Foundations of Uncertain-Data Integration*

Parag Agrawal[#], Anish Das Sarma[+], Jeffrey Ullman[#], and Jennifer Widom[#]
[#]*Stanford University,* [+]*Yahoo! Research*

ABSTRACT

There has been considerable past work studying data integration and uncertain data in isolation. We develop the foundations for *local-as-view* (LAV) data integration when the sources being integrated are uncertain. We motivate two distinct settings for uncertain-data integration. We then define *containment* of uncertain databases in these settings, which allows us to express uncertain sources as views over a virtual mediated uncertain database. Next, we define *consistency* of a set of uncertain sources and show intractability of consistency-checking. We identify an interesting special case for which consistency-checking is polynomial. Finally, the notion of *certain answers* from traditional LAV data integration does not generalize to the uncertain setting, so we define a corresponding notion of *correct answers*.

1. INTRODUCTION

Many modern applications such as information extraction, deduplication and data cleaning, sensor deployments, and scientific databases generate *uncertain data*. While there has been a flurry of recent work focusing on modeling and managing uncertain data [4, 5, 6, 8, 19, 32, 35], little work has been done on integrating uncertain data. Similarly, several decades of research have focused on the theory and practice of data integration [23], but only considering integration of certain data. This paper develops theoretical foundations for *local-as-view* (LAV) integration [22] of uncertain data.

The combined study of data integration and data uncertainty is important for several reasons. The traditional benefits of data integration still apply when sources are uncertain: Integrating data from multiple sources allows a uniform query interface to access their combined information. In addition, integrating multiple sources of uncertain data may help resolve portions of the uncertainty, yielding more accurate results than any of the individual sources. As a very simple example, if one sensor reports that an object is either in location A or in location B, and a second sensor says it is either in location B or in location C, by integrating the sensor reports we may conclude that the object is in location B.

Even when the sources are certain, data integration may introduce uncertainty. For example, different data capturing redundant information may disagree on some attributes. Furthermore, data integration often relies on *mappings* between sources [23]. One approach has been to use automatically-generated *probabilistic mappings*, which introduce uncertainty. Reference [12] points out that uncertainty introduced during data integration can equivalently be treated as integration of uncertain sources.

Now let us see why models for uncertain data do not adapt directly to the context of data integration. In general, the semantics of uncertain databases is based on *possible worlds* [2]: an uncertain database represents a set of possible certain databases. (We do not consider *confidence values* or continuous probability distributions over possible worlds in this paper.) Now consider integrating multiple uncertain databases. The natural extension might be to consider all combinations of all possible worlds across sources, but this approach can yield undesirable results. Intuitively, we would instead like to preserve and combine possible worlds that corroborate each other, while discarding possible worlds that are contradicted. Returning to our very simple sensor example, one source gives the set of possible worlds $\{A, B\}$ and the other gives $\{B, C\}$. Combining all possible worlds yields (A, B), (A, C), (B, B), and (B, C). Since the two sensors are describing the same real-world object, we prefer to discard all combinations except (B, B). As we will see, there are several challenges to generalizing and formalizing this intuition to solve the overall data integration problem.

We consider specifically the local-as-view (LAV) setting of data integration [22]. In this setting, there is a single logical *mediated database*, and each data source is mapped to its *mediated schema* by specifying the source as a (logical) view. Queries over the virtual mediated database are answered using these mappings. Formalizing LAV data integration over uncertain data requires redefining the two theoretical foundations of the LAV approach [22]: *containment* and *certain answers*. In addition, uncertain data requires us to introduce a formal notion of *consistency* of a set of sources. Next we summarize our contributions in the context of these three building blocks, with details provided in the subsequent sections.

Containment

LAV data integration typically uses the *open world* assumption: Consider a mapping view query Q for a source S. When Q is applied to the (logical) mediated database, we do not require the result to be S exactly, but only require it to *contain* S. For the case of certain databases, containment is straightforward. To extend LAV data integration to the uncertain data setting, we need to find an appropriate definition of containment. We will see that by defin-

*This work was supported by the National Science Foundation under grants IIS-0414762 and IIS-0904497 and by a Stanford Graduate Fellowship from Cisco Systems.

Proceedings of the VLDB Endowment, Vol. 3, No. 1

ing containment carefully, we can capture the "contradiction" and "corroboration" intuition motivated above. (In some sense, defining containment is the most important and fundamental contribution of this paper.)

We will see that for uncertain data, two different integration settings require two somewhat different notions of containment. In one setting, which we call *equality-containment*, the sources were derived from an existing but unknown uncertain mediated database that we are trying to reconstruct. In the other setting, which we call *superset-containment*, there is no actual mediated database from which the sources were derived, so our goal is to come up with a logical mediated database that captures the information from the sources. We will give examples to illustrate the differences. This distinction is new for handling uncertain data. For certain data, these two settings can be handled identically.

Consistency

When sources contain uncertain data, we need to define what it means for sources to be *consistent*. (As an extremely simple example, one sensor reporting location A or B and the other reporting C or D for the same object is inconsistent.) Informally, a set of sources is consistent if there exists a mediated database that contains all sources. We will formalize this notion and then study the problem of consistency-checking under both equality-containment and superset-containment. We show that in general, consistency-checking is NP-hard in the size of the view schema for both of our settings.

Next we identify a class of sources where consistency-checking is polynomial. We describe the construction of a hypergraph for a set of sources, and we provide a PTIME consistency-checking algorithm when this induced hypergraph is acyclic. We also show that the *extensional complexity* of consistency-checking is PTIME for both of our settings.

Query Answers

Lastly, we consider the problem of defining correct query answers over mediated uncertain databases. Once again, the definitions used for certain databases do not adapt directly to the uncertain case. The conventional LAV setting uses *certain answers* (where the use of the word "certain" here is not to be confused with certain data). A certain answer is a set of tuples that is guaranteed to be contained in any mediated database [22]. We define a corresponding notion for uncertain data, which we call *correct answers*, that incorporates possible worlds through our containment definitions. Further, we seek to find a unique *strongest correct answer (SCA)* defined using the partial orders implied by the containment definitions. For superset-containment, we prove by construction the existence of an SCA. However, for equality-containment an SCA does not always exist, and hence we define a relaxed notion for the "best" query answer.

Discussion

For ease of presentation, we restrict ourselves to identity views and queries for most of the paper. In Section 7, we extend our techniques for monotonic views with some restrictions, and for monotonic queries over uncertain data.

The results in this paper are independent of the specific representation used for uncertain data. (The computational complexity of certain problems considered may depend on the specific representation, and we point out these differences in the relevant places.) Also, although our results are presented for discrete uncertain data with a finite set of possible worlds, they can be generalized for *continuous* uncertain data with an infinite set of possible worlds. We emphasize that our foundations are defined in terms of possible worlds, but we neither rely on nor advocate possible worlds as an actual representation of uncertain data. Obviously, we would represent actual uncertain data in more succinct data models, such as in [2, 4, 5, 8, 10, 19, 24, 32].

Finally, note that the primary goal of this paper is to create a theoretical basis for data integration with uncertainty, and to establish a solid foundation for additional investigation. Section 9 suggests numerous concrete directions for future research.

2. CONTAINMENT

This section introduces and formalizes the notions of equality-containment and superset-containment. We begin by reviewing uncertain databases defined in terms of possible worlds and motivate some intuition about them (Section 2.1). We then present our containment definitions (Section 2.2).

2.1 Uncertain Databases

To simplify presentation, we assume possible worlds are sets of tuples in a single relation; extension to the general multi-relation case is straightforward.

DEFINITION 1 (UNCERTAIN DATABASE). *An uncertain database U consists of a finite set of tuples $T(U)$ and a nonempty set of possible worlds $PW(U) = \{D^1, \ldots, D^m\}$, where each $D^i \subseteq T(U)$ is a certain database.*

The possible worlds of an uncertain database U contain information about the tuples in $T(U)$. Intuitively, one of U's possible worlds captures the true database with respect to the tuples in $T(U)$. Consistent with traditional *information theory* [9], an uncertain database with fewer possible worlds contains more information than an uncertain database with more possible worlds, if they contain information about the same set of tuples. When T is omitted in the description of an uncertain database U, it is assumed to be the union of all possible worlds in U.

EXAMPLE 1. *Consider tuples A and B, and the four possible worlds: $P_1 = \emptyset$, $P_2 = \{A\}$, $P_3 = \{B\}$, and $P_4 = \{A, B\}$. An uncertain database U_1 that contains no information about the existence and co-existence of tuples A and B consists of all four possible worlds. An uncertain database U_2 with information that at least one of A or B exists, and that they cannot co-exist, does not contain either P_1 or P_4 . U_2 contains more information than U_1 since it asserts that P_1 and P_4 are not possible. An uncertain database U_3 with $T(U_3) = \{A, B\}$ but containing only the possible world $P_2 = \{A\}$ asserts that the tuple A is contained in the database and that B cannot be contained in the database since B is contained in $T(U_3)$ but not contained in any possible world of U_3. U_3 contains more information than either U_1 or U_2.*

Informally, the information content in an uncertain database U may be thought of as composed of two components: (1) *data-information*, represented by the set of tuples $T(U)$. More tuples indicate more information. (2) *Specificity-information*, represented by the possible worlds $PW(U)$. Fewer possible worlds indicate more information.

2.2 Containment Definitions

Here we motivate and define the notions of equality-containment and superset-containment and illustrate them with simple examples. In Section 3, we give a lengthier example to present further practical motivation for our two definitions.

Equality-Containment

Equality-containment integration is relevant in situations where each source has access only to a portion of an uncertain database that is existing but unknown. There are many real-world applications where access is controlled, and only slices of the data may be visible to various parties (sources). For example, an actual uncertain database may be hidden behind a web service, or people may only be given access to data depending on what they pay for. The goal of data integration in this setting is to answer queries using the best (virtual) reconstruction of the unknown actual uncertain database. Also, when smaller pieces of sensitive data are given out to multiple people so that no single piece leaks information, this reconstruction allows one to detect whether the pieces can be combined to obtain sensitive information. Finally, this setting captures the problem of answering queries using materialized views over uncertain data, where each source is a view.

DEFINITION 2 (EQUALITY-CONTAINMENT). *Consider uncertain databases U_1 and U_2. We say that U_2* equality-contains *U_1, denoted $U_1 \sqsubseteq_E U_2$, if and only if:*

$$\begin{aligned} T(U_1) &\subseteq T(U_2) \\ &\textit{and} \\ PW(U_1) &= \{W \cap T(U_1) \mid W \in PW(U_2)\} \end{aligned}$$

Informally, if we remove from any possible world of U_2 those tuples not contained in $T(U_1)$, then the resulting possible world is a world of U_1, and U_1 may not contain additional possible worlds.

Superset-Containment

Superset-containment integration is relevant in settings where we obtain uncertain data about the real world from different sources, and the goal is to combine information from these sources to construct a logical "real-world truth" as accurately as possible. The simplest example of this scenario was given in Section 1, where one sensor reported A or B for an object and another reported B or C. When we integrate these sources to obtain our best guess at the real-world truth, we decide the location is likely to be B.

Superset-containment also arises in information extraction: several parties may extract structured data from unstructured data (e.g., extracting relations from text, or extracting text in an OCR context) using different techniques, and integration can be used to resolve uncertain results from the sources. Another setting where superset-containment integration is relevant is the combination of information from multiple sources that attempt to make predictions, such as weather forecasts from different websites, or sales projections using different techniques.

In contrast to equality-containment, under superset-containment the sources may not have been derived from an actual uncertain database.

DEFINITION 3 (SUPERSET-CONTAINMENT). *Consider uncertain databases U_1 and U_2. We say that U_2* superset-contains *U_1, denoted $U_1 \sqsubseteq_S U_2$, if and only if:*

$$\begin{aligned} T(U_1) &\subseteq T(U_2) \\ &\textit{and} \\ PW(U_1) &\supseteq \{W \cap T(U_1) \mid W \in PW(U_2)\} \end{aligned}$$

Superset-containment differs from equality-containment in that U_1 may contain possible worlds that are not obtained by intersecting a possible world of U_2 with $T(U_1)$. While this definition may seem counter-intuitive at first glance, recall from Section 2.1 our intuition that an uncertain database with more possible worlds contains less information.

We shall use $\sqsubseteq$ when we refer to either $\sqsubseteq_S$ or $\sqsubseteq_E$.

Power Domains Correspondence

We now discuss how our containment definitions relate to notions in the theory of *Power Domains* [3, 20]. Specifically, we demonstrate that superset-containment and equality-containment correspond to *Smyth order* and *Plotkin order* respectively: Smyth and Plotkin orders are frequently-used in many applications[1] as devices to "lift" a partial order defined over elements of a set S to finite subsets of S.

Consider a possible world W in an uncertain database with tuple set T. Tuples from T that are absent from W also represent information, so we consider possible worlds in the context of the overall tuple set. We define a *world pair* as the pair (W, T) such that $W \subseteq T$. Consider the following partial order over world pairs: $(W_1, T_1) \leq_p (W_1, T_2)$ iff

$$(W_1 \subseteq W_2) \wedge ((T_1 \setminus W_1) \subseteq (T_2 \setminus W_2))$$

This partial order captures the intuition that the larger pair contains more information, with respect to both presence and absence of tuples.

The Smyth lifting of the partial order above yields the definition of superset-containment, while the Plotkin lifting yields the definition for equality-containment. The Plotkin order is stricter than the Smyth order, and similarly, we have:

$$(U_1 \sqsubseteq_E U_2) \Longrightarrow (U_1 \sqsubseteq_S U_2)$$

3. EXAMPLES

In this section, we use two examples to motivate our definitions of containment. We start with an abstract but simple example that illustrates the differences between the two notions of containment. Then, we present a practical example from a real-world application to illustrate the utility of our approach. Our example also demonstrates the notion of consistency, which is formally studied in Section 5.

EXAMPLE 2. *Recall Example 1 where uncertain databases over tuple set $T = \{A, B\}$ were:*

$$\begin{aligned} PW(U_1) &= \emptyset, \{A\}, \{B\}, \{A, B\} \\ PW(U_2) &= \{A\}, \{B\} \\ PW(U_3) &= \{A\} \end{aligned}$$

Now suppose there is a (logical or actual) mediated uncertain database M with $T(M) = \{A, B, C\}$ and:

$$PW(M) = \{A\}, \{A, C\}$$

Suppose M is an actual database, and a source S obtained from M doesn't have the privileges to access tuple C. Then S would be represented by U_3. Intuitively, we should have $U_3 \sqsubseteq_E M$ under equality-containment, and indeed this is the case according to Definition 2. Notice that $U_1 \not\sqsubseteq_E M$ and $U_2 \not\sqsubseteq_E M$, consistent with the fact that U_1 and U_2 cannot be obtained as a result of restricting access on M.

Now consider our other setting, where the sources were not derived from an actual uncertain database, and the job of integration is to logically construct an uncertain database. Specifically, we consider the sensor example from earlier, and whether our example

[1] See [20, 26] for examples.

uncertain databases would be consistent with logical construction of M. M states that there is an object in location A, no object in location B, and location C may or may not have an object. U_3 corresponds to a sensor that locates the object in A but may not have sufficient range to locate the object in C. U_2 corresponds to a less precise sensor, reporting the location to be either A or B. Although B is not in M, we still permit a source with a possible world containing B. Thus, we should have $U_3 \sqsubseteq_S M$ and $U_2 \sqsubseteq_S M$ under superset-containment, and indeed this is the case (as well as $U_1 \sqsubseteq_S M$) according to Definition 3.

EXAMPLE 3. *Suppose the FBI maintains a single relation* `Suspects(name,age,crime,...)` *containing information about suspects in the USA. Most of the information about crimes and suspects isn't certain, but is hypothesized based on evidence.*

Suppose the Southern California Police Department (SCPD) and Western California Police Department (WCPD) have access to just suspects in their region. Let this information be stored in relations `SCPD(name,age,crime,...)` *and* `WCPD(name,age,crime,...)`*, respectively. Further suppose*

PW(`SCPD`*)=*{`(Henry,...)`}, {`(George,...)`}

PW(`WCPD`*)=*{`(George,...)`,`(Kenny,...)`}, $\emptyset$

Suppose that the actual FBI database `Suspects` *is:*

PW(`Suspects`*)=*{`(Henry,...)`}, {`(George,...)`,`(Kenny,...)`} *Note that* `Suspects` *equality-contains both* `SCPD` *and* `WCPD`.

Now consider a third source, the San Francisco Police Department (`SFPD`*):*

PW(`SFPD`*)=*{`(Kenny,...)`}

The three sources `SCPD`*,* `WCPD`*, and* `SFPD` *are inconsistent under equality-containment; i.e., there can be no actual database that contains each of these sources. The inconsistencies arise because* `SFPD` *insists that Kenny is present in all possible worlds of FBI's* `Suspects` *relation, contrary to information in* `WCPD`.

Next consider the three uncertain relations `SCPD`*,* `WCPD`*, and* `SFPD` *under superset-containment. In this setting, instead of being derived from FBI's* `Suspects` *relation, these relations were obtained by collecting evidence locally. We now have the following mediated uncertain database U that superset-contains the three sources: $T(U)$ contains all three tuples* `(George,...)`*,* `(Kenny,...)`*, and* `(Henry,...)`*, while $PW(U)$ contains a single possible world with two tuples:* `(George,...)` *and* `(Kenny,...)`.

Intuitively, the three sources were resolved to conclude that George and Kenny were suspects while Henry was not: `SFPD` *insists that Kenny is a suspect, while* `WCPD` *says that either both Kenny and George are suspects, or neither is. Since Kenny is a suspect, we conclude that both Kenny and George are suspects. Finally, from* `SCPD` *we rule out Henry being a suspect, since* `SCPD` *says that exactly one of Henry and George is a suspect.*

Recalling the intuition from Section 1, notice that the `(Henry,...)` *possible world from* `SCPD` *and the $\emptyset$ possible world from* `WCPD` *are contradicted by a "corroboration" of all other possible worlds.*

4. QUERIES, VIEWS, AND SOURCES

Before proceeding we need a few definitions. Specifically, we define the semantics of monotonic queries over uncertain databases, the notion of views under equality-containment and superset-containment, and how we denote sources.

DEFINITION 4 (QUERIES OVER UNCERTAIN DATABASES). *The result of a monotonic query Q over an uncertain database U is an uncertain database $Q(U)$. The tuple set of $Q(U)$ is obtained by applying Q to the tuple set of U, and the possible worlds of $Q(U)$ are obtained by applying Q to each possible world of U:*

$$\begin{aligned} T(Q(U)) &= Q(T(U)) \\ PW(Q(U)) &= \{Q(W) \mid W \in PW(U)\} \end{aligned}$$

Notice that for monotonic queries, each possible world in $Q(U)$ is a subset of the tuple set of $Q(U)$, ensuring that $Q(U)$ is indeed an uncertain database.

The next definition specifies the semantics of LAV mappings by defining the notions of *view extension* and *view definition*. In this paper, these definitions are always used in conjunction with an implicit logical mediated database.

DEFINITION 5 (VIEW). *Consider an uncertain database V and a query Q. For a (logical) uncertain database M, V is a* view extension *under equality-containment (respectively superset-containment) with respect to* view definition *Q if and only if $V \sqsubseteq_E Q(M)$ (respectively $V \sqsubseteq_S Q(M)$).*

Next we formalize the notion of a source for LAV data integration in terms of views.

DEFINITION 6 (SOURCE). *A source $S = (V, Q)$ is specified by a view extension V and a view definition Q. V contains the data in the source while Q is the query used to map the source to the mediated schema. A set of sources $\mathcal{S} = \{S_1, \ldots, S_m\}$, where $S_i = (V_i, Q_i)$, is denoted as $\{\mathcal{V}, \mathcal{Q}\}$, where $\mathcal{V} = \{V_1, \ldots, V_m\}$, and $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$.*

5. CONSISTENCY

In this section, we formally define *consistency* of a set of uncertain data sources. We then present complexity results for the problem of consistency-checking under equality-containment and superset-containment.

Roughly speaking, a set of sources is consistent if there exists some mediated database that contains each source.

DEFINITION 7 (CONSISTENCY). *The set of sources $\mathcal{S} = \{S_1, \ldots, S_m\}$, where $S_i = (V_i, Q_i)$, is* consistent *if and only if there exists an uncertain database M such that:*

- $PW(M) \neq \emptyset$
- $\forall_{i \in \{1,\cdots,m\}} V_i \sqsubseteq Q_i(M)$ $\quad (\sqsubseteq$ *denotes* $\sqsubseteq_E$ *or* $\sqsubseteq_S)$

M is called a consistent mediated database for $\mathcal{S}$.

Under superset-containment, a set of sources can be "resolved" if and only if they are consistent. Similarly, under equality-containment, a set of sources is consistent if and only if there exists a mediated database from which they could have been derived.

In Section 5.1 we study *intensional complexity*: the complexity of consistency-checking in the size of the source data. In Section 5.2 we study *extensional complexity*: the complexity of consistency-checking in the size of the data assuming a fixed number of sources.[2] In this entire section, we restrict ourselves to identity views: Q_i is the identity query for every source S_i. In Section 7 we show that all of our results carry over to views defined by monotonic queries under some restrictions.

[2]Note that intensional complexity and extensional complexity correspond to the traditional notions of query and data complexity.

5.1 Intensional complexity

We now study the complexity of the consistency-checking problem in terms of the size of the data. This problem is interesting for applications that may integrate data from a large number of sources. For instance, web information extraction can involve combining information from a large number of webpages or websites, where extractors introduce uncertainty. We start by showing that in general consistency-checking is NP-hard. We then identify an interesting PTIME subclass by establishing a polynomial consistency check for that subclass.

Intractability

Theorem 1 below (proved in Appendix B) establishes the NP-hardness of consistency checking of a set of sources under both superset-containment and equality-containment. For both cases we show reductions from the well-known 3-coloring problem [17], although the arguments are slightly different. The reductions in the proof use one source for every node and edge, giving us NP-hardness in the size of source schemas. In Section 5.2 we will show that consistency-checking is tractable when the number of sources is fixed.

THEOREM 1. *Checking consistency of a set of sources under superset-containment and equality-containment is NP-hard.*

Tractable subclass

Next we show that for an interesting subclass, the intensional complexity of consistency-checking is PTIME. This subclass is based on a mapping from sets of uncertain databases to hypergraphs. First, we formally establish this mapping. We then show that if the set of uncertain data sources induces an *acyclic hypergraph*[3], then this set of data sources admits PTIME consistency-checking algorithms for both equality-containment and superset-containment.

Note that the notion of acyclic hypergraphs has been used extensively in database theory to identify polynomial subclasses for hard problems. See [7, 29, 34] for a few examples.

The nodes in the hypergraph represent tuples from uncertain databases, and each uncertain database is represented by a hyperedge in the hypergraph.

DEFINITION 8. *Consider a set of uncertain databases $\mathcal{U} = \{U_1, \ldots, U_m\}$. We construct the hypergraph $H = (N, E)$ as follows:*

$$N = \bigcup_i T(U_i), E = \{T(U_i) \mid i \in \{1, \ldots, m\}\}$$

The hypergraph H is said to be induced by $\mathcal{U}$.

We argue that practical uncertain databases often satisfy the acyclic hypergraph structure. Consider, for instance, our FBI data from Example 3 under the equality-containment setting. In addition to the zone- and city-level police departments, suppose we have state-level police departments: states subsuming zones and zones subsuming cities. The resulting uncertain database yields an acyclic hypergraph. Under the superset-containment setting, consider a series of sensors monitoring sets of rooms in a hallway; when the rooms are placed in an "acyclic fashion" (i.e., the hallway isn't a circle, but a set of chained rooms), the uncertain database representing sensor readings gives an acyclic hypergraph.

While we've shown practical scenarios where acyclicity arises in practice, we note that even when an uncertain database does not exhibit an acyclic hypergraph, we can impose acyclicity by "splitting" some sources. The consequence of splitting a source is that we may lose some specificity-information. (For example, U with $PW(U) = \{A\}, \{B\}, \{C\}$ may be split to get U_1 and U_2 such that $PW(U_1) = \{A\}, \{B\}, \emptyset$ and $PW(U_2) = \emptyset, \{C\}$.) Effectively, our results enable any set of uncertain databases to have tractable consistency-checking, but with some information loss when the acyclic hypergraph property isn't satisfied.

Our results are framed for the possible-worlds representation, but they also hold for more compact representations that satisfy conditions outlined in the respective theorems (such as the existence of a polynomial containment check). The following two theorems are the most technically challenging of the paper. Their proofs appear in the Appendix B (along with proofs for all subsequent theorems and lemmas).

THEOREM 2. *Consider a set of uncertain sources $\mathcal{S} = \{S_1, \ldots, S_m\}$ where each source is a described by the identity query, i.e., $S_i = (V_i, I)$. If the corresponding source extensions $\mathcal{V} = \{V_1, \ldots, V_m\}$ induce an acyclic hypergraph, checking consistency of the sources under equality-containment is PTIME for all representations that allow a PTIME containment check.*

THEOREM 3. *Consider a set of uncertain sources $\mathcal{S} = \{S_1, \ldots, S_m\}$ where each source is a described by the identity query, i.e., $S_i = (V_i, I)$. If the corresponding source extensions $\mathcal{V} = \{V_1, \ldots, V_m\}$ induce an acyclic hypergraph, checking consistency of the sources under superset-containment is PTIME for all representations that allow a PTIME containment operation.*

5.2 Extensional complexity

We now turn to extensional complexity of consistency-checking. We are interested in studying the complexity of consistency-checking in terms of the total data size when the number of sources is fixed. The following theorems give the good news that for both superset-containment and equality-containment, consistency-checking is PTIME in the size of the data. The constructive proofs of the theorems also indicate the consistency-checking algorithms that achieve PTIME complexity. Once again, our results are for the possible-worlds representation of uncertain data.

THEOREM 4. *Checking consistency of m (a constant) number of sources is PTIME in their total data size, under superset-containment.*

THEOREM 5. *Checking consistency of m (a constant) number of sources is PTIME in their total data size, under equality-containment.*

6. QUERY ANSWERS

In this section, we address the problem of defining correct answers for queries posed over the mediated schema in our uncertain LAV data integration settings. Since the sources are themselves uncertain, the answer is typically an uncertain database as well. There are multiple consistent mediated uncertain databases for a given input, hence the challenge is in defining the notion of "best" query answers corresponding to *certain answers* in certain data integration [22]. Informally, we would like to the best answer to contain all the information implied by the sources, and nothing more.

Note that query answering only makes sense when the input sources are consistent. Also, we restrict ourselves to identity queries; extensions to the class of monotonic views and queries follows using additional results presented in Section 7.

[3] See Appendix A for definition reproduced from [29, 34].

6.1 Definitions

We define the notion of *correct answer* and *strongest correct answer*, analogous to the traditional notions of *certain answer* and *maximal certain answer* for data integration without uncertainty [22].

DEFINITION 9 (CORRECT ANSWER). *Given a set of sources $\mathcal{S}$, an uncertain database A is a correct answer to a query Q if it is contained in the answers over all consistent mediated databases: $\forall_{M \in \mathcal{M}_c} A \sqsubseteq Q(M)$, where $\mathcal{M}_c$ is the set of all consistent mediated databases for $\mathcal{S}$.*

DEFINITION 10 (STRONGEST CORRECT ANSWER (SCA)). *A correct answer C is the strongest correct answer to a query Q if it contains all correct answers to the query: $\forall_{A \in \mathcal{A}_C} A \sqsubseteq C$, where $\mathcal{A}_C$ is the set of all correct answers to query Q.*

Under the superset-containment setting, we show by construction the existence of a unique SCA. However, under equality-containment, a nontrivial SCA may not always exist. Hence we introduce a weaker requirement than SCA, and construct the unique answer that satisfies the new requirement. For the results in this section, we need the following definitions.

DEFINITION 11 (FICTITIOUS TUPLE). *For a set of identity views $\{V_i\}$, a tuple t is said to be* fictitious *in a consistent mediated database M if t is not present in any of the view extensions; i.e., $\forall i, t \notin V_i$.*

DEFINITION 12 (COLLECTED DATABASE). *For a set of consistent sources, consider the set of mediated databases $\mathcal{M}_{res} = \{M \mid M \in \mathcal{M}_C, T(M) = \cup_i T(V_i)\}$, where $\mathcal{M}_c$ is the set of all consistent mediated databases. The* collected database *M_C has tuple set $T(M_C) = \cup_i T(V_i)$ and contains all possible worlds in all mediated databases in $\mathcal{M}_{res}$:*

$$PW(M_C) = \bigcup_{M \in \mathcal{M}_{res}} PW(M).$$

Notice that $\mathcal{M}_{res}$ is the set of mediated databases that do not contain any fictitious tuples.

6.2 Superset-Containment

The following theorem shows how to obtain the SCA to a query for a set of sources.

THEOREM 6. *For a set of consistent sources, where each source is described by the identity view, there exists an uncertain database M_I that gives the SCA C_Q to any query Q:*

$$\exists_{M_I} \forall_Q : C_Q = Q(M_I)$$

In fact, in the proof we see that the collected database produces the SCA to all queries.

6.3 Equality-Containment

This section studies query answering under the equality-containment setting. Unfortunately, a nontrivial SCA may not always exist for a set of views. However, the construction from the previous section still gives us good answers to queries in this setting: we show that it yields a unique procedure that satisfies certain natural properties.

THEOREM 7. *There exist sets of view extensions for which even though there are several nontrivial mediated databases, there is no SCA for the identity query.*

Since an SCA may not always exist, we relax our requirements from the best answer. We introduce the notion of a "query answering mechanism", and we define two desirable properties. We prove that the query answering mechanism we propose is the only mechanism that satisfies these properties.

DEFINITION 13 (QA MECHANISM AND PROCEDURE). *A* query answering mechanism *P, when initialized with a set of sources $\mathcal{S}$, yields a* query answering procedure *$P_{\mathcal{S}}$. Procedure $P_{\mathcal{S}}$ produces an uncertain database $A = P_{\mathcal{S}}(Q)$ as the result of a query Q. We say that mechanisms P^1 and P^2 are distinct if $\exists Q, \mathcal{S}$ such that $P^1_{\mathcal{S}}(Q) \neq P^2_{\mathcal{S}}(Q)$.*

The *consistency property* requires that the results for a query be obtained from a consistent mediated database. It also asserts that the query answering mechanism must not add data-information to the result beyond what is entailed by the sources, hence disallowing fictitious tuples (Definition 11).

PROPERTY 1 (CONSISTENCY PROPERTY). *A mechanism P satisfies the* consistency *property if and only if, for all initializations $\mathcal{S}$, P yields a procedure $P_{\mathcal{S}}$ that answers queries using a consistent mediated database M for $\mathcal{S}$ with no fictitious tuples.*

The *all-possibility property* requires that the query answering mechanism must not add specificity-information to the result beyond what is entailed by the sources. It asserts that the mechanism must answers queries without ruling out any possible world that could exist in some consistent mediated database.

PROPERTY 2 (ALL-POSSIBILITY PROPERTY). *A mechanism P satisfies the all-possibility property if and only if, for all initializations $\mathcal{S}$, P yields a procedure $P_{\mathcal{S}}$ satisfying the following: For any possible world W in any consistent mediated database, $Q(W) \in PW(P_{\mathcal{S}}(Q))$.*

The following fairly straightforward theorem states that answering queries using the collected database (Definition 12) is the only procedure that satisfies the two properties above.

THEOREM 8. *A query answering mechanism P^{cd} that answers queries using the collected database under equality-containment for any set of sources $\mathcal{S}$ is the only mechanism satisfying the consistency and all-possibility properties.*

7. MONOTONIC QUERIES

We now extend our results to view definitions and queries over the mediated database that are *monotonic queries*, i.e., composed of select, project, join, and union. (A straightforward extension to containment for multi-relation schemas is necessary.) In the extensions presented here, we require that the monotonic queries do not project away the keys of relations. Intuitively, if a query projects away the keys, tuples may no longer be equated correctly.

The consequence of this requirement is, essentially, that an entity resolution (or reference reconciliation) algorithm be used as a first step of data integration, to identify records that provide information about the same entity. Entity resolution is well accepted as a part of the data integration process even in the case of certain data. In fact, imperfect entity resolution may be one of the sources of uncertainty in the sources.

For a set of sources whose view definitions are monotonic queries, the intractability of consistency-checking in the size of the source schemas follows directly from the corresponding results on identity views. Consider extensional complexity and the polynomial subclass for intensional complexity. Using inverse rules [13],

we transform a source defined by a monotonic query to an *inverted skolemized source* and an *inverted deskolemized source*, both defined by the identity query. We then show that these polynomial transformations preserve consistency, hence our tractability results generalize to monotonic views for the possible worlds representation.

DEFINITION 14 (INVERTED SKOLEMIZED SOURCE). *For a source with view extension V and view definition Q, let the inverse rules be R. Consider the source (V^S,I) over the mediated schema, where V^S is obtained by applying the inverse rules to each possible world of the source extension.*

$$PW(V^S) = \{R(W) \mid W \in PW(V)\}$$

V^S is called the inverted skolemized version of the source.

DEFINITION 15 (INVERTED DESKOLEMIZED SOURCE). *When all tuples from the inverted skolemized version (V^S) of a source that contain at least one skolem constant are dropped, the uncertain database obtained (V^D) is called the inverted deskolemized version of the source.*

The following theorems show that the inverted deskolemized source and the inverted skolemized source are consistency preserving.

THEOREM 9. *A set of inverted deskolemized sources is consistent if and only if the corresponding set of skolemized sources is consistent.*

THEOREM 10. *A set of sources with extensions $\mathcal{V} = \{V_1, \ldots, V_m\}$ and view definitions $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ are consistent if and only if the inverted skolemized versions of the sources $\mathcal{V}^{\mathcal{S}} = \{V_1^S, \ldots, V_m^S\}$ (all defined by the identity query) are consistent:*

$$\{\mathcal{V}, \mathcal{Q}\} \textit{ is consistent } \equiv \{\mathcal{V}^{\mathcal{S}}, \mathcal{I}\} \textit{ is consistent}$$

THEOREM 11. *A set of sources with extensions $\mathcal{V} = \{V_1, \ldots, V_m\}$ and view definitions $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ is consistent if and only if the set of inverted deskolemized versions of the sources $\mathcal{V}^{\mathcal{D}} = \{V_1^D, \ldots, V_m^D\}$ (all defined by the identity query) is consistent:*

$$\{\mathcal{V}, \mathcal{Q}\} \textit{ is consistent } \equiv \{\mathcal{V}^{\mathcal{D}}, \mathcal{I}\} \textit{ is consistent}$$

The above theorems together show that all of our PTIME results (for the tractable query-complexity subclass as well as the extensional complexity results) for both superset-containment and equality-containment carry over for monotonic views: the consistency checks are now applied on the inverted deskolemized sources.

Next we turn to answering monotonic queries over a a set of sources with monotonic queries as views definitions. For monotonic views, we use their inverted skolemized versions to construct the set of mediated databases. Note that, this allows non-fictitious tuples to have skolem constants. The construction of the collected database from Section 6 uses the above set of mediated databases. The result of query over a skolemized relation retains only tuples with no skolem constants.

The query-answering results for equality-containment presented in Section 6.3 carry over from the above observations. However, as described in Section 6.2, extending our superset-containment results to arbitrary monotonic queries additionally requires us to show that containment is preserved by our class of queries:

LEMMA 1. *For uncertain databases U_1, U_2 with the same schema $\mathcal{SC}$, for any monotonic query Q over $\mathcal{SC}$ that retains a key $\mathcal{K}$ of $\mathcal{SC}$, we have: $U_1 \sqsubseteq_S U_2 \implies Q(U_1) \sqsubseteq_S Q(U_2)$.*

8. RELATED WORK

This paper introduces a theory for LAV integration of uncertain data. Several decades of work have been done on data integration (refer to [22, 23] for surveys) as well as on uncertain data (a small subset of which can be found in [1, 2, 4, 5, 6, 8, 16, 19, 24, 25, 32, 35]), and we do not review this past work here. There has been little work on integrating uncertain data.

There has been a lot of work in the area of incomplete information databases. These are sets of certain global databases that arise as a result of data integration of certain sources. Reference [18] presents a good overview. In contrast, in our setting integration of uncertain sources results in sets of uncertain global databases.

Our theory is based on possible worlds and some of our results rely on the existence of an efficient containment check over the model used for representing uncertain databases. In contrast, reference [2] presents complexity results about representing and asking questions about sets of possible worlds. This work is in fact complimentary to our work, and provides a natural starting point for our investigation about compact representations.

There has been a flurry of recent work on using probabilistic techniques for data integration [15, 27, 28, 30]. This work looks at uncertain integration of certain data and is not to be confused with our work, which addresses integration of uncertain data itself.

Recently, data exchange has been studied over probabilistic databases [14]. In contrast to our work, which combines information from multiple sources to a single target, the work in [14] only considers a single source. However in the context of a single source: (1) it allows more general kinds of mappings than just local-as-view mappings; (2) has probabilities associated with possible worlds; and (3) it studies some compact representations.

Reference [11] studies the problem of answering queries using imprecise sources, where the imprecision is captured by a probabilistic database. The paper presents conditions under which view statistics are sufficient to answer a query over a mediated database and describes algorithms for computing result-tuple probabilities. In contrast, the goal of our work is to develop a theory for integrating uncertain sources starting with the fundamental notion of containment. To this end, we introduce superset-containment and equality-containment, and address the problem of consistency, none of which are the subject of [11].

Finally, several papers [12, 21, 23, 31] mention that the problem of integrating uncertain data is important, but do not address it.

9. FUTURE WORK

We laid the foundation for uncertain-data integration by introducing the notions of superset-containment and equality-containment, formalizing consistency-checking, and studying the notion of query answers. Our work suggests several interesting directions for future work:

- **Confidence values.** This paper considered non-probabilistic uncertain data, as we encountered sufficient challenges with this case itself. We are currently investigating how our definitions and results extend to the case of uncertain data with confidence values or probabilities. Our approach relies on theories of evidence and belief [33], along with generalizations of set-containment.

- **Inconsistent sources.** We defined the notion of consistency of sources in this paper, and studied the complexity of consistency-checking. An interesting direction of future research is to devise techniques to deal with inconsistent sources, possibly providing best-effort answers.

- **Efficient representations.** Since uncertain databases are traditionally defined through sets of possible worlds, we developed the theory in this paper based on possible worlds. It would be interesting to consider the implications of our results on compact models for representing uncertain data, such as those in [2, 4, 5, 8, 10, 19, 24, 32]. We are also exploring new models that might be less expressive, but permit efficient consistency checks and query answering.

- **Query processing.** In this paper, we defined the notion of strongest correct answers, but did not provide a way to compute them. An important direction of future work is to develop efficient query processing techniques over compact representations.

- **Applications.** We are currently applying the techniques developed in this paper to real-world data integration settings, where uncertainty arises as a result of entity resolution, probabilistic mappings, and noisy data. This investigation currently is focused on modeling of application data as uncertain sources and quality of answers based on our techniques. Efficiency is a next step.

- **Alternative approach.** An alternative to our approach for uncertain data integration could be to perform probabilistic data exchange [14] followed by imposing some flavor of functional or equality generating dependencies for uncertain data. The hope is that this approach is in fact equivalent to the current approach. While the two approaches provide similar utility for integration, our idea is especially appealing because it can potentially allow us to perform entity resolution (recall Section 7) after the data from all sources has been migrated to the mediated schema.

10. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] S. Abiteboul, P. Kanellakis, and G. Grahne. On the representation and querying of sets of possible worlds. *SIGMOD Record*, 16(3), 1987.

[3] Samson Abramsky and Achim Jung. Domain theory. In *Handbook of Logic in Computer Science*, pages 1–168. Clarendon Press, 1994.

[4] L. Antova, C. Koch, and D. Olteanu. MayBMS: Managing Incomplete Information with Probabilistic World-Set Decompositions. In *Proc. of ICDE*, 2007.

[5] O. Benjelloun, A. Das Sarma, A. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In *Proc. of VLDB*, 2006.

[6] J. Boulos, N. Dalvi, B. Mandhani, S. Mathur, C. Re, and D. Suciu. MYSTIQ: a system for finding more answers by using probabilities. In *Proc. of ACM SIGMOD*, 2005.

[7] C. Chekuri and A. Rajaraman. Conjunctive query containment revisited. In *Proc. of ICDT*, 1997.

[8] R. Cheng, S. Singh, and S. Prabhakar. U-DBMS: A database system for managing constantly-evolving data. In *Proc. of VLDB*, 2005.

[9] Thomas M. Cover and Joy Thomas. *Elements of Information Theory*. Wiley, 1991.

[10] N. Dalvi and D. Suciu. Efficient Query Evaluation on Probabilistic Databases. In *Proc. of VLDB*, 2004.

[11] N. Dalvi and D. Suciu. Answering queries from statistics and probabilistic views. In *Proc. of VLDB*, 2005.

[12] X. Dong, A. Y. Halevy, and C. Yu. Data integration with uncertainty. In *Proc. of VLDB*, 2007.

[13] O. M. Duschka. *Query planning and optimization in information integration*. PhD thesis, 1998.

[14] R. Fagin, B. Kimelfeld, and P. G. Kolaitis. Probabilistic data exchange. In *Proc. of ICDT*, 2010.

[15] D. Florescu, D. Koller, and Alon Y. Levy. Using probabilistic information in data integration. In *Proc. of VLDB*, 1997.

[16] N. Fuhr and T. Rölleke. A Probabilistic NF2 Relational Algebra for Imprecision in Databases. *Unpublished Manuscript*, 1997.

[17] M. R. Garey and D. S. Johnson. Computers and Intractability. *W. H. Freeman and Company*, 1979.

[18] G. Grahne. Information integration and incomplete information. *IEEE Data Engineering Bulletin*, 25(3), 2002.

[19] T. J. Green and V. Tannen. Models for incomplete and probabilistic information. In *Proc. of IIDB Workshop*, 2006.

[20] Carl A. Gunter. *Semantics of programming languages: structures and techniques*. MIT Press, 1992.

[21] L. Haas. Beauty and the beast: The theory and practice of information integration. In *ICDT*, 2007.

[22] A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 4, 2001.

[23] A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *VLDB*, 2006.

[24] T. Imielinski and W. Lipski. Incomplete Information in Relational Databases. *Journal of the ACM*, 31(4), 1984.

[25] L. V. S. Lakshmanan, N. Leone, R. Ross, and V.S. Subrahmanian. ProbView: A Flexible Probabilistic Database System. *ACM TODS*, 22(3), 1997.

[26] Leonid Libkin and Limsoon Wong. On representation and querying incomplete information in databases with bags. *Inf. Process. Lett.*, 1995.

[27] M. Magnani and D. Montesi. Uncertainty in data integration: current approaches and open problems. In *VLDB workshop on Management of Uncertain Data*, pages 18–32, 2007.

[28] M. Magnani, N. Rizopoulos, P. Brien, and D. Montesi. Schema integration based on uncertain semantic mappings. *Lecture Notes in Computer Science*, pages 31–46, 2005.

[29] David Maier. *The Theory of Relational Databases*. Computer Science Press, 1983.

[30] A. Das Sarma, L. Dong, and A. Halevy. Bootstrapping pay-as-you-go data integration systems. In *Proc. of ACM SIGMOD*, 2008.

[31] A. Das Sarma, L. Dong, and A. Halevy. Uncertainty in data integration. *C. Aggarwal, editor, Managing and Mining Uncertain Data*, 2009.

[32] P. Sen and A. Deshpande. Representing and Querying Correlated Tuples in Probabilistic Databases. In *Proc. of ICDE*, 2007.

[33] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[34] J. D. Ullman. *Principles of Database and Knowledge-Base Systems, Volume I*. Computer Science Press, 1988.

[35] J. Widom. Trio: A System for Integrated Management of Data, Accuracy, and Lineage. In *Proc. of CIDR*, 2005.

APPENDIX

A. DEFINITIONS

We reproduce definitions of *GYO-reductions* and *acyclic hypergraphs* from [29, 34].

DEFINITION 16 (GYO-REDUCTION). *The GYO-reduction repeatedly applies the following two rules to the hypergraph $H = (N, E)$ until none can be applied further:*

- **Node Removal** *If a node t is contained in at most one hyperedge e in H, remove t from e, and from N.*
- **Edge Removal** *If a hyperedge e is is contained in another hyperedge f, remove e from E.*

DEFINITION 17 (ACYCLIC HYPERGRAPHS). *A hypergraph is* acyclic *if its GYO-reduction results in a simple (empty) hyperedge.*

The following preliminary definitions are used to define shorthand notations used in proofs.

DEFINITION 18. *$U \downarrow S$ denotes the result of removing a set S of tuples from an uncertain database U:*

$$PW(U \downarrow S) \equiv_{defn} \{W \setminus S \mid W \in PW(U)\}$$
$$T(U \downarrow S) \equiv_{defn} T(U) \setminus S$$

DEFINITION 19 (RESTRICTION). *$U_1 \Downarrow U_2$ denotes the result of* restricting *an uncertain database U_1 to an uncertain database U_2:*

$$PW(U_1 \Downarrow U_2) \equiv_{defn} \{W \cap T(U_2) \mid W \in PW(U_1)\}$$
$$T(U_1 \Downarrow U_2) \equiv_{defn} T(U_1) \cap T(U_2)$$

The above definitions satisfy the following properties, which we will use later in proofs:

$$(U_1 \sqsubseteq_E U_2) \equiv (U_1 = U_2 \Downarrow U_1)$$
$$(U_1 \sqsubseteq_S U_2) \equiv (PW(U_1) \supseteq PW(U_2 \Downarrow U_1)) \wedge (T(U_1) = T(U_2 \Downarrow U_1))$$
$$(U_1 \Downarrow U_2) \equiv (U_2 \downarrow (T(U_2) \setminus T(U_1)))$$
$$(PW(U_1) \neq \emptyset) \Longrightarrow \forall_{U_2} PW(U_1 \Downarrow U_2) \neq \emptyset$$
$$(PW(U_1) \neq \emptyset) \Longrightarrow \forall_S PW(U_1 \downarrow S) \neq \emptyset$$

The last two statements indicate that the restriction and removal operations cannot make the set of possible worlds empty, although individual possible worlds may become empty.

B. PROOFS

Proof of Theorem 1: *Reduction from 3-coloring*
Let the mediated schema be a single table with only one column. For every vertex v, we use 3 symbols v_0, v_1, v_2 corresponding to its 3 colorings. Given a graph, we construct the following views (each described by the identity query):

- For every vertex v, construct a view extension V_v with 3 possible worlds representing its 3 colorings: $PW(V_v) = \{\{v_0\}, \{v_1\}, \{v_2\}\}$.
- For every edge (u, v), construct a view V_{uv} with 6 possible worlds representing the 6 allowed colorings of the nodes u, v: $PW(V_{uv}) = \{\{u_0, v_1\}, \{u_1, v_0\}, \{u_1, v_2\}, \{u_2, v_1\}, \{u_2, v_0\}, \{u_0, v_2\}\}$

Consistent $\Longrightarrow$ 3-coloring: Let W be a possible instance of a mediated database M. The following shows that it represents a 3-coloring:

- Every vertex v is assigned exactly one color: $W \cap \{v_0, v_1, v_2\} \in PW(V_v)$
- For every edge (u, v), u and v are assigned different colors: $W \cap \{u_0, u_1, u_2, v_0, v_1, v_2\} \in PW(V_{uv})$

3-coloring $\Longrightarrow$ Consistent (superset-containment): A 3-coloring can be represented as a possible instance W with a symbol for each vertex chosen according to the color assigned to it. Consider the uncertain database M, such that $PW(M) = \{W\}$. M is a consistent mediated database under superset-containment.

3-coloring $\Longrightarrow$ Consistent (equality-containment): There are 6 permutations of the 3 colors, hence if a valid 3-coloring exists, 6 valid 3-colorings exist each derived from one of the 6 permutations of the 3 colors. Each 3-coloring can be represented as a possible instance with a symbol for each vertex chosen according to the color assigned to it. Consider the uncertain database M, such that $PW(M) = \{W_1, \ldots, W_6\}$. M is a consistent mediated database under equality-containment. □

Proof of Theorem 2: We "reduce" (using polynomial-time) the set of uncertain databases along with the corresponding GYO-reduction on the induced hypergraph. Let the uncertain database corresponding to an edge e be denoted by $V(e)$.

- **Node Removal:** We remove the tuple corresponding to the node t from $V(e)$, as the node t is removed. I.e., we replace $V(e)$ by $V(e) \downarrow \{t\}$.
- **Edge Removal:** If $V(e) \sqsubseteq_E V(f)$, remove the source corresponding to $V(e)$, along with the the edge e. I.e., we replace $\mathcal{V}$ by $\mathcal{V} \setminus \{V(e)\}$.

A set of sources inducing an acyclic hypergraph is consistent if and only if the above reduction results in a hypergraph with just one hyperedge. The lemmas below complete the proof.

LEMMA 2. *Node removal preserves consistency.*

PROOF. Let the node removal step remove tuple t from the source V_i. Recall that t is not contained in any other source. Let $\mathcal{V}_I = \{V_1, \ldots, V_i, \ldots, V_m\}$ denote the view extensions before the node removal, and let $\mathcal{V}_F = \{V_1, \ldots, V_i \downarrow \{t\}, \ldots, V_m\}$ denote the view extensions after the node removal.

$\mathcal{V}_I$ is consistent $\Longrightarrow$ $\mathcal{V}_F$ is consistent: Let M_I be a consistent mediated database corresponding to $\mathcal{V}_I$. $M_F = M_I \downarrow \{t\}$ is a consistent mediated database for $\mathcal{V}_F$.

$\mathcal{V}_F$ is consistent $\Longrightarrow$ $\mathcal{V}_I$ is consistent: Let M_F be a consistent mediated database corresponding to $\mathcal{V}_F$. We construct M_I with $T(M_I) = T(M_F) \cup \{t\}$ with $PW(M_I)$ given by :

$$\begin{aligned} \{V \mid V &= W \cup \{t\}, \\ &\text{if } (W \cap T(V_i \downarrow \{t\}) \cup \{t\} \in PW(V_i) \\ V &= W, \text{ otherwise}\} \end{aligned}$$

In the above equation, W iterates over possible worlds of V_i. M_I is a consistent mediated database for $\mathcal{V}_I$. □

LEMMA 3. *Edge removal preserves consistency.*

PROOF. Consider the edge-removal step on edge $e \subseteq f$. Let the set of view extensions be $\mathcal{V}_I$ and $\mathcal{V}_F = \mathcal{V}_I \setminus V(e)$ before and after the edge removal respectively.

$\mathcal{V}_I$ **is consistent** $\Longrightarrow$ $\mathcal{V}_F$ **is consistent:** A database M that is consistent for $\mathcal{V}_I$ is also consistent for $\mathcal{V}_F$.

$\mathcal{V}_F$ **is consistent** $\Longrightarrow$ $\mathcal{V}_I$ **is consistent:** A database M that is consistent for $\mathcal{V}_F$ is also consistent for $\mathcal{V}_I$: the following shows $V(e) \sqsubseteq_E M$.

$$\begin{aligned} M \Downarrow V(e) &= (M \Downarrow V(f)) \Downarrow V(e) \quad (\text{since } e \subseteq f) \\ &= V(f) \Downarrow V(e) \quad (\text{since } V(f) \sqsubseteq_E M) \\ &= V(e) \quad (\text{since } V(e) \sqsubseteq_E V(f)) \end{aligned}$$

□

LEMMA 4. *For consistent source extensions $\mathcal{V}$, $e \subseteq f \Longrightarrow V(e) \sqsubseteq_E V(f)$.*

PROOF. Let M be a consistent mediated database for $\mathcal{V}$. We show $V(e) \sqsubseteq_E V(f)$:

$$\begin{aligned} V(f) \Downarrow V(e) &= (M \Downarrow V(f)) \Downarrow V(e) \quad (\text{since } V(f) \sqsubseteq_E M) \\ &= M \Downarrow V(e) \quad (\text{since } e \subseteq f) \\ &= V(e) \quad (\text{since } V(e) \sqsubseteq_E M) \end{aligned}$$

□

□

Proof of Theorem 3: The node removal step of the GYO-reduction is the same as Theorem 2.

- **Edge Removal:** Remove the source corresponding to $V(e)$, along with the the edge e, and modify the uncertain database associated with f to $V_R(f)$ by retaining the same tuple set and making the possible worlds $PW(V_R(f))$ equal to:

  $$\{W \mid W \in PW(V(f)), W \cap T(V(e)) \in PW(V(e))\}$$

  We require the above operation to be polynomial in the size of the representation. Note by construction that $PW(V_R(f))$ is the largest subset of $PW(V(f)$ such that $V(e) \sqsubseteq_S V_R(f)$. We replace $\mathcal{V}$ by $(\mathcal{V} \setminus \{V(e), V(f)\}) \cup \{V_R(f)\}$.

The sources are consistent if and only if the above reduction results in a simple hyperedge. If during an edge removal step we obtain $V_R(f) = \emptyset$, we declare the sources inconsistent. Lemma 2 above continues to hold. Lemmas 5 and 6 complete the proof.

LEMMA 5. *Edge removal preserves consistency.*

PROOF. Consider the edge removal step on edge $e \subseteq f$. Let the set of view extensions be $\mathcal{V}_I$ and $\mathcal{V}_F = (\mathcal{V}_I \setminus \{V(e), V(f)\}) \cup \{V_R(f)\}$, before and after the removal.

$\mathcal{V}_I$ **is consistent** $\Longrightarrow$ $\mathcal{V}_F$ **is consistent:** A database M that is consistent for $\mathcal{V}_I$ is also consistent for $\mathcal{V}_F$. The following shows $V_R(f) \sqsubseteq_S M$:

$PW(M \Downarrow V_R(f)) \subseteq PW(M \Downarrow V(f))$
(since $T(V_R(f)) = T(V(f))$)
$\subseteq PW(V(f))$ (since $V(f) \sqsubseteq_S M$)

$PW((M \Downarrow V_R(f)) \Downarrow V(e)) = PW(M \Downarrow V(e))$
(since $T(V(e)) \subseteq T(V_R(f))$)
$\subseteq PW(V(e))$ (since $V(e) \sqsubseteq_S V_R(f)$)

$\mathcal{V}_F$ **is consistent** $\Longrightarrow$ $\mathcal{V}_I$ **is consistent:** A database M that is consistent for $\mathcal{V}_F$ is also consistent for $\mathcal{V}_I$: the following shows $V(f) \sqsubseteq_S M$, and $V(e) \sqsubseteq_S M$:

$PW(M \Downarrow V(f)) = PW(M \Downarrow V_R(f))$
(since $T(V_R(f)) = T(V(f))$)
$\subseteq PW(V(f))$ (since $V_R(f) \sqsubseteq_S M$

$PW(M \Downarrow V(e)) \subseteq PW((M \Downarrow V_R(f)) \Downarrow V(e))$
(since $T(V(e)) \subseteq T(V_R(f))$)
$\subseteq PW(V(e))$ (since $V(e) \sqsubseteq_S V_R(f)$)
□

LEMMA 6. *For a consistent set $\mathcal{V}$ of source extensions, $PW(V_R(f)) \neq \phi$.*

PROOF. Let M be a consistent mediated database for $\mathcal{V}$. By definition, $PW(M) \neq \emptyset$. Since consistency is preserved by an edge removal step, $V_R(f) \sqsubseteq_S M$, $PW(V_R(f)) \supseteq PW(M \Downarrow V_R(f))$. □

□

Proof of Theorem 4: Consider sources with extensions $\mathcal{V} = \{V_1, \ldots, V_m\}$, with $n_1, \ldots, n_m$ possible worlds respectively. We consider all $\mathcal{N} = \prod_{i \in \{1,\cdots,m\}} n_i$ ways of picking one possible world from each source. In one such instance, let W_i be the possible world picked from the source V_i. Consider the uncertain database with one possible world $W = \cup_{i \in \{1,\cdots,m\}} W_i$. The following lemma completes the proof.

LEMMA 7. *The set $\mathcal{V}$ of sources is consistent if and only if at least one of these $\mathcal{N}$ uncertain databases, say U, is a consistent mediated database.*

PROOF.
$\exists$ **Consistent** $U \Longrightarrow$ **Consistent** $\mathcal{V}$**:** U is a consistent mediated database for the set $\mathcal{V}$.

Consistent $\mathcal{V} \Longrightarrow \exists$ **Consistent** U**:** Consider a consistent mediated database M for $\mathcal{V}$ and let W_V be one of its possible worlds. For each source $V_i \sqsubseteq_S M$, hence $W_i = (W_V \cap T(V_i))$ is a possible world of V_i. Construct U as the uncertain database with one possible world $\cup_{i \in \{1,\cdots,m\}} W_i$. Note that U is consistent, and is one of the $\mathcal{N}$ uncertain databases above. □

□

Proof of Theorem 5: Consider sources with extensions $\mathcal{V} = \{V_1, \ldots, V_m\}$, with $n_1, \ldots, n_m$ possible worlds respectively. We consider all $\mathcal{N} = \prod_{i \in \{1,\cdots,m\}} n_i$ ways of picking one possible world from each source. In one such instance, let W_i be the possible world picked from the source V_i. Consider the uncertain database $U(W)$ with one possible world $W = \cup_{i \in \{1,\cdots,m\}} W_i$. Out of the $\mathcal{N}$ candidate W's, we construct an uncertain database M by adding all possible worlds W whose corresponding uncertain database $U(W)$ is consistent for $\mathcal{V}$. Note that $M \neq \emptyset$ iff $\mathcal{V}$ is consistent. The following lemma completes the proof.

LEMMA 8. *The set $\mathcal{V}$ of sources is consistent if and only if M is a consistent mediated database.*

PROOF.
Consistent $M \Longrightarrow$ **Consistent** $\mathcal{V}$**:** M is a consistent mediated database for $\mathcal{V}$.

Consistent $\mathcal{V} \Longrightarrow$ Consistent M: Consider a consistent mediated database M_V for $\mathcal{V}$ and let W_V be one of its possible worlds. For each source $V_i \sqsubseteq_S M$, hence $W_i = (W_V \cap T(V_i))$ is a possible world of V_i. Notice that $W = \cup_{i\in\{1,\cdots,m\}} W_i$ is a possible world in M, since $W \subseteq W_V$. For each source, W collapses to the same possible world as W_V, by construction of W. Hence M is a consistent mediated database for $\mathcal{V}$. □

□

Proof of Theorem 6: We show that answering queries using the collected database M_C (from Definition 12) gives the SCA for all queries.

$Q(M_C)$ **is a correct answer :** Consider any mediated database M' in $\mathcal{M}_C$. $M' \Downarrow M_C$ is equivalent to eliminating fictitious tuples from M'. Hence,

$$\exists_{M\in\mathcal{M}_{res}} PW(M' \Downarrow M_C) = PW(M)$$

Note by definition of the collected database:

$$\forall_{M\in\mathcal{M}_{res}} PW(M) \subseteq PW(M_C)$$

Hence, all consistent mediated databases are contained in M_C: $\forall_{M'\in\mathcal{M}_C} M_C \sqsubseteq_S M'$. Hence for identity queries, M_C gives a correct answer. Note that to extend our result to monotonic views and queries, we only need: $U_1 \sqsubseteq_S U_2 \Longrightarrow Q(U_1) \sqsubseteq_S Q(U_2)$. This result is provided by Lemma 1 from Section 7.

$Q(M_C)$ **contains all correct answers :** The collected database is a consistent mediated database; i.e., $M_C \in \mathcal{M}_{res} \subseteq \mathcal{M}_C$. By definition of correct answers, every correct answer A to the identity query is contained in the collected database: A is correct $\Longrightarrow$ $A \sqsubseteq_S Q(M_C)$. □

Proof of Theorem 7: Consider two views: V_1 with possible worlds $W_{11} = \{a\}$ and $W_{12} = \emptyset$, and V_2 with possible worlds $W_{21} = \{b\}$ and $W_{22} = \emptyset$. The two views give several consistent mediated databases, such as $M_1 = \{\{a, b\}, \{a\}, \{b\}, \emptyset\}$ and $M_2 = \{\{a, b\}, \emptyset\}$. While V_1 and V_2 themselves are correct answers, any uncertain database A with $T(A) = \{a, b\}$ is not contained in at least one of the above mediated databases. Hence, there is no SCA for the identity query. □

Proof of Theorem 8: We prove the theorem in three parts.

P^{cd} **satisfies the Consistency Property:** Recall the construction of the collected mediated database M_C. Since all possible worlds of at least one consistent mediated database are possible worlds of M_C, M_C is also consistent under equality-containment.

P^{cd} **satisfies the All-Possibility Property:** Follows directly from the construction of the collected mediated database.

Uniqueness: Consider any other mechanism $P' \neq P^{cd}$. There must exist $\mathcal{S}, Q$ such that $P'_{\mathcal{S}}(Q) \neq P^{cd}_{\mathcal{S}}(Q)$. Therefore, by the consistency property, P' uses some other mediated database M, such that $PW(M) \subset PW(M_C)$. Hence, $W \in PW(M_C) - PW(M)$, violating the all-possibility property. □

Proof of Theorem 9: Consider a tuple t that contains a skolem constant. Since skolem constants are unique to the rule applied, the tuple t can only exist in one deskolemized source. Hence, the tuple t can be dropped using the node-removal step of Theorem 2, which preserves consistency. □

Proof of Theorem 10: $\{\mathcal{V}, \mathcal{Q}\}$ **is consistent** $\Longrightarrow$ $\{\mathcal{V}^{\mathcal{S}}, \mathcal{I}\}$ **is consistent:** Let M be a consistent mediated database for $\{\mathcal{V}, \mathcal{Q}\}$. M is also a consistent mediated database for $\{\mathcal{V}^{\mathcal{D}}, \mathcal{I}\}$. Using this observation with Theorem 9 shows that $\{\mathcal{V}^{\mathcal{S}}, \mathcal{I}\}$ is indeed consistent.

$\{\mathcal{V}^{\mathcal{S}}, \mathcal{I}\}$ **is consistent** $\Longrightarrow$ $\{\mathcal{V}, \mathcal{Q}\}$ **is consistent:** Let M be a consistent mediated database for $\{\mathcal{V}^{\mathcal{S}}, \mathcal{I}\}$. We show that M is a consistent mediated database for $\{\mathcal{V}, \mathcal{Q}\}$. For each view V_i:

$$\begin{aligned} V_i &= Q_i(V_i^S) \quad \text{(by construction)} \\ &\sqsubseteq Q_i(M) \quad (\text{since } V_i^S \sqsubseteq M, \text{Lemma 1}) \end{aligned}$$

□

Proof of Theorem 11: The result follows directly from Theorems 9 and 10. □

Proof of Lemma 1: By definition of $U_1 \sqsubseteq_S U_2$:

$$\forall_{W_2\in PW(U_2)} \exists_{W_1\in PW(U_1)} W_2 \cap T(U_1) = W_1$$

For any $W_2 \in PW(U_2)$, consider any tuple $t \in W_2 \setminus T(U_1)$. (If no such t exists, clearly, $Q(W_2) = Q(W_1)$.) For key-preserving monotonic queries, $t.\mathcal{K} \notin T(Q(W_1).\mathcal{K})$, hence $Q(W_2) = Q(W_1)$. Hence

$$\forall_{QW_2\in PW(Q(U_2))} \exists_{QW_1\in PW(Q(U_1))} QW_2 \cap T(Q(U_1)) = QW_1$$

The above completes the proof. □