$. (c) In order to obtain an instance D for $\mathcal{Q}'$, we start from D_0, making D an extension of D_0 by adding exogenous tuples to the latter. In this way, every tuple in D_0 has a corresponding tuple in D. The new tuples in D will be endogenous only if the original tuples in D_0 are endogenous. Furthermore, the tuples in D that correspond to tuples in D_0, have the same CES and* Responsibility *scores as in D_0.* ■

We now proceed to establish and prove Proposition 8 by applying the general technique just described, starting with a (counter-)example to be used in its proof.

Example 12. Consider the query $\mathcal{Q}_2 \colon \exists x \exists y (R(x, y) \wedge S(x))$, and D_2 the instance in Table 7.6, and with $D^{ex} = \{\tau_4\}$ the set of exogenous tuples. It holds: $\tau_3 \preceq^{CE^{UI}} \tau_1$ and $\tau_1 \preceq^{\rho} \tau_3$. So, CES and *Responsibility* are not aligned for $(D_2, \mathcal{Q}_2)$. ■

Proposition 8. Let $\mathcal{Q}$ be a BCQ with a single component, without non-trivial coincident sets of variables, and with $Var(\mathcal{Q}) \geq 2$. There is an instance D with exogenous tuples, such the CES and *Responsibility* are not aligned for $(D, \mathcal{Q})$.

Proof. (of Proposition 8) Let $x, y \in Var(\mathcal{Q})$ such $Atoms(y) \subsetneq Atoms(x)$. Now, select two atoms R_x, R_y from $Atoms(\mathcal{Q})$ such $R_x \in (Atoms(x) \setminus Atoms(y))$ and $R_y \in Atoms(y)$. We build a database D from D_2 (as in Example 12) as follows:

(a) For each atom $U_R \in Atoms(y)$ and for each tuple τ_R from the relation R of D_2, we create a tuple from U_R by putting in the x and y's position the value of x and y in the

[1]A similar technique was used in (Livshits et al., 2021b; Dalvi & Suciu, 2007).

R_1	A	B	C
$\underline{\tau_1}$	c'	a	a
$\underline{\tau_2}$	c'	a	b
$\underline{\tau_3}$	c'	b	a

R_2	A	B
$\underline{\tau_4}$	a	c'
τ_5	b	c'

R_3	A	B	C
τ_6	a	a	c'
τ_7	a	b	c'
τ_8	b	a	c'

R_4	A
$\underline{\tau_9}$	c'

τ	*CES*	*Resp.*
τ_6	0.375	1/3
τ_8	0.125	1/2

TABLE 7.7: (a) Instance D built from D_2 (Ex. 12). (b) CES and *Responsibility* for τ_6 and τ_8.

tuple τ_S, and replacing by a constant c' the rest of variables. Only the tuples created from R_y are endogenous.

(b) For each atom $U_S \in (Atoms(x) \smallsetminus Atoms(y))$ and for each tuple τ_S from the relation S of D_2, we create a tuple from U_S by putting in the x and y's position the value of x's position the value of x in the tuple τ_R, and replacing by a constant c' the rest of variables. Only the tuples created from R_x are endogenous, with exception of the tuple with a constant a in the position of x, which will be exogenous.

(c) For each atom $U \notin Atoms(x)$ a tuple is created by replacing all variables in U with a constant c'. All tuples created in this way will be exogenous.

It follows that the CES and *Responsibility* are not aligned for $(D, \mathcal{Q})$. □

We now illustrate the construction of the database D of Proposition 8.

Example 13. Consider the BCQ $\mathcal{Q}\colon \bar{\exists}(R_1(x,y,z) \wedge R_2(y,w) \wedge R_3(y,z,w) \wedge R_4(w))$.

We select variables y and z, for which $Atoms(z) \subsetneqq Atoms(y)$, and the atoms R_2 and R_3. The variables and atoms selected are similar to the variables x, y and the atoms R_x and R_y used in the proof of Proposition 8, resp. Next, we build an instance D as in that proof, which results in the relations in Table 7.7.

The set of exogenous tuples in D is $D^{ex} = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_9\}$. CES and *Responsibility* are not aligned for $(D, \mathcal{Q})$: $\tau_8 \preceq^{CE^{UI}} \tau_6$ and $\tau_6 \preceq^{CE^{UI}} \tau_8$. ■

The results so far are for queries with a single component and without non-trivial coincident sets of variables. The extension to queries with a single component, but possibly with non-trivial coincident sets of variables follows from the following technical lemma.

Lemma 1. Let $\mathcal{Q}$ be a BCQ with a single component, and a non-trivial coincident set of variables. For every instance D, CES and *Responsibility* are aligned for the pair $(\mathcal{Q}, D)$ iff they are aligned for $(\mathcal{Q}^{red}, D^{red})$, where $\mathcal{Q}^{red}$ and D^{red} are the *reduced forms* of $\mathcal{Q}$ and D, respectively, as introduced in Remark 5.

Proof. Recalling Proposition 6, let f^{red} be the transformation function that outputs D^{red}. Since $\mathcal{Q}[S] = \mathcal{Q}^{red}[S^{red}]$ for every $S \subseteq D$, and $S^{red} = \{f^{red}(\tau) : \tau \in S\}$, it follows, for every $\tau \in D^{en}$, that $CE^{UI}(D, \mathcal{Q}, \tau) = CE^{UI}(D^{red}, \mathcal{Q}^{red}, f_R(\tau))$ and $\rho(D, \mathcal{Q}, \tau) = \rho(D^{red}, \mathcal{Q}^{red}, f^{red}(\tau))$. Then, CES and *Responsibility* are aligned for $(D, \mathcal{Q})$ iff they are aligned for $(D^{red}, \mathcal{Q}^{red})$. □

We now illustrate the use of Lemma 1 to obtain alignment and non-alignment results.

Example 14. Consider the following BCQs $\mathcal{Q}$ and $\mathcal{Q}'$ with their *reduced versions*, $\mathcal{Q}^{red}$ and $\mathcal{Q}'^{red}$, respectively (see Remark 5):

$$\mathcal{Q}\colon \bar{\exists}(R(x,y,z) \wedge S(x,y,z) \wedge T(x,y,z)), \qquad \mathcal{Q}^{red}\colon \bar{\exists}(R(v) \wedge S(v) \wedge T(v)),$$
$$\mathcal{Q}'\colon \bar{\exists}(R(x,y,w) \wedge S(x,y) \wedge T(x,y)), \qquad \mathcal{Q}'^{red}\colon \bar{\exists}(R(v',w) \wedge S(v') \wedge T(v')).$$

By Proposition 7, CES and *Responsibility* are aligned for $(D^{red}, \mathcal{Q}^{red})$, for every instance D^{red}. From Lemma 1 it follows that the scores are aligned for $(D, \mathcal{Q})$, for every instance D.

Regarding $\mathcal{Q}'^{red}$, by Proposition 8, there is an instance D'^{red}, such that CES and *Responsibility* are not aligned for $(D'^{red}, \mathcal{Q}'^{red})$. Again, by Lemma 1, there is an instance D', such that CES and *Responsibility* are not aligned for $(D', \mathcal{Q}')$. ■

According to Remark 6, we now address the case where the query has more than one component.

Proposition 9. Let $\mathcal{Q}$ be a BCQ with components $C_1, \ldots, C_n$, with $n \geq 2$. Let $\mathcal{Q}_i$ be the the subquery built as the conjunction of the atoms in C_i. If $|Atoms(\mathcal{Q}_i)| = 1$ for $i = 1, \ldots, n$, then CES and *Responsibility* are aligned for $(D, \mathcal{Q})$, for every instance D, with oe without exogenous tuples.

Proof. Let $\mathcal{Q}$ be a BCQ with two components, C_R and C_S, each with one atom, $R \in C_R$ and $S \in C_S$. We will prove first that the CES and *Responsibility* are always aligned for $\mathcal{Q}$ and we will later generalize this fact for queries with more than one component.

First, consider that, for any τ_R from relation R, its *Responsibility*, if there are no exogenous tuples in this relation, will be $\rho(\tau_R) = \frac{1}{r}$, where r corresponds the number of tuples from R, and $\rho = 0$ if there exists some exogenous tuple in said relation. Analogously, a tuple τ_S from relation S will have a *Responsibility* of $\rho(\tau_S) = \frac{1}{s}$, being s is the number of tuples from relation S, if no exogenous tuples exists from this relation, and $\rho(\tau_S) = 0$ if there exists some.

Similarly, the CES for τ_R and τ_S will be $CE^{UI}(\tau_R) = 1 - \frac{1}{2^s}$ and $CE^{UI}(\tau_S) = 1 - \frac{1}{2^r}$ if no exogenous tuples exists from R and S, respectively, and $CE^{UI}(\tau_R) = 0$ and $CE^{UI}(\tau_S) = 0$ if there exists some exogenous tuples in R or S, respectively.

It is clear that the CES and *Responsibility* will be aligned for $(D, \mathcal{Q})$, where D is any possible database, with or without exogenous tuples.

Now, consider a query $\mathcal{Q}'$ with one or more components that have only one atom each, and consider the query $\mathcal{Q}^\star$ as the conjunction of $\mathcal{Q}$ and $\mathcal{Q}'$. The *Responsibility* of τ_R and τ_S for $\mathcal{Q}^\star$ and some database $D^\star$ will be the same as only the query $\mathcal{Q}$ and the original database D, and thus, they are not altered. Regarding the CES, we have that

$$CE^{UI}(D^\star, \mathcal{Q}^\star, \tau_R) = CE^{UI}(D, \mathcal{Q}, \tau_R) \times P_{D^\star}(\mathcal{Q}')$$
$$CE^{UI}(D^\star, \mathcal{Q}^\star, \tau_S) = CE^{UI}(D, \mathcal{Q}, \tau_S) \times P_{D^\star}(\mathcal{Q}'),$$

and therefore, the orders are not altered either. Now, note that the initial components selected are arbitrary and therefore, we can perform the same pairwise comparison with any other pair of components (or atoms) in the query $\mathcal{Q}^\star$. By this, it follows that the CES and *Responsibility* are aligned for $(D^\star, \mathcal{Q}^\star)$, where $D^\star$ is any possible database with or without exogenous tuples. □

R	A
τ_1	a
τ_2	b
τ_3	c

S	A
τ_4	a
τ_5	b

T	A
τ_6	a
τ_7	b

τ	*CES*	*Resp.*
τ_1	0.28125	1/3
τ_4	0.125	1/2

TABLE 7.8: (a) Database $D^\star$. (b) CES and *Responsibility* for τ_1 and τ_3.

The next proposition, Proposition 10, will be proved by the technique described in Remark 7. We start by providing the (counter)-example that will be later use in the proof.

Example 15. Consider $\mathcal{Q}\colon \exists x \exists y (R(x) \wedge S(y) \wedge T(y))$, and the instance $D^\star$ in Table 7.8(a). CES and *Responsibility* of τ_1 and τ_4 are shown in Table 7.8(b). It holds: $\tau_4 \preceq^{CE^{UI}} \tau_1$ and $\tau_1 \preceq^{\rho} \tau_4$. So, CES and *Responsibility* are not aligned for $(D^\star, \mathcal{Q})$. ■

Proposition 10. Let $\mathcal{Q}$ be BCQ with two or more components, and at least one of them with at least two atoms. Then, there is an instance D, such the CES and *Responsibility* are not aligned.

Proof. First, two components, namely C_1 and C_2, are identify from the query $\mathcal{Q}$ such: $|C_1| \geq 1$ (trivial condition) and $|C_2| \geq 2$. Note that we can always select the component C_2 from $\mathcal{Q}$ since there exists at least one of component that has at least 2 atoms.

Now, from C_1, we select one atom that will be denoted by A_R. It will be assumed that $A_R \in Atoms(x)$. Similarly, from C_2 we select two atoms, A_S and A_T, and, for a variable y, $A_S, A_T \in Atoms(y)$. We build a database D from $D^\star$ (from Example 15) as follows:

(a) For each atom $U_x \in Atoms(x)$ and for each tuple τ_R from the relation R of $D^\star$, we create a tuple from U_R by putting in the x's position the value of x in the tuple τ_R, and replacing by a constant c' the rest of variables. Only the tuples created from A_R are endogenous.

(b) For each atom $U_S, U_T \in Atoms(y)$ and for each tuple τ_S and τ_T from the relations S and T of $D^\star$, we create a tuple from U_S and U_T by putting in the y's position the value of y in the tuple τ_S and τ_T, respectively, and replacing by a constant c' the rest of variables. Only the tuples created from A_S and A_T are endogenous, the rest will be exogenous.

(c) For each atom $U \notin (Atoms(x) \cup Atoms(y))$ a tuple is created by replacing all variables in U with a constant c'. All tuples created in this way will be exogenous.

By the construction of D we have that each endogenous tuple in it has its corresponding tuple in $D^\star$, and the CES and *Responsibility* of each are the same. It follows that both scores are not aligned for $(D, \mathcal{Q})$. □

We now illustrate the use of Proposition 10.

Example 16. Consider the BCQ $\mathcal{Q}\colon \bar{\exists}(R_1(x,y) \wedge R_2(z) \wedge R_3(z) \wedge R_4(z) \wedge R_5(w,v))$. It has three components: $C_1 = \{R_1(x,y)\}$, $C_2 = \{R_2(z), R_3(z), R_4(z)\}$, and $C_3 = \{R_5(w,v)\}$. Since C_2 has three components, by Proposition 10, there is an instance

R_1	A	B
$\underline{\tau_1}$	c'	c'

R_2	A
τ_2	a
τ_3	b

R_3	A
τ_4	a
τ_5	b

R_4	A
$\underline{\tau_6}$	a
$\underline{\tau_7}$	b

R_5	A	B
τ_8	a	c'
τ_9	b	c'
τ_{10}	c	c'

τ	*CES*	*Resp.*
τ_2	0.125	1/2
τ_8	0.28125	1/3

TABLE 7.9: (a) Database D. (b) CES and *Responsibility* for τ_2 and τ_8.

D, such the scores are not aligned. This instance is built as in the proof of that Proposition, which -in this example- leads us to select components C_2 and C_3. Next, from C_2, we select atoms $R_2(z)$ and $R_3(z)$; and atom $R_5(w, v)$ from C_3. Only the tuples from relations R_2, R_3, R_5 become endogenous; and the rest, exogenous.

Table 7.9(a) shows instance D, with $D^{ex} = \{\tau_1, \tau_6, \tau_7\}$. The CES and *Responsibility* for τ_2 and τ_8 are also shown. It holds: $\tau_2 \preceq^{CE^{ui}} \tau_8$ and $\tau_8 \preceq^{\rho} \tau_2$. Then, CES and *Responsibility* are not aligned for $(D, \mathcal{Q})$. ■

Notice that, with Proposition 9 and 10, we have fully characterized the space of multi-component queries, that is, for any query with two or more components, we can determine if CES and *Responsibility* are aligned for any possible instance or there exists one such they are not aligned. Following this, we can obtain: *Let $\mathcal{Q}$ be a multi-component query. Then, it holds that, if the scores are always aligned for $\mathcal{Q}$, then the scores are always aligned for each of the sub-queries built from its components.*

Proof. Let $\mathcal{Q}$ be a multi-component BCQ such CES and *Responsibility* are aligned for $(D, \mathcal{Q})$, where D is any possible instance. By Proposition 9 and 10, the query $\mathcal{Q}$ has all its components with a single atom. Then, by Lemma 1, the scores will be always aligned for a query $\mathcal{Q}_i$ with a single atom iff the scores are always for its reduced version $\mathcal{Q}_i^{red}$, a query with the single atom and one variable. Then, by Proposition 7, the scores are always aligned for each query $\mathcal{Q}_i^{red}$, which proves the statement. □

The reversed not always holds, that is, there exists some multi-component query $\mathcal{Q}$, such the scores are always aligned for each sub-queries of $\mathcal{Q}$, but there exists some instance D were the scores are not aligned. In fact, we will provide a query in Example 17 that constitutes a counter example for this.

We have established so far both the positive case of Theorem 1, i.e. alignment of CES and *Responsibility*; and also the negative case, i.e. non-alignment of CES and *Responsibility*, resp. In essence, these two cases correspond to two cases for the query at hand: when it has a single component, and when it has more than one component. Gathering all previous Propositions and Lemmas, we obtain the proof of Theorem1. We conclude next with its formal proof.

Proof of Theorem 1. Consider a BCQ with a single component $\mathcal{Q}$ and its reduced version $\mathcal{Q}^{red}$. Since $|Coin(\mathcal{Q})| = |Var(\mathcal{Q}^{red})|$, Proposition 7 and Proposition 8 that applies to $\mathcal{Q}^{red}$, also holds for $\mathcal{Q}$, by Lemma 1. It follows that, if $\mathcal{Q}$ has a single component, Theorem 1 holds.

Now, for a query with more than one component, Proposition 10 and Proposition 9 holds. It follows that, if $\mathcal{Q}$ has more than one component, Theorem 1 holds. ■

Notice that the positive results given in this section also holds when the instance does not have exogenous tuples, contrary to the non-alignment results, which requires the existence of exogenous tuples. This fact will be investigated in the following section.

7.2 CES vs *Responsibility* Alignment in Absence of Exogenous Tuples

In the preceding section, the (positive) alignment results, those in Theorem 1(a), do not require the presence of exogenous tuples, and they hold with and without them. However, for the non-alignment results, those in Theorem 1(b), exogenous tuples were required, and used in the proof. So far, we do not know if the non-alignment result still holds without exogenous tuples (and the same class of queries). Actually, in this Section, we show that exogenous tuples are indeed required for the non-alignment result for two particular subclasses of the queries considered in Theorem 1(b): for them, CES and *Resposibility* are always aligned. *Accordingly, in this section, we assume that database instances do not have exogenous tuples.*

Proposition 11. Let $\mathcal{Q}$ be a BCQ with a single component, $|Atoms(\mathcal{Q})| \leq 2$, and $|Coin(\mathcal{Q})| \leq 3$. Then, for every database D (without exogenous tuples), CES and *Responsibility* are always aligned for $(\mathcal{Q}, D)$.

Proof. The case $|Atoms(\mathcal{Q})| = 1$ is included in Proposition 7. We first consider a query $\mathcal{Q}$ with a single component and without non-trivial sets of coincident variables. We will extend the result for such query using Lemma 1.

Consider the case $|Atoms(\mathcal{Q})| = 2$, and $|Var(\mathcal{Q})| = 2$ or 3. We do this by showing that the CES and *Responsibility* is aligned for any pair $(D, \mathcal{Q}_{\mathsf{RS}})$, where D is any database, with or without exogenous tuples, and $\mathcal{Q}_{\mathsf{RS}}$ (also used in Example 8) is the following query:

$$\mathcal{Q}_{\mathsf{RS}}\colon\ \exists x \exists y \exists z (R(x,y) \wedge S(y,z)), \tag{7.4}$$

Note that $|Atoms(\mathcal{Q}_{\mathsf{RS}})| = 2$ and $|Var(\mathcal{Q}_{\mathsf{RS}})| = 3$. Moreover, if the CES and *Responsibility* are aligned for this case, the scores will be also aligned for a query $\mathcal{Q}'$ such $|Atoms(\mathcal{Q}')| = 2$ and $|Var(\mathcal{Q}')| = 2$.

W.l.o.g, consider a database D as shown in Table 7.10. where $n_1 < n_2 < m_1 < m_2$. For the moment, all tuples will be considered endogenous. We will denote $r_a = n_1$, $r_b = n_2 - n_1$, $s_a = m_1 - n_2$ and $s_b = m_2 - m_1$. Again, w.l.o.g., we will assume that $r_a \leq s_a$ and $r_b \leq s_b$. Also, denote R_a,R_b,S_a and S_b subsets of D such for all c_i, the tuple (if exists) $R(c_i, a) \in R_a$, $R(c_i, b) \in R_b$, $S(a, c_i) \in S_a$ and $S(b, c_i) \in S_b$. Note that $R_a \cup R_b \cup S_a \cup S_b = D$ and any pairwise intersection of the sets is empty, therefore $\{R_a, R_b, S_a, S_b\}$ can be seen as a *partition* of D. Note that $|R_a| = r_a$, $|R_b| = r_b$, $|S_a| = s_a$ and $|S_b| = s_b$. In addition, for the rest of the proof, τ_{ra}, τ_{rb}, τ_{sa} and τ_{sb} will denote tuples of R_a, R_b, S_a and S_b, respectively.

Note that the values of the CES and *Responsibility* are equal for any two tuples of a particular subset, i.e. for any two tuples τ, τ' in one of the following R_a, R_b, S_a and S_b, $\rho(D, \mathcal{Q}, \tau) = \rho(D, \mathcal{Q}, \tau')$ and $CE^{UI}(D, \mathcal{Q}, \tau) = CE^{UI}(D, \mathcal{Q}, \tau')$. Therefore, in order

R	A	B
τ_1	c_1	a
$\vdots$	$\vdots$	$\vdots$
τ_{n_1}	c_{n_1}	a
τ_{n_1+1}	c_{n_1+1}	b
$\vdots$	$\vdots$	$\vdots$
τ_{n_2}	c_{n_2}	b

S	A	B
τ_{n_2+1}	a	c_{n_2+1}
$\vdots$	$\vdots$	$\vdots$
τ_{m_1}	a	c_{m_1}
τ_{m_1+1}	b	c_{m_1+1}
$\vdots$	$\vdots$	$\vdots$
τ_{m_2}	b	c_{m_2}

TABLE 7.10: Database with relations R and S.

to show that the scores are aligned, the following must hold: $\tau_i \preceq^{\rho} \tau_j$ iff $\tau_i \preceq^{CE^{UI}} \tau_j$ for any $\tau_i \in A, \tau_j \in B$, with $A, B \in \{R_a, R_b, S_a, S_b\}$ and $A \neq B$.

We start by computing the *Responsibility* for each tuple, resulting in the following:

$$\rho(D, \mathcal{Q}_{\mathsf{RS}}, \tau) = \begin{cases} \frac{1}{r_a+r_b} & \text{, if } \tau \in R_a, R_b \\ \frac{1}{r_b+s_a} & \text{, if } \tau \in S_a \\ \frac{1}{r_a+s_b} & \text{, if } \tau \in S_b \end{cases}$$

Observe that: (a) $\tau_{sa} \preceq^{\rho} \tau_r$ for $\tau_r \in R_a \cup R_b$, and (b) $\tau_{sb} \preceq^{\rho} \tau_r$ for $\tau_r \in R_a \cup R_b$. It is also worth noting that, since $\rho(D, \mathcal{Q}_{RS}, \tau_r)$ is the same for all $\tau \in R_a \cup R_b$, the exact value of the CES will be irrelevant.

For the CES, recall that it can be computed using the number of swinging worlds or contingencies according to Equations (6.5) and (6.6). As an example, the CES of $\tau_{ra} \in R_a$ is computed. A set $W \subseteq D$ will be a swinging world of τ_{ra} if: (1) contains a positive number of tuples in S_a and (2) there are only tuples of R_b or S_b. Then, the $CE^{UI}(D, \mathcal{Q}_{\mathsf{RS}}, \tau_{ra}) = \frac{(2^{s_a}-1)\cdot(2^{r_b}+2^{s_b}-1)}{2^{m_2-1}}$. Analogously, the CES for the rest of tuples is computed, resulting in the following:

$$CE^{UI}(D, \mathcal{Q}_{\mathsf{RS}}, \tau) = \begin{cases} \frac{(2^{s_a}-1)\cdot(2^{r_b}+2^{s_b}-1)}{2^{m_2-1}} & \text{, if } \tau \in R_a \\ \frac{(2^{s_b}-1)\cdot(2^{r_a}+2^{s_a}-1)}{2^{m_2-1}} & \text{, if } \tau \in R_b \\ \frac{(2^{r_a}-1)\cdot(2^{r_b}+2^{s_b}-1)}{2^{m_2-1}} & \text{, if } \tau \in S_a \\ \frac{(2^{r_b}-1)\cdot(2^{r_a}+2^{s_a}-1)}{2^{m_2-1}} & \text{, if } \tau \in S_b \end{cases}$$

Note that: (a) $\tau_{sa} \preceq^{CE^{UI}} \tau_{ra}$, and (b) $\tau_{sb} \preceq^{CE^{UI}} \tau_{rb}$. This results coincide with the results (a) and (b) of *Responsibility*.

Having these results, the remaining scenarios are proved: (1) $\tau_{sa} \preceq^{\rho} \tau_{sb}$ iff $\tau_{sa} \preceq^{CE^{UI}} \tau_{sb}$, (2) $\tau_{sb} \preceq^{\rho} \tau_{ra}$ iff $\tau_{sb} \preceq^{CE^{UI}} \tau_{ra}$ and (3) $\tau_{sa} \preceq^{\rho} \tau_{rb}$ iff $\tau_{sa} \preceq^{CE^{UI}} \tau_{rb}$.

For (1), note that $\tau_{sa} \preceq^{\rho} \tau_{sb}$ iff $r_a + s_b \leq r_b + s_a$. Also, $CE^{UI}(D, \mathcal{Q}, \tau_{sa}) \leq CE^{UI}(D, \mathcal{Q}, \tau_{sb})$ iff $2^{s_b+r_a} - 2^{s_b} \leq 2^{s_a+r_b} - 2^{s_a}$. For both, $r_a + s_b = r_b + s_a$ and $r_a + s_b < r_b + s_a$, the scores are *aligned*, since, in the first case, the *Responsibility* is the same for τ_{ra} and τ_{sb} (making the exact value of CES irrelevant) and, in the second case, the induced orders are the same.

For (2), note that $\tau_{sb} \preceq^{\rho} \tau_{ra}$ holds since $r_b \leq s_b$. Regarding the CES , $\tau_{ra} \preceq \tau_{sb}$ iff $2^{r_a} \cdot (2^{r_b} - 1) + 2^{s_a+r_b} \leq 2^{s_a+s_b} + 2^{s_b} \cdot (2^{s_a} - 1)$. Then, for both, $r_b = s_b$ and $r_b < s_b$, the

R_1	A	B
	a	c_1
	$\vdots$	$\vdots$
	a	c_{r_a}
	b	c_{r_a+1}
	$\vdots$	$\vdots$
	b	$c_{r_a+r_b}$

S_1	A
	a
	b

S_2	A
	a
	b

...

S_m	A
	a
	b

TABLE 7.11: Database D_{R_1,S_m} with relations $R_1, S_1, S_2, \ldots, S_m$.

scores are *aligned*, since, in the first case *Responsibility* is the same for τ_{ra} and τ_{sb}, and in the second case the induced orders are the same.

The proof of (3) is analogous to (2), but considering τ_{sa} and τ_{rb} instead of τ_{sb} and τ_{ra}, respectively.

Now, consider a new database D' with extra endogenous tuples. W.l.o.g. it is assumed that $Adom(D) \cap Adom(D') = \emptyset$ and $MSS(D \cup D', \mathcal{Q}) = MSS(D, \mathcal{Q}) \cup MSS(D', \mathcal{Q}) = \emptyset$. Then, by Proposition 2 and Proposition 3, the *Responsibility* and the CES for each tuple $\tau \in D$ will be:

$$\begin{aligned} \rho(D \cup D', \mathcal{Q}_{\mathsf{RS}}, \tau) &= \left((\rho(D, \mathcal{Q}_{\mathsf{RS}}, \tau))^{-1} + \min_{\Gamma \in MAS(D', \mathcal{Q})} |\Gamma| \right)^{-1} \\ CE^{UI}(D \cup D', \mathcal{Q}_{\mathsf{RS}}, \tau) &= CE^{UI}(D, \mathcal{Q}_{\mathsf{RS}}, \tau) \cdot P_{D'}(\mathcal{Q}_{\mathsf{RS}}), \end{aligned}$$

where $P_{D'}(\mathcal{Q}_{\mathsf{RS}})$ is the probability of $\mathcal{Q}_{\mathsf{RS}}$ on the PDB $\langle \mathcal{W}(D'), p^{D'} \rangle$ and $p^{D'}$ the uniform distribution over $\mathcal{W}(D')$. It follows that all of the previously obtained results would hold true also for $D \cup D'$. □

Proposition 12. Consider the BCQ $\mathcal{Q}_{R_1,S_m}$: $\bar{\exists}(R_1(\bar{x}, \bar{y}) \wedge S_1(\bar{x}) \wedge \ldots \wedge S_m(\bar{x}))$, where $\bar{x}$ and $\bar{y}$ are non-empty strings of variables. For it, CES and *Responsibility* are always aligned for every database with an empty set of exogenous tuples.

Proof. Consider the query $\mathcal{Q}^{red}_{R_1,S_m}$ the reduced version of $\mathcal{Q}_{R_1,S_m}$ (see Remark 5), which is given by $\mathcal{Q}^{red}_{R_1,S_m}$: $\bar{\exists}(R_1(x, y) \wedge S_1(x) \wedge \ldots \wedge S_m(x))$, and consider the database D_{R_1,S_m} described in Table 7.11.

Denote with R_a and R_b the set of tuples of the form $R_1(a, x)$ and $R_1(b, x)$, respectively, where a and b are constants and x is any constant. Similarly, denote S_a and S_b the set of tuples of the form $S_i(a)$ and $S_i(b)$, respectively, where $1 \leq i \leq m$. Note $|R_a| = r_a$, $|R_b| = r_b$ and $|S_a| = |S_b| = m$. Moreover, the sets (R_a, R_b, S_a, S_b) forms a partition of D. Also note that $\rho(\tau) = \rho(\tau')$ for any pair of tuples that belongs to the same set R_a, R_b, S_a, S_b. Therefore, we need to prove that $\tau \preceq^{\rho} \tau'$ iff $\tau \preceq^{CE^{UI}} \tau'$ for: (a) $\tau \in R_a$ and $\tau' \in S_a$; (b) $\tau \in R_b$ and $\tau' \in S_b$; (c) $\tau \in R_a$ and $\tau' \in R_b$; (d) $\tau \in S_a$ and $\tau' \in S_b$; (e) $\tau \in R_a$ and $\tau' \in S_b$; (f) $\tau \in R_b$ and $\tau' \in S_b$.

Now, we compute the *Responsiblity* and CES for each tuple in D:

$$\rho(D_{R_1,S_m}, \mathcal{Q}_{R_1,S_m}, \tau) = \begin{cases} 1/(r_a + 1) & \text{, if } \tau \in R_a \\ 1/(r_b + 1) & \text{, if } \tau \in R_b \\ 1/2 & \text{, if } \tau \in S_a, S_b \end{cases}$$

R	A	B
τ_1	a	a
τ_2	b	a
τ_3	b	b
τ_4	b	c
τ_5	b	d
τ_6	b	e
τ_7	c	a
τ_8	c	b
τ_9	c	c
τ_{10}	c	d
τ_{11}	c	e
τ_{12}	c	f
τ_{13}	c	g

S	A
τ_{14}	a
τ_{15}	b
τ_{16}	c

T	A	B
τ_{17}	a	a
τ_{18}	a	b
τ_{19}	a	c
τ_{20}	b	a

U	A
τ_{21}	a
τ_{22}	b

τ	CES	Resp.
τ_1	0.0751	1/3
τ_{17}	0.0754	1/4
τ_{21}	0.5284	1/2

TABLE 7.12: (a) Database D and (b) CES and *Responsibility* τ_1, τ_{17} and τ_{21}.

$$CE^{UI}(D_{R_1,S_m}, \mathcal{Q}_{R_1,S_m}, \tau) = \begin{cases} \frac{(2^{r_b}\cdot(2^m-1)+1)}{2^{r_a+r_b+2m}} & \text{, if } \tau \in R_a \\ \frac{(2^{r_a}\cdot(2^m-1)+1)}{2^{r_a+r_b+2m}} & \text{, if } \tau \in R_b \\ \frac{(2^{r_a}-1)\cdot(2^{r_b}\cdot(2^m-1)+1)}{2^{r_a+r_b+2m}} & \text{, if } \tau \in S_a \\ \frac{(2^{r_b}-1)\cdot(2^{r_a}\cdot(2^m-1)+1)}{2^{r_a+r_b+2m}} & \text{, if } \tau \in S_b \end{cases}$$

Note that $\rho(\tau) \leq \rho(\tau')$ and $CE^{UI}(\tau) \leq CE^{UI}(\tau')$ for any $\tau \in R_a$ and $\tau' \in S_a$. The same holds for $\tau \in R_b$ and $\tau' \in S_b$. This proves case (a) and (b). Moreover, $\rho(\tau) = \rho(\tau')$ for $\tau \in R_a$ and $\tau' \in R_b$, and thus, case (c) holds.

Consider $\tau \in S_a$ and $\tau' \in S_b$. Then, $\rho(\tau) < \rho(\tau')$ iff $r_a > r_b$. Similarly, $CE^{UI}(\tau) < CE^{UI}(\tau')$ iff $r_a > r_b$. Note that if $r_a = r_b$, then $\rho(\tau) = \rho(\tau')$ and $CE^{UI}(\tau) = CE^{UI}(\tau')$. This proves case (d).

Consider $\tau \in R_a$ and $\tau' \in S_b$. Then, if $r_a = 1$, $\rho(\tau) = \rho(\tau')$, and thus, the exact value of CES is not important. Consider now $r_a > 1$, then $\rho(\tau) < \rho(\tau')$. Then, we must show that $CE^{UI}(\tau) \leq CE^{UI}(\tau')$. This latter holds iff $(2^m - 1)\cdot(2^{r_a+r_b} - 2^{r_a} - 2^{r_b}) - 2 \geq 0$. If we substitute $r_b = m = 1$, which are the minimum values that both could take, the following must holds: $2^{r_b}\cdot 3 - 6 \geq 0$, which is always true for $r_b \geq 1$. This proves case (e).

Case (f) is analogous to case (e), but considering $\tau \in R_b$ and $\tau' \in S_a$. □

Now, consider a multi-component query $\mathcal{Q}$. As highlighted in (Dalvi & Suciu, 2012), the probability of $\mathcal{Q}$ being true in a given probabilistic database is equal to the multiplication of the probabilities of each of its components being true, that is, the probabilities of the components are independent. Reasoning in the same way, one may expect that CES and *Responsibility* are always aligned for every multi-component query if the scores are also always aligned for all sub-queries built from the its components. It turns out that this is not the case, as the following counter-example shows.

Example 17. Consider the query $\mathcal{Q}\colon \exists x \exists y \exists z (R(x,y) \wedge S(x) \wedge T(z,w) \wedge U(z))$, and the database with the relations in Table 7.12. It holds: $\tau_1 \preceq^{CE^{UI}} \tau_{17} \preceq^{CE^{UI}} \tau_{21}$ and $\tau_{17} \preceq^{\rho} \tau_1 \preceq^{\rho} \tau_{21}$. This shows that CES and *Responsibility* are not aligned for $(D, \mathcal{Q})$. ■

In this example, that query $\mathcal{Q}$ has the sub-queries: $\mathcal{Q}_a\colon \exists x \exists y (R(x,y) \wedge S(x))$, and $\mathcal{Q}_b : \exists z \exists w (T(z,w) \wedge U(z))$. By Proposition 12, CES and *Responsiblity* are always aligned for the pairs $(D, \mathcal{Q}_a)$ and $(D', \mathcal{Q}_b)$ for any database D and D', with or without exogenous tuples. This fact highlights the alignment property is not preserved for queries with multiple components.

7.3 Summary of Results in Sections 7.1 and 7.2

In order to have a global -and also more accurate view- of the results obtained so far in this section, we summarize them next; highlighting, in particular, some open cases.

1. Positive Results for Alignment:

   a. Proposition 7 and Lemma 1: If a query $\mathcal{Q}$ has a single component, and $|Coin(\mathcal{Q})| = 1$, then CES and *Responsibility* are aligned for every possible instance, with or without exogenous tuples.

   b. Proposition 9: If a query $\mathcal{Q}$ has two or more components, and each component has only one atom, then, the scores are aligned for every possible instance, with or without exogenous tuples.

   c. Propositions 11 and 12: If a query $\mathcal{Q}$ has a single component, with $|Atoms(\mathcal{Q})| \leq 2$, or is the query $\mathcal{Q}_{R_1,S_m}$ (see Proposition 12), then the scores are aligned for every possible instance that does not have exogenous tuples.

2. Negative Results, i.e. for Non-Alignment:

   d. Proposition 8 and Lemma 1: If a query $\mathcal{Q}$ has a single component, and $|Coin(\mathcal{Q})| \geq 2$, then, there is an instance D, with exogenous tuples, such the scores are not aligned for $(D, \mathcal{Q})$.

   e. Proposition 10: If a query $\mathcal{Q}$ has two or more components, with one of them at least with two or more atoms, then, there is an instance, with exogenous tuples, such the scores are not aligned.

Notice the following:

1. For instances with or without exogenous tuples, we are able to determine if the scores are aligned or not for any BCQ based on its syntactic structure. The positive side of this result applies to instances with or without exogenous tuples, however, exogenous tuples were *required* for the proof of the negative side, which means that this side cannot be extended to the setting where instances only contain endogenous tuples.

2. If the instances in item 1.c. are not constrained to only have endogenous tuples, then (by Proposition 8) for any query in a particular sub-class of BCQs, there exists and instance, with exogenous tuples, where the scores are not aligned. This particular sub-class of queries corresponds to all queries with exactly two atoms and two or more coincident sets of variables. For instance, consider the query $\mathcal{Q}_{\mathsf{RS}}$ (see Ex. 8). If exogenous tuples are allowed, we can build an instance, with exogenous tuples, such the scores are not aligned (Prop. 8). However, if the instances only have endogenous tuples, then the scores are always aligned (Prop. 11).

3. The class of queries where the scores are always aligned increases when exogenous tuples became not allowed. For instance, the class of queries from Proposition 7 is included in the class of queries from Proposition 11.

A complete characterization, in terms of the structure of the query, of the alignment of the scores *for instances without exogenous tuples* remains open. In particular, the following problems remain open:

4. *Alignment for queries with a single component:* We do not know whether the scores are aligned or not for queries with three or more atoms (and instances without exogenous tuples). A starting line of investigation on this matter is studying the non-hierarchical queries (see Section 3.1) [2]. For instance, for the single-component and three-atom query $\mathcal{Q}_{\mathsf{RST}}$ in Equation (7.1), it was shown in Example 7, that there is an instance, without exogenous tuples, where the scores are not aligned. We conjecture that, for any non-hierarchical query, there is an instance, without exogenous tuples, where the scores are not aligned. This could be possible since any other non-hierarchical query contains, in some way, the query $\mathcal{Q}_{\mathsf{RST}}$. We speculate that a proof for this would be carried out along the line of Remark 7, but without the use of exogenous tuples.

5. *Alignment for multi-component queries:* There is no characterization, in terms of the structure of a multi-component query, for the alignment of scores, ith the exception of multi-component queries where each component only has 1 atom (Prop. 9).

6. *Preservation of the alignment property for multi-component queries:* As shown with Example 17, the property of alignment is not preserved for multi-component queries, that is, even if the scores are aligned for any possible instance (without exogenous tuples) and for each individual component of a given query $\mathcal{Q}$, it is not guaranteed that the scores will be also aligned for $\mathcal{Q}$. However, it may exists some special class of multi-component queries where the scores are always aligned. This fact, as shown in Corollary 7.1 for instances with exogenous tuples, will immediately imply that the scores are aligned for any sub-query built from its components.

7.4 Non-Alignment of CES and Responsibility with the Shapley Value

In this section we present a negative result about the alignment of the Shapley Value with CES (and also with *Responsibility*). Actually, we establish that, for every query $\mathcal{Q}$ in a broad class of BCQs, there is instance D where each of the pairs of scores are *not aligned* for $(D, \mathcal{Q})$. As for the non-alignment result in Section 7.1, we require the existence of exogenous tuples in D.

In order to prove this result, the technique given in Remark 7 is used. The base (counter)-example to be used will be Example 8. We now state and prove the main result of this section.

Proposition 13. Let $\mathcal{Q}$ be a BCQ, such that it contains two atoms A_R and A_S, with predicates R, S, resp., satisfying: (a) $Var(A_R) \cap Var(A_S) \neq \emptyset$, (b) $Var(A_R) \not\subseteq Var(A_S)$, and (c) $Var(A_S) \not\subseteq Var(A_R)$. There is a database D containing exogenous tuples, such that CES and Shapley Value; and *Responsibility* and Shapley Value are not aligned.

[2]The reduced versions of all non-hierarchical queries are included in the syntactic class given in Proposition 8 and therefore, the exists some instance, with exogenous tuples, such the scores are not aligned.

R_1	A	B	C
τ_1	c_1	a	a
τ_2	c_2	b	b
τ_3	c_3	b	b

R_2	A	B
τ_4	a	a
τ_5	b	b

R_3	A	B	C	D
τ_6	a	a	c_4	c'
τ_7	a	a	c_5	c'
τ_8	b	b	c_6	c'
τ_9	b	b	c_7	c'
τ_{10}	b	b	c_8	c'
τ_{11}	b	b	c_9	c'

R_4	A
τ_{12}	c'

τ	*CES*	*Resp.*	*Shapley*
τ_1	57/256	1/3	400/2520
τ_6	19/256	1/4	151/2520
τ_8	15/256	1/5	169/2520

TABLE 7.13: (a) Database D. (b) CES, *Responsibility* and Shapley Value for $(D, \mathcal{Q})$.

Proof. Let $\mathcal{Q}$ be a BCQ such (a) $Atoms(\mathcal{Q}) \geq 2$ and (b) for two atoms A_R and A_S in $\mathcal{Q}$, $Var(A_R) \cap Var(A_S) \neq \emptyset$, and $Var(A_R) \not\subseteq Var(A_S)$ nor $Var(A_S) \not\subseteq Var(A_R)$. Then, we create a database D to recreate the query $\mathcal{Q}_{\mathsf{RS}}$ and the database $D^\star$ from Example 8 with the query Q. W.l.o.g, we will assume that $|Var(A_R) \cap Var(A_S)| = 1$, which will be denoted by x. Also, consider two different variables $y \in Var(R)$ and $z \in Var(A_S)$. We build the database D, starting from $D^\star$ (from Ex. 8), in the following way:

(a) For each atom $U_y \in Atoms(y)$ and for each tuple τ_R from the relation R of $D^\star$, we create a tuple from U_y by putting in the x and y's position the value of x and y in the tuple τ_R, and replacing by a constant c' the rest of variables. Only the tuples created from A_R are endogenous.

(b) For each atom $U_z \in Atoms(z)$ and for each tuple τ_S from the relation S of $D^\star$, we create a tuple from U_z by putting in the x and z's position the value of x and z in the tuple τ_S, and replacing by a constant c' the rest of variables. Only the tuples created from A_S are endogenous.

(c) For each atom $U_x \in Atoms(x) \setminus (Atoms(y) \cup Atoms(z))$, we create two tuples from U_z by putting in the x's position the constants a and b respectively, and replacing by a constant c' the rest of variables. All tuples created this way will be exogenous.

(d) For each atom $U \notin (Atoms(x) \cup Atoms(y) \cup Atoms(z))$, a tuple is created by replacing all variables in U with a constant c'. All tuples created this way will be exogenous.

By the construction of D we have that each endogenous tuple in it has its corresponding tuple in $D^\star$, and the CES, *Responsibility*, and Shapley Value of each are the same. It follows that both scores are not aligned for $(D, \mathcal{Q})$. □

We now illustrate Proposition 13, in particular, the construction of the database where the scores are not aligned.

Example 18. Consider the BCQ $\mathcal{Q} : \bar{\exists}(R_1(x,y,z) \wedge R_2(y,z) \wedge R_3(y,z,w,v) \wedge R_4(v))$. Now, atoms $R_1(x,y,z)$ and $R_3(y,z,w,v)$ satisfy the conditions of Proposition 13. The database D, for which CES and Shapley Value, so as *Responsibility* and Shapley Value, are not aligned, is built according to the proof of the proposition.

Table 7.13(a) shows the resulting instance; and Table 7.13(b) shows CES, *Responsibility* and Shapley Value for each of the tuples τ_1, τ_6 and τ_8. It holds: $\tau_8 \preceq^{CE^{UI}} \tau_6 \preceq^{CE^{UI}} \tau_1$, $\tau_8 \preceq^{\rho} \tau_6 \preceq^{\rho} \tau_1$ and $\tau_6 \preceq^{Shapley} \tau_8 \preceq^{Shapley} \tau_1$. Then, CES and *Responsibility*, so as CES and Shapley Value, are not aligned for $(D, \mathcal{Q})$. ■

In order to obtain Proposition 13, we require the existence of exogenous tuples in the database. We do not know if the result still holds without exogenous tuples. Actually, this latter case already emerges as a problem in Game Theory, where commonly no distinction between exogenous and endogenous players is made; namely as the problem of the alignment (or not) of the (general) Shapley value and the Banzhaf Power-Index (that in our database setting becomes CES). To the best of our knowledge, this is still an open problem (Freixas, 2010).

Chapter 8

Complexity of Computing CES and GCES

In this section we investigate the complexity of computing CES and GCES [1](for a TID) for *Unions of BCQs* (UBCQs). The results take the form of a dichotomy: computing CES and GCES, for a given query and tuple, is either in PTIME or #P-hard depending on whether the query belongs to a subclass of UBCQs or not.

In (Livshits et al., 2021b), the complexity of computing the Shapley value and the Banzhaf Power-Index (BPI) of a tuple for self-join-free BCQs (and aggregations on top of them) was investigated. It was also established there, that for this class of queries, BPI and CES (with tuple-independence and uniform distribution of $\frac{1}{2}$) coincide. As a consequence, the following dichotomy result was established: For BCQ without self-joins: (a) If the query is hierarchical, then computing CES can be done in PTIME; otherwise, (b) CES computation is $FP^{\#P}$*-complete*. As mentioned at the beginning of this section, we extend this result to the cases of CES and GCES for UBCQs. Formally, the result is the following:

Theorem 2. Let $\mathcal{Q}$ be a UBCQ, D a (non-probabilistic) instance, $\langle \mathcal{W}(D), p^D \rangle$ a TID instance based on D, and $\tau \in D$ an endogenous tuple. Computing $CE^{p^D}(D, \mathcal{Q}, \tau)$ and $CE^{UI}(D, \mathcal{Q}, \tau)$ is either PTIME or #P-hard. Moreover, there exists a syntactic characterization of $\mathcal{Q}$ called *safety* such, if $\mathcal{Q}$ is *safe*, then the scores can be computed in PTIME and if the query is *unsafe*, then the complexity of the scores is #P-hard. ■

The *safety* syntactic characterization mentioned in the theorem is not given by its usual definition: queries that are independent from its domain, rather queries that are *accepted* by the algorithm in (Dalvi & Suciu, 2012) that computes the probability of a query being true. Regarding the present work, we are not interested in the formal definition of this class. Moreover, thorough this section, we will refer to the *safety* class of queries as this way.

We now introduce some useful notation regarding UBCQ. Let $\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_k$ denote a series of BCQ. Now, let $\mathcal{Q}$ be the disjunction of all $\mathcal{Q}_i$, which will be written as:

$$\mathcal{Q} : (Q_1 \vee Q_2 \vee \ldots \vee Q_k)$$

For any UBCQ, it will be assumed that no (existentially quantified) variable in $\mathcal{Q}$ appears in two different BCQ, that is, $Var(Q_i) \cap Var(Q_j) = \emptyset$.

[1]The GCES was defined for an arbitrary distribution over $\mathcal{W}(D)$. All complexity results for the GCES given in this section will be for a TID.

The following example illustrates the use of Theorem 2.

Example 19. Consider the UBCQs

$$\mathcal{Q}: \bar{\exists}(R_1(x_1) \wedge R_2(x_1, y_1, z)) \vee \bar{\exists}(R_2(x_2, y_2) \wedge R_3(y_2))$$
$$\mathcal{Q}': \bar{\exists}(R_1(x_1) \wedge R_2(x_1, y_1)) \vee \bar{\exists}(R_2(x_2, y_2) \wedge R_3(y_2)) \vee \bar{\exists}(R_1(x_3) \wedge R_3(y_3))$$

The query $\mathcal{Q}$ is *unsafe* and the query $\mathcal{Q}'$ is *safe* (according to (Dalvi & Suciu, 2012)). By Theorem 2, given a TID instance$\langle \mathcal{W}(D), p^D \rangle$, with D a database instance, and a endogenous tuple $\tau \in D$, computing $CE^{p^D}(D, \mathcal{Q}, \tau)$ and $CE^{UI}(D, \mathcal{Q}, \tau)$ is #P-hard, and computing $CE^{p^D}(D, \mathcal{Q}', \tau)$ and $CE^{UI}(D, \mathcal{Q}', \tau)$.

As we will later show, the problem of computing the GCES (for a TID) and CES is closely related to the problem of computing the probability of a query in a probabilistic database: Query Evaluation Problem (QEP). In fact, the proof of Theorem 2 relays on two previous complexity results regarding this problem, which we now recall.

Remark 8. *(Theorem 4.21 from (Dalvi & Suciu, 2012)) Consider $\mathcal{Q}$ a UBCQ and a TID instance $D^p = \langle \mathcal{W}(D), p^D \rangle$. Then, one of the following holds:*

(a) If $\mathcal{Q}$ is safe, *then $P_{D^p}(\mathcal{Q})$ can be computed in* PTIME

(b) If $\mathcal{Q}$ is unsafe, *then computing $P_{D^p}(\mathcal{Q})$ is #P-hard.*

where $P_{D^p}(\mathcal{Q})$ denotes the probability of $\mathcal{Q}$ being true in the probabilistic database D^p.

Remark 9. *(Theorem 2.2 from (Kenig & Suciu, 2021)) Consider $\mathcal{Q}$ a UBCQ. Then, for any TID $D^p = \langle \mathcal{W}(D), p^D \rangle$ where $p^D(\{\tau\}) \in \{0, \frac{1}{2}, 1\}$, if $\mathcal{Q}$ is* unsafe, *then computing $P_{D^p}(\mathcal{Q})$ is #P-hard.*

The proof of the positive side of Theorem 2 is simply about computing the two probability distributions, for the CES or GCES from their definitions. By Remark 8, those probabilities can be computed in PTIME for *safe* queries, and therefore, the result holds. To prove the negative side. we provide a reduction from Query Evaluation Problem (QEP) (Dalvi & Suciu, 2007, 2012) to the problem of computing CES and GCES. By Remarks 8 and 9, the complexity of the former is #P-hard, and thus, the result also holds. This reduction is accomplished by the following proposition, which relates the computation of both problems.

Proposition 14. Let $\mathcal{Q}$ be a UBCQ. For every TID $D^p = \langle \mathcal{W}(D), p^D \rangle$ associated to a (non-probabilistic) database D, there is a TID $D^{p'} = \langle \mathcal{W}(D'), p^{D'} \rangle$ with an endogenous tuple $\tau \in D'$, such that $P_{D^p}(\mathcal{Q}) = 1 - CE^{p^{D'}}(D', \mathcal{Q}, \tau)$, where $P_{D^p}(\mathcal{Q})$ denote the probability of $\mathcal{Q}$ being true in the TID D^p; and $D^{p'}$ can be computed in constant time from the original TID.

Proof. Let $\langle \mathcal{W}(D), p^D \rangle$ be a TID instance and $\mathcal{Q}_d$ a UBCQ. Denote $\mathcal{Q}$ be one of the BCQs that are part of $\mathcal{Q}_d$. Additionally, consider a mapping $f : Var(\mathcal{Q}) \to C$, where: (a) C is a set of fresh constants, that is, $Adom(D) \cap C = \emptyset$ and (b) each variable is mapped to a unique constant in C.

Now, we create the new TID instance $\langle \mathcal{W}(D'), p^{D'} \rangle$, being D' an (non-probabilistic) instance, starting from D^p, and adding additional tuples. These new tuples are created as follows: for each $U \in Atoms(\mathcal{Q})$, create a tuple τ_U by replacing each variable v with $f(v)$, and assigning it a probability of 1, that is $p^{D'}(\{\tau_U\}) = 1$. The TID instance that includes D and all of these fresh tuples is denoted by D.

R_1	A	B	P
τ_1	a	a	0.9
τ_2	b	b	0.3
τ_3	c	b	0.8

R_2	A	P
τ_4	d	0.5

R_3	A	B	P
τ_5	a	c	0.1
τ_6	b	d	0.3
τ_7	e	e	0.8
τ_8	f	f	0.5

R_4	A	B	P
τ_9	a	a	0.9
τ_{10}	a	c	0.1
τ_{11}	c	b	0.1

TABLE 8.1: Database D^p.

Now, we compute the GCES for one of the created tuples $\tau_U^\star$.

$$\begin{aligned} CE^{p^{D'}}(D', \mathcal{Q}_d, \tau'_U) &= \mathbb{E}(\mathcal{Q}_d | do(\tau'_U \text{ in})) - \mathbb{E}(\mathcal{Q}_d | do(\tau'_U \text{ out})) \\ &= P_{D^{p'}}(\mathcal{Q}_d | do(\tau'_U \text{ in})) - P_{D^{p'}}(\mathcal{Q}_d | do(\tau'_U \text{ out})), \end{aligned}$$

where $P_{D^{p'}}(\mathcal{Q})$ denotes the probability of $\mathcal{Q}$ being true in the TID $D^{p'}$. Note the following: (a) $P_{D^{p'}}(\mathcal{Q}_d) = 1$, since the created tuples by themselves constitute a MSS (see Def. 3). This probability will not change by the intervention $do(\tau'_U \text{ in})$; and (b) when computing $P_{D^{p'}}(\mathcal{Q}_d | do(\tau'_U \text{ out}))$, the before created tuples become *dummy tuples*, since they only belong to one MSS. It follows that:

$$\begin{aligned} CE^{D^{p'}}(D', \mathcal{Q}_d, \tau'_U) &= P_{D^{p'}}(\mathcal{Q}_d | do(\tau'_U \text{ in})) - P_{D^{p'}}(\mathcal{Q}_d | do(\tau'_U \text{ out})) \\ &= 1 - P_{D^p}(\mathcal{Q}_d) \end{aligned}$$

Then, by rearranging the terms, we obtain the expression in the proposition. Since the new PDB $D^{p'}$ was obtained from D^p with a number of additional tuples depending on the size of the selected BCQ, then, $D^{p'}$ can be obtained in constant time with respect to D^p. □

Notice that Proposition 14 does not make any assumption about the probability distribution p^D other that tuple-independence. It can be uniform or not. Remember that CES uses the uniform distribution. Also, it is worth mentioning that the proof for this proposition requires the existence of some exogenous tuples. We can remove this requirement by using a technique also used in (Deutch et al., 2022) in relation to the Shapley Value. The following example illustrates the use of the proposition.

Example 20. With the same query $\mathcal{Q}_d$ and instance D, consider the TID $D^p = \langle \mathcal{W}(D), p^D \rangle$, where a distribution p^D assigns probabilities to each world according to the probabilities shown in Table 8.1.

According to Proposition 14, we can build a TID $D^{p'} = \langle \mathcal{W}(D'), p^{D'} \rangle$ with a tuple $\tau \in D'$, such that $P_{D^p}(\mathcal{Q}_d) = 1 - CE^{p^{D'}}(D', \mathcal{Q}_d, \tau)$. This TID is built as follows: (a) First, select one of the BCQs from $\mathcal{Q}_d$, say $\mathcal{Q} : \bar{\exists}(R_1(x,y) \wedge R_2(x))$. Next, (b) Map each variable to a fresh constant. Here, x is mapped to c_1, and y to c_2. After that, (c) Create two tuples: τ_{R_1} and τ_{R_2}, the first being the ground atom $R_1(c_1, c_2)$, and the second, $R_2(c_1)$. Each of these two tuples is assigned a probability 1.

It holds $P_{D^p}(\mathcal{Q}_d) = 1 - CE^{D^{p'}}(D', \mathcal{Q}, \tau_{R_1})$. And since, $CE^{p^{D'}}(D', \mathcal{Q}, \tau_{R_1}) = 0.991$, it follows $P_{D^p}(\mathcal{Q}_d) = 1 - 0.991 = 0.009$. ■

It the previous example we created two tuples τ_{R_1} and τ_{R_2}. The first one is used to compute the CES and GCES, and the second one is only used to generate a MSS, and thus, make $P_{D^{p'}}(\mathcal{Q} \mid do(\tau_{R_1} \text{ in})) = 1$. This would be the same if we have selected τ_{R_2} to compute the scores.

We now provide the proof of Theorem 2.

Proof. (of Theorem 2) Let $\mathcal{Q}$ be UBCQ and $D^p = \langle \mathcal{W}, p^D \rangle$ a TID instance. Proving the positive side of this theorem is simple. If $P_{D^p}(\mathcal{Q})$ can be computed in PTIME, then, for a given endogenous tuple $\tau \in D^{en}$, computing $CE^{p^D}(D, \mathcal{Q}, \tau)$ and $CE^{UI}(D, \mathcal{Q}, \tau)$ can also be computed in PTIME. Then, we simply compute $P_\tau(\mathcal{Q})$ and $P_{\neg\tau}(\mathcal{Q})$, where P_τ and $P_{\neg\tau}$ are the same probability distribution as P, but changing the probability of $\{\tau\}$ to 1 and 0, respectively. Now, by Remark 8(a), $P_{D^p}(\mathcal{Q})$ can be done in PTIME if the query is *safe*. It follows that $CE^{p^D}(D, \mathcal{Q}, \tau)$ can be computed in PTIME if $\mathcal{Q}$ is *safe*. This result holds for any arbitrary probability distribution, and thus, computing $CE^{UI}(D, \mathcal{Q}, \tau)$ is also in PTIME.

For the negative side, a reduction from the problem of computing each score to the Query Evaluation Problem is made. This reduction is immediate using Proposition 14, since, having an oracle to compute GCES or CES, we can compute the probability of a query being true in the respective probabilistic database. Then, according to Remark 8(b) and 9, it follows that if $\mathcal{Q}$ is *unsafe*, then computing $CE^{p^D}(D, \mathcal{Q}, \tau)$ and $CE^{UI}(D, \mathcal{Q}, \tau)$, respectively, is #P-hard. □

Chapter 9

Axiomatization of the GCES

In this section we provide an axiomatic characterization of the GCES, that is, we show that the score is the only function satisfying a given set of properties. The GCES considered in this section is the most general one: for an arbitrary probability distribution over all possible worlds of an instance. For the purpose of this section we will consider the class of monotone Boolean queries, which includes that of all UBCQs.

In (Dubey & Shapley, 1979), an axiomatic characterization was given for the well-known Banzhaf Power Index (BPI). In (Livshits et al., 2021b), it was shown that, in the context of query answering in DBs, CES for the case of independent tuples with uniform distribution with parameter $\frac{1}{2}$ coincides with BPI (see also Chapter 6). This implies that there is already an axiomatic characterization for this CES, as a particular case of the GCES. We will show that some of the properties, or axioms, given in (Dubey & Shapley, 1979) also hold for the GCES.

We will start by reviewing the properties of CES, continuing with main result in this section, namely about the *categoricity* of the new set of properties for GCES (Theorem 3).[1] First, we need a definition.

Definition 6. Let $\mathcal{Q}$ be a monotone Boolean query, D a relational instance, and $D^p = \langle \mathcal{W}(D), p^D \rangle$ an associated PDB.

(a) The *power* of $W \subsetneqq D^{en}$ is: $Power(D, \mathcal{Q}, W) := \sum_{\tau \in D^{en}} \Delta(\mathcal{Q}, W, \tau)$, with $\Delta(\mathcal{Q}, W, \tau)$ as in (6.2).

(b) The *power* of a tuple $\tau \in D^{en}$ is: $Power(D, \mathcal{Q}, \tau) := \sum_{W \subseteq D^{en} \smallsetminus \{\tau\}} \Delta(\mathcal{Q}, W, \tau)$.

(c) The *weighted power* of $\tau \in D^{en}$ is:

$$Power^{p^D}(D, \mathcal{Q}, \tau) := \sum_{W \subseteq D^{en} \smallsetminus \{\tau\}} \Delta(\mathcal{Q}, W, \tau) \times p^D(W \cup D^{ex} \cup \{\tau\}).$$

(d) The *total power* of the pair $(D, \mathcal{Q})$ is:

$$Power(D, \mathcal{Q}) := \sum_{W \subsetneqq D^{en}} Power(D, \mathcal{Q}, W),$$

or equivalently: $Power(D, \mathcal{Q}) := \sum_{\tau \in D^{en}} Power(D, \mathcal{Q}, \tau)$.

(e) The *weighted total power* of the pair $(D^p, \mathcal{Q})$ is:

$$Power^{p^D}(D, \mathcal{Q}) = \sum_{W \subseteq D^{en}, W \neq \emptyset} Power(\mathcal{Q}, W, \tau) \times p^D(W \cup D^{ex}).$$ ■

[1] In logic, the notion of *categorical theory* is applied to one that has a single model (modulo isomorphism).

Example 21. Consider the instance $D = \{\tau_1 \colon R(a,b),\ \tau_2 \colon S(a),\ \tau_3 \colon S(b)\}$ with the set of exogenous tuples $D^{ex} = \{\tau_2\}$. Consider the BCQ $\mathcal{Q} \colon \exists x \exists y (R(x,y) \wedge S(x))$; and $W_1 = \{\tau_1\}$. It holds:

$$\begin{aligned} Power(D, \mathcal{Q}, W_1) &= \sum_{\tau \in \{\tau_1, \tau_3\}} \Delta(\mathcal{Q}, W_1, \tau) \\ &= \Delta(\mathcal{Q}, W_1, \tau_1) + \Delta(\mathcal{Q}, W_1, \tau_3) = 0 + 0 = 0. \end{aligned}$$

The *power* of τ_1 (the tuple, which is different than the set $\{\tau_1\}$) is:

$$Power(D, \mathcal{Q}, \tau_1) = \sum_{W \subseteq (D^{en} \smallsetminus \{\tau_1\})} \Delta(D, \mathcal{Q}, \tau_1) = \Delta(\mathcal{Q}, \{\tau_3\}, \tau_1) = 1$$

Consider: $W_2 = \{\tau_3\}$, $W_3 = \{\tau_1, \tau_3\}$, and $W_4 = \emptyset$. The *total power* of $(D, \mathcal{Q})$ is:

$$\begin{aligned} Power(D, \mathcal{Q}) &= \sum_{W \subseteq D^{en}} Power(D, \mathcal{Q}, \tau) \\ &= Power(D, \mathcal{Q}, W_1) + Power(D, \mathcal{Q}, W_2) \\ &\quad + Power(D, \mathcal{Q}, W_3) + Power(D, \mathcal{Q}, W_4) \\ &= 0 + 0 + 0 + 1 = 1. \end{aligned}$$

■

In the following we will assume that we have a fixed relational instance D, and possibly a PDB D^p associated to D. We will also consider the class of monotone Boolean queries, MBQ, for the schema of D. Accordingly, for every $\mathcal{Q} \in MBQ$, and subinstance D' of D, $\mathcal{Q}[D']$ takes the values 0 or 1.

Now, we assume we have a "vectorial score function", ψ that maps $\mathcal{Q} \in MBQ$ to a vector in $\mathbb{R}^{|D^{en}|}$: $\psi(\mathcal{Q}) = \langle \psi_{\tau_1}(\mathcal{Q}), \ldots, \psi_{\tau_N}(\mathcal{Q}) \rangle$, with $D^{en} = \{\tau_1, \ldots, \tau_N\}$ (so, $N = |D^{en}|$). The idea is that D, and possibly D^p, are used to compute the scores $\psi_\tau(\mathcal{Q})$ for tuples $\tau \in D^{en}$. We will investigate the following properties for this function:

- DUM: (for "dummy") If $\tau \in D^{en}$ is a dummy tuple (see Chapter 6), then $\psi_\tau(\mathcal{Q}) = 0$.
- EFF: (for "efficiency") $\sum_{\tau \in D^{en}} \psi_\tau(\mathcal{Q}) = Power(D, \mathcal{Q}) / (2^{N-1})$.
- SYM: (for "symmetry") If $\mathcal{Q}[S \cup D^{ex} \cup \{\tau\}] = \mathcal{Q}[S \cup D^{ex} \cup \{\tau'\}]$ for all sets $S \subseteq D^{en} \smallsetminus \{\tau, \tau'\}$, then $\psi_\tau(\mathcal{Q}) = \psi_{\tau'}(\mathcal{Q})$.
- LIN: (for "linearity") For any two monotone Boolean queries $\mathcal{Q}$ and $\mathcal{Q}'$, $\psi(\mathcal{Q} \vee \mathcal{Q}') + \psi(\mathcal{Q} \wedge \mathcal{Q}') = \psi(\mathcal{Q}) + \psi(\mathcal{Q}')$.

In (Dubey & Shapley, 1979), adapted to our setting, it was shown that there is a unique function $\psi \colon MBQ \to \mathbb{R}^N$ that satisfies the properties DUM, EFF, SYM and LIN, and it corresponds to the BPI (that is, CES under tuple-independence and uniform distribution with parameter $\frac{1}{2}$). The Shapley value also satisfies DUM, SYM and LIN, but not EFF. However, it does satisfy slightly modified version of EFF, where the sum of the value for all tuples is equal to $\mathcal{Q}[D]$ (Shapley, 1953; Aumann & Shapley, 2015).

Regarding GCES, we will show that its satisfies only DUM and LIN. However, it does satisfy a slightly modified versions of SYM and EFF. Example 22 below, illustrates these issues.

Example 22. (ex. 21 cont.) Consider the PDBs $D^p = \langle \mathcal{W}(D), p^D \rangle$ and $D^{p'} = \langle \mathcal{W}(D), p^{D'} \rangle$ associated to instance D in Example 21. They have the following probability distributions p^D and $p^{D'}$, resp.:

$$\begin{array}{ll} p^D(W_1 \cup \{\tau_2\}) = 1/4, & p^{D'}(W_1 \cup \{\tau_2\}) = 1/6, \\ p^D(W_2 \cup \{\tau_2\}) = 1/4, & p^{D'}(W_2 \cup \{\tau_2\}) = 1/3, \\ p^D(W_3 \cup \{\tau_2\}) = 1/4, & p^{D'}(W_3 \cup \{\tau_2\}) = 1/6, \\ p^D(W_4 \cup \{\tau_2\}) = 1/4, & p^{D'}(W_4 \cup \{\tau_2\}) = 1/3, \end{array}$$

and $p^D(W_i) = p^{D'}(W_i) = 0$, for $i = 1,2,3,4$, because those W_i do not contain D^{ex}). Notice that p^D entails tuple independence, and is a uniform probability with parameter $\frac{1}{2}$; in fact: (a) for each tuple $\tau \in D^{en}$, $p_\tau = \sum_{W \in \{W : \tau \in W\}} p^D(W) = \frac{1}{2}$, and (b) for $W \subseteq D^{en}$, $p^D(W \cup D^{ex}) = \prod_{\tau \in W} p^D(\{\tau\}) \times \prod_{\tau \in (D^{en} \smallsetminus W)} (1 - p^D(\{\tau\}))$ (see Section 3.3). However, $p^{D'}$ is not independent nor uniform.

Here, we will use GCES as a score function, and we compute the sum of the GCES of τ_1 and τ_3 for the two PDBs, and check if the DUM and EFF properties hold: (here $N = |D^{en}|$)

$$CE^{p^D}(D, \mathcal{Q}, \tau_1) + CE^{p^D}(D, \mathcal{Q}, \tau_3) = 1/2 + 0 = \frac{Power(D, \mathcal{Q})}{2^{N-1}},$$
$$CE^{p^{D'}}(D, \mathcal{Q}, \tau_1) + CE^{p^{D'}}(D, \mathcal{Q}, \tau_3) = 2/3 + 0 \neq \frac{Power(D, \mathcal{Q})}{2^{N-1}}.$$

We can see that: (a) In both cases, property DUM is satisfied by the GCES, that is, since τ_3 is a dummy tuple, then $CE^{p^D}(D, \mathcal{Q}, \tau_3) = CE^{p^{D'}}(D, \mathcal{Q}, \tau_3) = 0$; (b) GCES satisfies EFF when using D^p, but it does not when using $D^{p'}$. ■

The previous example, motivates us to modify the properties EFF and SYM, proposing their generalized versions:

G-EFF: (for "generalized-efficiency")

$$\sum_{\tau \in D^{en}} \psi_\tau(\mathcal{Q}) = Power^{p^D}(D, \mathcal{Q}) + \sum_{W \subsetneq D^{en}} \sum_{\tau \in D^{en}} \Delta(\mathcal{Q}, W, \tau) \times p^D(W \cup D^{ex}).$$

G-SYM: (for "generalized-symmetry") For tuples $\tau, \tau' \in D^{en}$, if $\Delta(\mathcal{Q}, W, \tau) = \Delta(\mathcal{Q}, W, \tau')$, then

$$\psi_\tau(\mathcal{Q}) - Power^{p^D}(D, \mathcal{Q}, \tau) = \psi_{\tau'}(\mathcal{Q}) - Power^{p^D}(D, \mathcal{Q}, \tau').$$

Notice that now both properties rely on the probability distribution p^D for the PDB at hand. It turns out that, if we set p^D to be the independent and uniform distribution with parameter $\frac{1}{2}$, both properties become their respective non-generalized versions. Now we can state the main result of this section.

Theorem 3. Let $D^p = \langle \mathcal{W}(D), p^D \rangle$ be a PDB associated to a (non-probabilistic) instance D. There is a unique score function ψ from MBQ (the class of monotone Boolean queries in the language associated to D's schema) to a real vector in $\mathbb{R}^N$ (with $N = |D^{en}|$) that satisfies the properties DUM, G-EFF, G-SYM and LIN. Moreover, this function corresponds to GCES. ■

Notice that this theorem does not make any assumption on the distribution p^D. In particular, neither tuple-independence nor uniformity are required.

Before proving this theorem, we need one technical results. First, we need to introduce a particular query and its notation. For a fixed instance D, and $S \subseteq D$, $\mathcal{Q}_S$ denotes the following monotone Boolean query:[2]

$$\mathcal{Q}_S[W] = \begin{cases} 1 & \text{, if } S \subseteq W \\ 0 & \text{, otherwise} \end{cases} \tag{9.1}$$

Example 23. (ex. 21 cont.) Let D, $\mathcal{Q}$ and W_3 be the instance, query and possible world as in that example. Then, query $\mathcal{Q}_{W_3}$ is as follows:

$$\mathcal{Q}_{W_3}[W] = \begin{cases} 1 & \text{, if } W_3 \subseteq W \\ 0 & \text{, otherwise} \end{cases}$$

This query can be expressed as the conjunction of the tuples, as ground atoms, in the W_3, namely as: $\mathcal{Q}_{W_3}: (R(a,b) \wedge S(b))$. ■

Lemma 2. Let D denote an instance, and let $\mathcal{Q}$ be a Boolean monotone query. Additionally, $MSS(D, \mathcal{Q})$ will denote the set of minimal swinging sets. Then, this query can be expressed by:

$$\mathcal{Q}[W] = (\mathcal{Q}_{S_1} \vee \cdots \vee \mathcal{Q}_{S_m})[W]$$

where each $S_i \in MSS(\mathcal{Q})$ for $i \in \{1, \ldots, m\}$ and $\mathcal{Q}_{S_i}$ denote the query in Equation (9.1). That is, all Boolean monotone queries $\mathcal{Q}$ can be written as a disjunction of its minimal swinging sets.[3]

Proof. Consider the following problem as an equivalence of the previous lemma: $\mathcal{Q}[W] = 1$ iff there exists $S_i \in MSS(D, \mathcal{Q})$ such that $S_i \subseteq W$, for $i \in \{1, \ldots, m\}$. We will prove this lemma by contradiction.
Let W^* be a set such $\mathcal{Q}[W^*] = 1$ and $S_i \not\subseteq W$ for all $S_i \in MSS(\mathcal{Q})$. Now two cases arises: (a) if $\mathcal{Q}[W^* \setminus \{\tau\}] = 0$ for all $\tau \in W^*$, then $W^* \in MSS(D, \mathcal{Q})$ or (b) there exists some tuple τ such $\mathcal{Q}[W^* \setminus \{\tau\}] = 1$. If (a) is clearly a contradiction, if (b), we can remove tuples from W^* until case (a) happens, which will lead to a contradiction too. Therefore, W^* cannot exist. □

We now proceed to prove Theorem 3. This proof is similar to the proof in (Dubey & Shapley, 1979) for the BPI.

[2]Notice that, since S is fixed and finite, it can be expressed in the FO language of the schema. It can be written as the conjunction of the tuples, as ground atoms, of the set S.

[3]This Lemma is inspired by results in (Dubey & Shapley, 1979).

Proof. (of Theorem 3) First, we prove the uniqueness of the function ψ, and then, that the GCES satisfies all properties.

Consider $D^p = \langle \mathcal{W}(D), p^D \rangle$ a PDB, with D an (non-probabilistic) instance and a monotone Boolean query $\mathcal{Q}$. Also, consider the set of minimal swinging sets, say $MSS(D, \mathcal{Q}) = \{S_1, \ldots, S_m\}$. By Lemma 2, any monotone Boolean query $\mathcal{Q}$ can be decomposed in queries of the form of the query in Equation (9.1), that is , $\mathcal{Q}[W] = (\mathcal{Q}_{S_1} \vee \cdots \vee \mathcal{Q}_{S_m})[W]$ for any $W \subseteq D$.

First, consider an individual query $\mathcal{Q}_{S_i}$. If an endogenous tuple $\tau \notin S_i$, τ is *dummy tuple*, and by property DUM, $\psi_\tau(\mathcal{Q}_{S_i}) = 0$. Now, by property G-EFF the following holds:

$$\sum_{\tau \in D^{en}} \psi_\tau(\mathcal{Q}_{S_i}) = Power^{p^D}(D, \mathcal{Q}_{S_i}) + \sum_{W \subsetneq D^{en}} \sum_{\tau \in D^{en}} \Delta(\mathcal{Q}_{S_i}, W, \tau) \times p^D(W \cup D^{ex})$$

Notice the following: (a) for any two tuples $\tau, \tau' \in S_i$ and for any $W \subseteq D^{en} \setminus \{\tau, \tau'\}$, $\mathcal{Q}_{S_i}[W \cup D^{ex} \cup \{\tau\}] = \mathcal{Q}_{S_i}[W \cup D^{ex} \cup \{\tau'\}] = 0$, and thus, by G-SYM, for all tuples $\tau, \tau' \in S \cap D^{en}$, the following holds

$$\psi_\tau(\mathcal{Q}_{S_i}) - Power^{p^D}(D, \mathcal{Q}_{S_i}, \tau) = \psi_{\tau'}(\mathcal{Q}_{S_i}) - Power^{p^D}(D, \mathcal{Q}_{S_i}, \tau'),$$

and (b) the weighted total power can be expressed as:

$$Power^{p^D}(D, \mathcal{Q}_{S_i}) = \sum_{\tau \in D^{en}} Power^{p^D}(D, \mathcal{Q}_{S_i}, \tau).$$

Then, for each tuple $\tau \in (S \cap D^{en})$, $\psi_\tau(\mathcal{Q}_{S_i}) - Power^{p^D}(D, \mathcal{Q}_{S_i}, \tau) = k$, where k corresponds to:

$$k = \sum_{\tau \in D^{en}} \sum_{W \subseteq D^{en} \setminus \{\tau\}} \frac{\Delta(\mathcal{Q}_{S_i}, W, \tau) \times p^D(W \cup D^{ex})}{|S \cap D^{en}|}.$$

This expression uniquely defines the function ψ_τ for the query:

$$\psi_\tau(\mathcal{Q}_{S_i}) = \begin{cases} k - Power^{p^D}(D, \mathcal{Q}_{S_i}, \tau) & \text{, if } \tau \in S \cap D^{en} \\ 0 & \text{, otherwise} \end{cases}$$

Now, note that by property LIN, we can recursively obtain $\psi(\mathcal{Q})$ by:

$$\psi(\mathcal{Q}) = \psi(\mathcal{Q}_l \vee \mathcal{Q}_r) = \psi(\mathcal{Q}_l) + \psi(\mathcal{Q}_r) - \psi(\mathcal{Q}_l \wedge \mathcal{Q}_r),$$

where $\mathcal{Q}_l[W] = (\mathcal{Q}_{S_1} \vee \cdots \vee \mathcal{Q}_{S_k})[W]$ and $\mathcal{Q}_r[W] = (\mathcal{Q}_{S_{k+1}} \vee \cdots \vee \mathcal{Q}_{S_m})[W]$, with $1 \leq k < m$ and $W \subseteq D$. Then, as $\psi_\tau(\mathcal{Q}_{S_i})$ is uniquely defined for any $S_i \in MSS(D, \mathcal{Q})$ and $\tau \in D^{en}$, then $\psi_\tau(\mathcal{Q})$ is uniquely defined for each $\tau \in D^{en}$.

After showing the uniqueness, we show that GCES satisfies all the given properties for a given PDB $D^p = \langle \mathcal{W}(D), p^D \rangle$.

First, note that, for a given Boolean monotone query $\mathcal{Q}$ and an endogenous tuple $\tau \in D$, the GCES for can be written as:

$$CE^{p^D}(D, \mathcal{Q}, \tau) = \sum_{W \subseteq D^{en} \setminus \{\tau\}} \Delta(\mathcal{Q}, W, \tau) \times (p^D(W \cup D^{ex}) + p^D(W \cup D^{ex} \cup \{\tau\})). \quad (9.2)$$

The property DUM is trivial, since if a tuple τ does not contribute to any tuple, then $\Delta(\mathcal{Q}, W, \tau) = 0$ for all worlds. Property G-EFF can be shown by algebraic manipulation:

$$\begin{aligned}\sum_{\tau \in D^{en}} CE^{p^D}(D, \mathcal{Q}, \tau) &= \sum_{\tau \in D^{en}} \sum_{W \subseteq D^{en} \setminus \{\tau\}} \Delta(\mathcal{Q}, W, \tau) \times (p^D(W \cup D^{ex}) + p^D(W \cup D^{ex} \cup \{\tau\})) \\ &= Power^{p^D}(D, \mathcal{Q}, \tau) + \sum_{\tau \in D^{en}} \sum_{W \subseteq D^{en} \setminus \{\tau\}} + p^D(W \cup D^{ex} \cup \{\tau\}))\end{aligned}$$

Then, by re-arranging the order of the sums, GCES satisfies property G-EFF.

For property G-SYM, note that $p^D(W \cup D^{ex})$ in Eq. 9.2 do not depend on the tuple τ and therefore, if, for two tuples $\tau, \tau' \in D^{en}$, $\Delta(\mathcal{Q}, W, \tau) = \Delta(\mathcal{Q}, W, \tau')$ for any $W \subseteq D \setminus \{\tau, \tau'\}$, then the sum of $p(W \cup D^{ex})$ will for both tuples will be also equal, that is:

$$\sum_{W \subseteq D^{en} \setminus \{\tau\}} \Delta(\mathcal{Q}, W, \tau) \times p^D(W \cup D^{ex}) = \sum_{W \subseteq D^{en} \setminus \{\tau'\}} \Delta(\mathcal{Q}, W, \tau') \times p^D(W \cup D^{ex}),$$

and since, for a tuple $\tau \in D^{en}$ we have that $CE^{p^D}(D, \mathcal{Q}, \tau) = Power^{p^D}(D, \mathcal{Q}, \tau) + \sum_{W \subseteq D^{en} \setminus \{\tau\}} \Delta(\mathcal{Q}, W, \tau) \times p^D(W \cup D^{ex})$, GCES satisfies G-SYM. For LIN, let $\mathcal{Q}$ and $\mathcal{Q}'$ be two monotone Boolean queries and denote with $\mathcal{Q} \wedge \mathcal{Q}'$ and $\mathcal{Q} \vee \mathcal{Q}'$ their conjunction and disjunction, respectively. Notice that the expression $p^D(W \cup D^{ex}) + p^D(W \cup D^{ex} \cup \{\tau\})$ does not depend on the query, only on the possible world W. Then, it is immediate that, for a given PDB instance D^p and any given endogenous tuple $\tau \in D^{en}$, the following holds:

$$CE^{p^D}(D^p, \mathcal{Q} \wedge \mathcal{Q}', \tau) + CE^{p^D}(D^p, \mathcal{Q} \vee \mathcal{Q}', \tau) = CE^{p^D}(D^p, \mathcal{Q}, \tau) + CE^{p^D}(D^p, \mathcal{Q}', \tau)$$

□

Chapter 10

Discussion and Conclusions

So far in this thesis, we addressed the problem of providing explanations in the context of probabilistic databases by introducing the Generalized Causal-Effect Score (GCES), This score extends the existing Causal-Effect Score (CES) to accommodate arbitrary probability distributions, which is useful in scenarios where tuple dependency or non-uniformity is present. This work also investigated the alignment CES with other well-known scores, such as Responsibility and Shapley Value, in non-probabilistic databases, establishing for each case some syntactic classes of queries for which the scores are aligned or not.

Furthermore, we study the computational complexity of computing GCES and CES, establishing in both cases a dichotomy: the computation of each can be done in PTIME or is #P-hard. We also provide a set of properties that uniquely defines GCES, similar to the existing one for the CES (BPI).

There are several research directions that have been left open by our work. They are matter of ongoing and future investigation. In the following we mention some of them.

Score Computation. Despite the intrinsic high complexity of computing the CES and GCES, it is worth exploring efficient or approximate algorithms for their computation, possibly for some interesting special cases of queries.

Aggregate Queries and Multidimensional DBs. In this work we have considered conjunctive queries. A natural extension is dealing queries with aggregation on CQs. We made early on the case for the convenience of CES for this kind of queries. Even more interesting would be dealing with aggregations and the associated scores in a multidimensional database, as in OLAP; at different levels of data and causality *abstraction*.

Semantic Constraints. *Responsibility* has been introduced and investigated in the presence of integrity constraints (Bertossi & Salimi, 2017a). This is something still to be done for the CE score, in its basic or generalized versions. Particularly interesting becomes dealing with constraints in probabilistic DBs (Suciu, 2020).

Alignment of the GCES. In Chapter 7 the alignment of CES with other scores was investigated. However, the generalized CES was not considered, mainly due to the presence of arbitrary distributions in GCES, that do not appear in *Responsibility* or

Shapley (as used so far in data management).[1] However, we could naturally extend *Responsibility* with a probabilistic component. Actually, this was done in (Bertossi et al., 2020) for its application to counterfactual explanations for ML-based classification. After doing that for DBs, an analysis of alignment could be attempted.

Attribute-Level GCES. We have defined and investigated the GCES at the tuple level. It would be interesting to extend its definition and investigation in order to quantify the causal effect of an attribute value in a tuple. This case is challenging in that it is not only about making an attribute value true or false anymore. Making the latter false may lead to consider multiple alternative values, as has been done for *Responsibility* (Bertossi et al., 2020).

GCES of Missing Values. Missing values (MVs) in DBs have been investigated from a causal point of view, leading to general PDBs (Bertossi, Buron, Moulay, & Toumani, n.d.). However, no attempt has been made to quantify the effect of MVs on query answering. The *possible worlds semantics* considered in (Bertossi et al., n.d.) leads considering multiple (real) values for a MV, and a GCES defined at the attribute level (see previous item).

GES Monotonicity. It would be interesting to investigate more deeply the monotonicity properties (or the lack thereof) of the GCES. For example, under what changes in the database, or in the tuple probabilities, the GCES of tuples change accordingly (or the other way around).

GES Robustness. It would be interesting to analyze the *robustness* of the Generalized CES, under small variations of different parameters, most prominently, of the distribution of the probabilistic DB. Analysis of *SHAP* in this regard has been attempted (Alvarez-Melis & Jaakkola, 2018), in particular, with changing underlying distribution on the entity population (Cifuentes et al., 2024). For robustness in XAI, see also (Huang & Marques-Silva, 2023).

Explanation Scores in Probabilistic DBs. The focus of our work has been the Causal-Effect Score, including some comparisons with other scores used in data management. However, problem of formalizing and investigating the use of *Responsibility* and the Shapley value in probabilistic DBs naturally arises. For Shapley in TIDs see (Karmarkar, Monet, Senellart, & Bressan, 2024).

[1] The *SHAP* version of Shapley as used in Explainable AI does have this probabilistic component (Lundberg & Lee, 2017).

References

Aas, K., Jullum, M., & Loland, L. (2021). Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, *298*, 103502. pages

Abramovich, O., Deutch, D., Frost, N., Kara, A., & Olteanu, D. (2024). Banzhaf values for facts in query answering. In *Proc. sigmod.* pages

Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. In *2018 icml workshop on human interpretability in machine learning (whi 2018).* (arXiv 1806.08049) pages

Arad, D., Deutch, D., & Frost, N. (2022). Learnshapley: Learning to predict rankings of facts contribution based on query logs. In *Proc. cikm* (p. 4788-4792). pages

Arenas, M., Barcelo, P., Bertossi, L., & Monet, M. (2023). On the complexity of shap-score-based explanations: Tractability via knowledge compilation and non-approximability results. *Journal of Machine Learning Research*, *24*(63), 1-58. pages

Aumann, R. J., & Shapley, L. S. (2015). *Values of non-atomic games*. Princeton University Press. pages

Banzhaf III, J. (1964). Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, *19*, 31. pages

Bertossi, L. (2021). Specifying and computing causes for query answers in databases via database repairs and repair programs. *Knowledge and Information Systems*, *63*(1), 199-231. pages

Bertossi, L. (2023a). Attribution-scores and causal counterfactuals as explanations in artificial intelligence. In *Reasoning web: Causality, explanations and declarative knowledge* (Vol. 13759). Springer LNCS. pages

Bertossi, L. (2023b). Attribution-scores in data management and explainable machine learning. In *Proc. adbis'23* (Vol. 13985, p. 16-33). Springer LNCS. pages

Bertossi, L. (2023c). Declarative approaches to counterfactual explanations for classification. *Theory and Practice of Logic Programming*, *23*(3), 559-593. pages

Bertossi, L. (2023d). From database repairs to causality in databases and beyond. *Transactions on Large-Scale Data- and Knowledge-Centered Systems LIV (TLDKS)*, *14160*, 119-131. pages

Bertossi, L., Buron, M., Moulay, I., & Toumani, F. (n.d.). Query answering in incomplete databases under causal specifications of missingness mechanisms. *Forthcoming*. pages

Bertossi, L., Kimelfeld, B., Livshits, E., & Monet, M. (2023). The shapley value in database management. *ACM Sigmod Record*, *52*(2), 6-17. pages

Bertossi, L., Li, J., Schleich, M., Suciu, D., & Vagena, Z. (2020). Causality-based explanation of classification outcomes. In *Proc. 4th international workshop on "data management for end-to-end machine learning" (deem) at acm sigmod/pods* (pp. Article 6, pp 1-10). pages

Bertossi, L., & Salimi, B. (2017a). Causes for query answers from databases: Datalog abduction, view-updates, and integrity constraints. *International Journal of Approximate Reasoning*, *90*, 226-252. pages

Bertossi, L., & Salimi, B. (2017b). From causes for database queries to repairs and model-based diagnosis and back. *Theory of Computing Systems*, *61*(1), 191-232. pages

Chockler, H., & Halpern, J. (2004). Responsibility and blame: A structural-model approach. *J. Artif. Intell. Res.*, *22*, 93-115. pages

Cifuentes, S., Bertossi, L., Pardal, N., Abriola, S., Martinez, M. V., & Romero, M. (2024). The distributional uncertainty of the shap score in explainable machine learning. *ArXiv paper 2401.12731.* pages

Dalvi, N., & Suciu, D. (2007). Efficient query evaluation on probabilistic databases. *VLDB J.*, *16*, 523-544. pages

Dalvi, N., & Suciu, D. (2012). The dichotomy of probabilistic inference for unions of conjunctive queries. *Journal of the ACM*, *59*(30), 1-87. pages

Davidson, S., Deutch, D., Frost, N., Kimelfeld, B., Koren, O., & Monet, M. (2022). Shapgraph: An holistic view of explanations through provenance graphs and shapley values. In *Sigmod conference* (p. 2373-2376). pages

Deutch, D., Frost, N., Kimelfeld, B., & Monet, M. (2022). Computing the shapley value of facts in query answering. In *Proc. sigmod* (p. 1570-1583). pages

Dubey, P., & Shapley, L. (1979). Mathematical properties of the banzhaf power index. *Math. Oper. Res.*, *4*, 99-131. pages

Freixas, J. (2010). On ordinal equivalence of the shapley and banzhaf values for cooperative games. *Int. J. Game Theory*, *39*, 513-527. pages

Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. Cambridge Univ. Press. pages

Gribkoff, E., Van den Broeck, G., & Suciu, D. (2014). The most probable database problem. In *Proc. sigmod-ws buda* (p. 1-7). pages

Halpern, J. (2015). A modification of the halpern-pearl definition of causality. In *Proc. ijcai* (p. 3022-3033). pages

Halpern, J. (2016). *Actual causality*. MIT Press. pages

Halpern, J., & Pearl, J. (2005). Causes and explanations: A structural-model approach. part i: Causes. *The British Journal for the Philosophy of Science*, *56*(4), 843-887. pages

Holland, P. W. (1986). Statistics and causal inference. *Journal of the American Statistical Association*, *81*(396), 945-960. pages

Huang, X., & Marques-Silva, J. (2023). From robustness to explainability and back again. *ArXiv paper 2306.03048*. pages

Huang, X., & Marques-Silva, J. (2024). A refutation of shapley values for explainability. *ArXiv paper 2309.03041*. pages

Hunter, A., & Konieczny, S. (2010). On the measure of conflicts: Shapley inconsistency values. *Artif. Intell.*, *174*(14), 1007-1026. pages

Kanagal, B., & Deshpande, A. (2010). Lineage processing over correlated probabilistic databases. In *Proc. sigmod* (p. 675-686). pages

Kara, A., Olteanu, D., & Suciu, D. (2024). From shapley value to model counting and back. In *Proc. pods.* pages

Karmarkar, P., Monet, M., Senellart, P., & Bressan, S. (2024). Expected shapley-like scores of boolean functions: Complexity and applications to probabilistic databases. In *Proc. acm manag. data* (Vol. 2, p. Article 92). pages

Kenig, B., & Suciu, D. (2021). A dichotomy for the generalized model counting problem for unions of conjunctive queries. In *Proc. pods* (p. 312-324). pages

Livshits, E., Bertossi, L., Kimelfeld, B., & Sebag, M. (2021a). Query games in databases. *ACM Sigmod Record*, *50*(1), 78-85. pages

Livshits, E., Bertossi, L., Kimelfeld, B., & Sebag, M. (2021b). The shapley value of tuples in query answering. *Logical Methods in Computer Science*, *17*(3), 22.1-22.33. pages

Lundberg, S., & Lee, S. (2017). A unified approach to interpreting model predictions. In *Proc. neurips* (p. 4765-4774). pages

Makhija, N., & Gatterbauer, W. (2023). A unified approach for resilience and causal responsibility with integer linear programming (ilp) and lp relaxations. In *Proc. sigmod* (p. 228:1-228:27). pages

Meliou, A., Gatterbauer, W., Halpern, J., Koch, C., Moore, K., & Suciu, D. (2010). Causality in databases. *IEEE Data Eng. Bull.*, 59-67. pages

Meliou, A., Gatterbauer, W., Moore, K., & Suciu, D. (2010). The complexity of causality and responsibility for query answers and non-answers. In *Proc. vldb* (Vol. 10, p. 34-45). pages

Pearl, J. (2009). *Causality: Models, reasoning and inference*. Cambridge University Press. pages

Roth, A. (Ed.). (1988). The shapley value: Essays in honor of lloyd s. shapley. pages

Roy, S., & Salimi, B. (2023). Causal inference in data analysis with applications to fairness and explanations. In L. Bertossi & G. Xiao (Eds.), *Reasoning web. causality, explanations and declarative knowledge* (Vol. 13759, p. 105-131). Springer LNCS. pages

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, *66*, 688-701. pages

Salimi, B., Bertossi, L., Suciu, D., & Van den Broeck, G. (2016). Quantifying causal effects on query answering in databases. In *Proc. 8th usenix workshop on the theory and practice of provenance (tapp'16).* pages

Sen, P., & Deshpande, A. (2007). Representing and querying correlated tuples in probabilistic databases. In *Proc. icde* (p. 596-605). pages

Sen, P., Deshpande, A., & Getoor, L. (2009). Prdb: Managing and exploiting rich correlations in probabilistic databases. *VLDB Journal*, *18*, 1065-1090. pages

Shapley, L. (1953). A value for an n-person game. , 307-331. pages

Suciu, D. (2020). Probabilistic databases for all. In *Proc. pods* (p. 19-31). pages

Suciu, D., Olteanu, D., Ré, C., & Koch, C. (2011). *Probabilistic databases*. Synthesis Lectures on Data Management, Morgan Claypool Pubs. pages