Universität Hamburg
Department Informatik
Vogt-Kölln-Str. 30
D-22527 Hamburg

Bericht 292

Towards Duplicate Detection and Data Fusion in Fuzzy Relational Databases

FBI-HH-B-292/10

Fabian Panse
Norbert Ritter
Universität Hamburg
Department Informatik
{panse, ritter@informatik.
uni-hamburg.de}

In die Reihe der Berichte des Fachbereichs Informatik aufgenommen durch
Prof. Dr. N. Ritter
Prof. Dr. C. Habel

April 2010

Neueste Berichte Recent Reports

B-293 M. Kudlek, P. Totzke, G. Zetzsche
Are there Universal Finite or Pushdown Automata? 2010

B-292 F. Panse, N. Ritter
Towards Duplicate Detection and Data Fusion in Fuzzy Relational Databases. 2010

B-291 A. Solth, B. Neumann, P. Stelldinger
Strichextraktion und – analyse Handschriftlicher chinesischer Schriftzeichen. 2009

B-290 M. Duvigneau, D. Moldt
Proceedings of the Fifth International Workshop on Modelling of Objects Components and Agents MOCA´09. 2009

B-289 G. Zetzsche
A Note on Hack´s Conjecture, Parikh Images of Matrix Languages and Multiset Grammars. 2009

B-288 K. Terzic, B. Neumann
Decision trees for probabilistic top-down and bottom-up integration. 2009

B-287 B. E. Wolfinger, K.-D. Heidtmann
Leistungs-, Zuverlässigkeits- und Verlässlichkeitsbewertung von Kommunikationsnetzen und verteilten Systemen : 5. GI/ITG Workshop. 2009

B-286 B. Beyene, M. Kudlek
Calendars in Ethiopia. 2009

B-285 R. A. Diaconu, M. Kudlek
Some Remarks on Multi-prime RSA. 2008

B-284 W. Menzel, K. Dalinghaus
An Implementation of the Argument Dependency Model. 2008

Das Department Informatik der Universität Hamburg veröffentlicht wichtige Ergebnisse seiner Arbeit in zwei Reihen, den Mitteilungen und den Berichten. Mitteilungen sind für die schnelle Verbreitung von aktuellen Forschungsergebnissen vorgesehen, Berichte dienen der Publikation von länger gültigen gewichtigeren Ergebnissen. Herausgeber beider Reihen ist das Department Informatik.

Die Aufnahme einer Arbeit in die Berichts-Serie liegt in der Verantwortung jeweils der beiden auf dem Titelblatt genannten gutachtenden Professoren des Departments. Bei den Mitteilungen liegt die Verantwortung beim Leiter des Arbeitsbereichs.

Listen der Berichte und Mitteilungen sowie einzelne Exemplare können Sie bei der unten angegebenen Adresse bestellen.

Department of Informatics (MIN-Faculty of Hamburg University) publishes important results of its work in two forms: Memos and Reports. Memos serve to disseminate current research results quickly, while Reports are intended to cover major results of long-term interest in depth. The publisher of both is the Department of Informatics.

The acceptance of a publication as a Report is based on the recommendation of two departmental professors whose names appear on the title page. Responsible for the publication of Memos is the head of the respective research group.

A list of all available titles as well as copies of publications may be obtained from:

Bibliothek des Departments Informatik
Vogt-Kölln-Str. 30
D-22527 Hamburg
Telefon +49 40 428832216
Telefax +49 40 428832217
infbib@informatik.uni-hamburg.de

Zusammenfassung

Die Durchführung von Methoden zur Duplikaterkennung und Datenfusion sind zwei wesentliche Schritte des Datenintegrationsprozesses um konsistente Ergebnisse zu gewährleisten. Aufgrund von Fehlern und Ungenauigkeiten während der Datenerhebung, der Datenmodellierung oder der Datenverwaltung sind Daten in praktischen Anwendungsbereichen oft inkorrekt und/oder unvollständig. Dies wiederum erschwert die Identifizierung und Zusammenführung mehrfacher Darstellungen des gleichen Realweltobjektes. Im momentan vorherrschenden relationalen Datenmodell lassen sich unvollständige Informationen nur durch einen Nullwert abbilden. Demzufolge fokussieren aktuelle Techniken der Duplikaterkennung und der Datenfusion zumeist auch nur auf die Behandlung widersprüchlicher Informationen, welche aus Tippfehlern, veralteten Daten oder falschen Schreibweisen resultieren. Für gewöhnlich sind Informationen über Phänomene der realen Welt jedoch selten vollständig, sondern eher ungewiss, unpräzise und vage. Aus diesem Grund wurden verschiedene Datenmodelle zur Handhabung ungenauer und unvollständiger Informationen entwickelt. Ein beträchtlicher Anteil dieser Modelle basiert auf der Wahrscheinlichkeitstheorie oder der Fuzzy-Set-Theorie. Aktuelle Techniken zum Abgleich und Zusammenfügen von Datensätzen sind allerdings nicht für den Umgang mit solchen Theorien konzipiert. Um dennoch eine Integration von verschiedenen Fuzzy-Datenbanken zu ermöglichen, präsentieren wir in dieser Arbeit einen Ansatz zur Duplikaterkennung und Fusion von unvollständigen Informationen, welche durch so genannte Möglichkeitsverteilungen (*Possibility Distributions*) modelliert sind.

Abstract

Duplicate detection and data fusion are two essential prerequisites for obtaining concise results from data integration processes. Caused by many deficiencies in data collection, data modeling or data management, real-life data is often incorrect and/or incomplete. Thus, identifying and unifying multiple representations of the same real-world object is not trivial. Since in the relational data model incomplete information can be represented only by null values, current techniques of duplicate detection and data fusion primarily focus on the handling of dissimilarities resulting from typos, data obsolescence or misspellings. Usually, information on real-world phenomena is rarely complete but rather uncertain, imprecise or vague. Therefore, different data models based on fuzzy set theory or probabilistic theory for modeling incomplete information have been proposed. Unfortunately, current techniques for tuple matching and tuple merging are not designed to deal with such concepts. To enable an integration of data originating from different fuzzy databases, we present a first analysis in duplicate detection and data fusion w.r.t. incomplete information represented by possibility distributions.

Towards Duplicate Detection and Data Fusion in Fuzzy Relational Databases

Fabian Panse #1, Norbert Ritter #2

#Computer Science Department, University of Hamburg
Vogt-Koelln Straße 33, 22527 Hamburg, Germany
[1]panse@informatik.uni-hamburg.de
[4]ritter@informatik.uni-hamburg.de

***Abstract*— Duplicate detection and data fusion are two essential prerequisites for obtaining concise results from data integration processes. Caused by many deficiencies in data collection, data modeling or data management, real-life data is often incorrect and/or incomplete. Thus, identifying and unifying multiple representations of the same real-world object is not trivial. Since in the relational data model incomplete information can be represented only by null values, current techniques of duplicate detection and data fusion primarily focus on the handling of dissimilarities resulting from typos, data obsolescence or misspellings. Usually, information on real-world phenomena is rarely complete but rather uncertain, imprecise or vague. Therefore, different data models based on fuzzy set theory or probabilistic theory for modeling incomplete information have been proposed. Unfortunately, current techniques for tuple matching and tuple merging are not designed to deal with such concepts. To enable an integration of data originating from different fuzzy databases, we present a first analysis in duplicate detection and data fusion w.r.t. incomplete information represented by possibility distributions.**

I. Introduction

The relational data model is principally designed for modeling accurate information, but a lossless collection of all the actual facts of a modeled world is an optimistic and mostly unrealistic assumption. In contrast, a large amount of collected information is uncertain, imprecise or vague (e.g. information resulting from human observations). Therefore, in order to model such imperfectness several kinds of data models (e.g. probabilistic data models [1], [2], [3], [4], [5] and fuzzy data models [6], [7], [8]) have been developed and become more and more important in current database research.

Furthermore, today data is often distributed among multiple sources which, in turn, are distributed all over the world. Information processing moves from monolithic systems to federated systems and hence to the integration of data from multiple heterogeneous sources [9]. For a long time, in data integration research only source and target schemas defined within a relational ([10] et al.) or semi-structured ([11] et al.) data model have been regarded. However, since both concepts, data integration as well as the modeling of imperfect information, have moved into the spotlight of the database community, a consideration of an integration of data by using uncertain data models has been just a logical consequence. Using probabilistic target schemas (schemas defined in a probabilistic data model) in order to handle uncertainties in the integration of multiple relational data sources has been investigated in several works [12], [13], [14]. In general, an integration of certain source data to an uncertain target schema is already discussed to a large extent. In contrast, an integration process of data originating from probabilistic databases or fuzzy databases (uncertain source data) is still an unexplored area of research. Nevertheless, an integration of uncertain data from multiple sources is an essential property in order to enable valuable analyses of inter-relations between different sets of uncertain data.

Data integration can be roughly divided into two phases each in turn consisting of two steps. In the first phase, semantic correspondences between the individual schemas have to be identified and mappings from the source schemas to the target schema need to be specified [15], [16]. Therefore, in the first phase primarily metadata is processed. In the second phase, the operational data of the individual sources have to be consolidated to a common integration result. In general, such a consolidation only increases the completeness of the resulting data, but in order to enable an effective usage of this data the integration result also has to be concise [17]. This requires the identification [18] and unification [19] of duplicate representations of same real-world objects.

The main contribution of this paper is that it constitutes a first approach to considering the integration of uncertain source data. Since the whole process of data integration w.r.t. all kinds of uncertain data models exceeds by far the extent of this paper, we focus on fuzzy data models and especially on the second integration phase composed of duplicate detection and data fusion. We consider different methods and concepts of both activities w.r.t. imperfect information represented by possibility distributions. In general, this paper has not the goal of completely handling such an extensive field of research, but it gives first insights into this area and introduces some theoretical fundamentals.

The remainder of the paper is structured as follows: Section 1 gives an overview of fuzzy data models and Section 2 shortly presents the activities of data integration, duplicate detection and data fusion. In Section 3 we introduce different types of equivalence and propose techniques of matching tuples in fuzzy databases. Current data fusion techniques are adapted to fuzzy data in Section 4. Section 5 examines related work and Section 6 summarizes the paper and gives a conclusion.

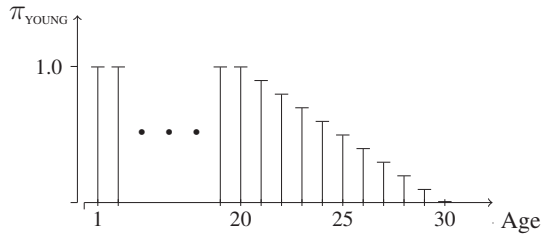

Fig. 1. Label YOUNG

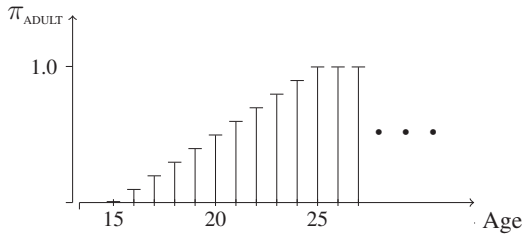

Fig. 2. Label ADULT

II. Fuzzy Relational Data Models

Similarity-based models and possibility-based models are the two major approaches which result from former research for introducing fuzzy set theory into databases. Similarity-based models (e.g. [6]) use similarity or proximity relationship functions to measure the nearness among different domain elements. Possibility-based models (e.g. [7], [8]) gather fuzzy information by using possibility distributions for attribute values. In this paper, as a representative we consider a mixed model which uses possibility distributions as well as similarity relationships and is hence one of the most powerful variants of fuzzy data models [20], [21]. Since a possibility distribution is based on the concept of fuzzy sets, we shortly present both the fuzzy set theory as well as the theory of possibility distributions, which are introduced by Zadeh [22], in more detail.

A. Fuzzy Sets and Possibility Distributions

In order to represent imprecise, uncertain and especially vague (”fuzzy”) information, elements of fuzzy sets have a ”degree of membership”. Thus, instead of a bivalent mapping saying that an element either belongs to a set or not, a more gradual differentiation is possible. A fuzzy set F is defined as a pair (A, μ) where A is the reference set (or discourse) and μ is the membership function $\mu : A \rightarrow [0, 1]$ of F, which expresses the degree of membership of the individual elements. A finite and discrete fuzzy set $F = (A, \mu)$ can be also expressed as:

$$F = \{\mu(a_1)/a_1, \ldots, \mu(a_n)/a_n\}, \quad a_i \in A,\ \mu(a_i) \neq 0$$

If a membership value $\mu(a)$ of a fuzzy set $F = (A, \mu)$ is explained to be a measure of the possibility that an attribute value X (which is defined in the domain A) is equal to the element $a \in A$, X is described by a possibility distribution $\Pi_X(A)$ with the possibility distribution function $\pi_X = \mu$ [23]. In this context, $\pi_X(a_i \in A)$ denotes the possibility that a_i is the *true value* of X. Since X takes only one value (its actual but unknown *true value*), all possible elements of A are mutually exclusive. If we assume that every attribute value is applicable, one of the elements of A has to be the actual *true value* of X. Thus, the possibility of at least one element $a \in A$ has to be $\pi_X(a) = 1$ and hence possibility distributions are always normalized. In general, if $\pi_X(a_1) > \pi_X(a_2)$ then a_1 is considered a more plausible value for X than a_2.

For simplification purposes, possibility distributions are often represented by linguistic labels (e.g. the labels YOUNG and ADULT shown in Figure 1 and 2). Since the concepts of fuzzy sets and their membership functions are interpreted as possibility distributions or labels and their corresponding possibility distribution functions all the properties of fuzzy sets are also applicable to possibility distributions. For example, the cardinality of a possibility distribution $\Pi_X(A)$ over a finite set A[1] is defined as:

$$Card(\Pi_X(A)) = \sum_{a \in A} \pi_X(a)$$

Since fuzzy sets are an extension of classical sets (every classical set is a special kind of fuzzy set) standard set operations as union and intersection can be defined. The most common definitions based on the s-norm *max()* and the t-norm *min()*:

- *Union:* The possibility distribution $\Pi_{X \cup Y}(A)$ resulting from the union of the two fuzzy sets X and Y has the distribution function:

$$\pi_{X \cup Y}(a) = max(\pi_X(a), \pi_Y(a))$$

- *Intersection:* The possibility distribution $\Pi_{X \cap Y}(A)$ resulting from the intersection of the two fuzzy sets X and Y has the distribution function:

$$\pi_{X \cap Y}(a) = min(\pi_X(a), \pi_Y(a))$$

- *Inclusion:* $\Pi_X(A)$ is said to be included in $\Pi_Y(A)$, if the possibility of each domain element to be the *true value* of X is lower or equal than the possibility of this element to be the *true value* of Y:

$$\Pi_X(A) \subseteq \Pi_Y(A) \Leftrightarrow (\forall a \in A) : \pi_X(a) \leq \pi_Y(a)$$

Considering the theory of fuzzy numbers, arithmetic functions as addition or multiplication can be defined for possibility distributions. For example, the possibility distribution $\Pi_{X+Y}(A)$ resulting from the sum of two fuzzy values X and Y with the respective possibility distributions $\Pi_X(A)$ and $\Pi_Y(A)$ can be defined [20] by the function:

$$\pi_{X+Y}(z) = sup_a\{min(\pi_X(a), \pi_Y(z - a))\}$$

For further details on fuzzy sets and possibility distributions we refer the interested reader to [23], [24] and [25].

[1]Since the handling of continuous and/or infinite possibility distributions is more complex and in the context of databases is more unusual, in this paper only discrete and finite ones are considered.

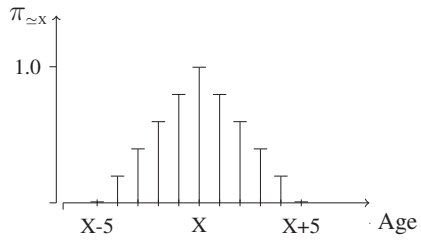

Fig. 3. approximately X (ca.X)

Similarity Haircolor	black	dark-brown	brown	light-brown	blond
black	1.0	0.8	0.5	0.2	0.0
dark-brown	0.8	1.0	0.9	0.6	0.2
brown	0.5	0.9	1.0	0.9	0.6
light-brown	0.2	0.6	0.9	1.0	0.8
blond	0.0	0.2	0.6	0.8	1.0

Fig. 4. Similarity relation of the haircolor domain

B. Fuzzy Attributes and Fuzzy Values

In order to represent fuzzy information in fuzzy relations classical attributes are extended to fuzzy attributes. Thus, for each attribute its domain D is extended to the domain $P(D)$, where $P(D)$ is the collection of all possibility distributions on D. The values of fuzzy attributes (fuzzy values) can represent different kinds of information:

- *crisp value*: These values are precise data as known from classical databases (e.g. age=25). A crisp value $X = c$ can be represented by the possibility distribution function $\pi_X = \{1/c\}$.
- *interval value*: An interval $[l, u]$ is a range in an ordered domain. This kind of imprecision can be represented by the possibility distribution function:

$$\pi_X(a) = \begin{cases} 1, & (\forall a \in [l, u]) \\ 0, & else \end{cases}$$

- *vague value*: These values represent vague information by using possibility distributions over ordered or non-ordered domains. As an example, we consider the vague information "the person is approximately X years old". A corresponding possibility distribution function is shown in Figure 3. In general, non-ordered and discrete domains are sets of labels (e.g. the possible colors of hair). A definition of a similarity relation (see Figure 4), which indicates to what extent two labels are similar, enables semantic comparisons between individual labels of non-ordered domains.
- *linguistic labels*: Linguistic labels are words in natural language which are linked with predefined possibility distributions (e.g. age=YOUNG). The corresponding domains can be ordered as well as non-ordered.

C. Fuzzy Database Relations

Besides incorporating fuzzy information into operational data by using possibility distributions for attribute values, fuzzy information can be also used in metadata in terms of fuzzy degrees. Fuzzy degrees can be used at different levels of granularity, but for simplification we only consider relations with a single fuzzy degree. Altogether other meanings (importance, possibility) are possible [20], we consider this degree as an uncertainty degree which specify the membership grade (the certainty to which a tuple belongs to a relation) of the individual tuples to the dedicated relation. Furthermore, we always consider fuzzy degrees as crisp values.

A fuzzy database is a collection of fuzzy database relations. Each fuzzy database relation is a set of n fuzzy attributes and hence is defined on several collections of possibility distributions over corresponding attribute domains (D_i):

$$\mathcal{R} = (P(D_1) \times P(D_2) \times ... \times P(D_n) \times [0, 1])$$

The last attribute is the fuzzy degree for representing the tuples' memberships and is therefore not a usual attribute. The tuple membership function of a relation $\mathcal{R}$ is defined as:

$$\begin{aligned} \mu_{\mathcal{R}}(t) &= P(D_1) \times P(D_2) \times ... \times P(D_n) \rightarrow [0, 1] \\ &= \{\mu_{\mathcal{R}}(t_1)/t_1, \mu_{\mathcal{R}}(t_2)/t_2, ..., \mu_{\mathcal{R}}(t_n)/t_n\} \end{aligned}$$

Two examples of a fuzzy database relation are shown in Figure 6. More information on fuzzy databases can be found in [20] and [26].

III. Data Integration

Although other operational areas are possible, we regard duplicate detection and data fusion as two phases of a data integration process. In general, the integration of multiple data sources (relational- (RDB) or in our case fuzzy relational databases (FRDB)) is composed of four steps (see Figure 5). In order to overcome schematic as well as semantic heterogeneities in the first two steps the source schemas are matched and mapped to the target schema. These activities bridge heterogeneity by identifying semantic relations between source and target schemas (schema matching) and determine how data conforming to the individual local schemas can be transformed to be conform to the global target schema (schema mapping). Altogether, the goal of the first two steps is that all objects of a certain type are represented in a homogeneous way.

After mapping data from the different sources to the target schema in order to obtain a concise result multiple representations of the same real-world object have to be discovered (duplicate detection) and unified (data fusion). If one of the sources is a fuzzy database or the target schema is a fuzzy database schema (e.g. to represent vagueness resulting from an imprecise schema matching, duplicate detection or data fusion) current techniques of all four steps have to be adapted to fuzzy information modeled in operational data (fuzzy values) as well as in metadata (fuzzy degrees).

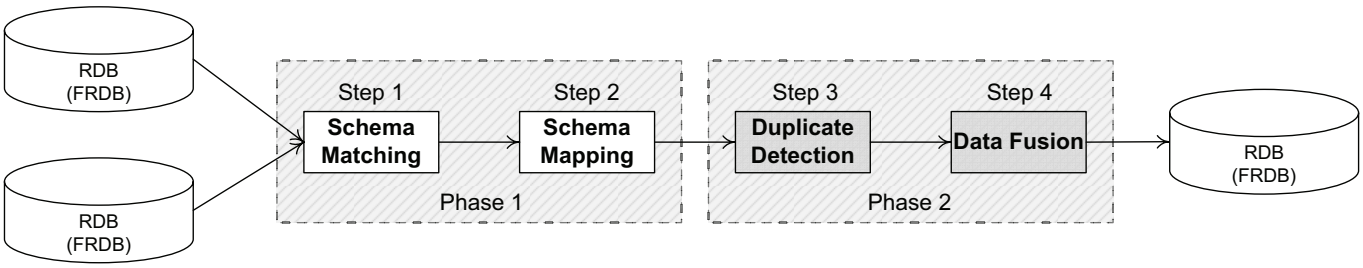

Fig. 5. Data integration process

While schema matching and schema mapping principally concern fuzzy metadata, duplicate detection and data fusion primarily affect fuzzy values. In general, both handling fuzzy metadata as well as fuzzy operational data play an important role for the integration of data originating from mutliple heterogenous fuzzy databases. However, we think that schema matching and schema mapping w.r.t. fuzzy databases is a topic of its own and is consequentially out of the scope of this paper.

In the following we refer to an integration process that incorporates the data of two fuzzy database relations R_1 and R_2 describing the first name, the age and the hair color of persons (see Figure 6) from two different sources into a single fuzzy schema. The linguistic labels YOUNG, ADULT and APPROXIMATELY ('ca.') are shown in Figure 1-3. The two labels DARK and LIGHT are the possibility distributions DARK={1/`black`, 0.8/`dark-brown`, 0.4/`brown`} and LIGHT={1/`blond`, 0.7/`light-brown`, 0.4/`brown`}.

We assume that after schema mapping the data from both sources is defined within a common fuzzy schema. Thus, we focus on the last two steps of data integration and present a global view of duplicate detection as well as data fusion before we consider both areas w.r.t. fuzzy values in the next sections in more detail.

A. Duplicate Detection

Duplicate Detection is an important data quality activity which is also known as record linkage, record matching, object identification, object resolution and many others. In the context of relational databases duplicate detection is mostly used in order to discover tuples that refer to the same object of the real world. Since tuples describing the same object often differ from each other, for example resulting from typos, subjective and/or erroneous data collections, or different times of data updates, duplicate detection is usually not trivial. In general, techniques for duplicate detection have a common structure [27] that can be described by five steps:

- **Data Preparation:** Data preparation (see [27]) is a preprocessing activity in order to minimize different representations of the same information resulting from different standards, measuring units or abbreviations.
- **Search Space Reduction:** In order to minimize the complexity of duplicate detection, first a reduction of the search space can be applied. Common techniques for search space reduction are sorted neighborhood, pruning or blocking [18], [27].
- **Comparison Functions:** After reducing the number of tuples that have to be compared to each other, functions for expressing the distances between attribute values of different tuples have to be chosen. Well-known functions for measuring the distance between two attribute values are the edit-distance, n-grams or the Jaro distance (for more details see [18]).
- **Decision Model:** A decision model [27], [28] is a method for assigning compared tuples to the set of matching tuples, the set of unmatching tuples or the set of possibly matching tuples, on the basis of the measured distances between the attribute values of both tuples. In general, such a decision has two contrary goals (precision-recall dilemma). First, to avoid clerical reviews as often as possible the set of possible matches has to be minimized. Second, to avoid incorrect decisions, the number of false positives and false negatives has to be reduced to a minimum.
- **Verification:** A closing verification (see [27]) checks the effectiveness of the applying methods in terms of recall, precision, false negative percentage, false positive percentage and F_1-measure. If the obtained effectiveness is not as expected, other comparison functions and decision models have to be chosen.

For comparing two fuzzy values, comparison functions cannot be directly used. Thus, functions for the matching of attribute values (step 3) in fuzzy databases is considered in Section 4 in more detail.

B. Data Fusion

After duplicate detection each tuple is assigned with an object identifier (object-ID) qualifying this tuple as a representation of the corresponding object. Thus by forming duplicate clusters all representations of an object are related to each other by the same object-ID. The goal of data fusion is to melt these multiple representations into a single one. Therefore, after an ideal data fusion no object has to be represented by more than one tuple. Two tuples of the same duplicate cluster can be in relation with each other in four different ways: Equality, subsumption, complementation and conflict. The most severe situation is given, if they are in conflict, meaning that they represent contrary information. In

	name	age	haircolor	$\mu_{\mathcal{R}_1}$
t_{11}	0.8/John,1/Johan	YOUNG	DARK	0.9
t_{12}	Johan	≤ 15	1/black,0.7/dark-brown,0.2/brown	1.0
t_{13}	1/Tim,1/Tom	ADULT	blond	0.4
t_{14}	0.7/Mia,1/Mira	33	1/brown,0.8/light-brown	0.5

	name	age	haircolor	$\mu_{\mathcal{R}_2}$
t_{21}	0.8/John,1/Johan	YOUNG	DARK	1.0
t_{22}	1/Johan,1/Johna	ca. 10	1/black,0.5/dark-brown	0.7
t_{23}	1/Tim,1/Jim,0.3/Kim	[24,25]	LIGHT	0.8
t_{24}	Kira	[32,34]	LIGHT	1.0

Fig. 6. Motivating example - fuzzy database relations $\mathcal{R}_1$ (Source 1) and $\mathcal{R}_2$ (Source 2)

general, there exists no major approach for resolving data conflicts. Several approaches may work extremely well in one application domain and fail in another. As a consequence, data fusion is a highly domain dependent task and has to be adapted to individual needs [19].

The most intuitive and most native approaches to unify data from different sources are relational combination techniques as union or join operators. However, the standard operators as Union, Equi-Join or Outer Join as well as the advanced operators as Outer-Union, Minimum Union or the Merge Operator are extremely limited w.r.t. handling conflicting data [19]. In Join approaches intrasource duplicates cannot be merged and data conflicts are generally ignored. In Union approaches only identical (Union or Outer-Union) or subsumed duplicates (Minimum Union) are removed. As with Join operators data conflicts are generally ignored. Nevertheless, better results for Join as well as Union operators can be achieved, if grouping (on the object-ID) and aggregation functions (for conflict resolving) are applied. Unfortunately, due to the small number of aggregation functions provided by the SQL standard, in most existing databases conflict resolution is very limited. As a consequence, to meet the requirements of different application domains additional adequate aggregation functions are required. In the context of the relational data model, such additional aggregation functions are considered in different works (e.g. [17]). In Section 5 we analyze data conflicts concerning fuzzy data and reflect on aggregation functions which can be used to resolve such conflicts between two non-crisp fuzzy values.

IV. Duplicate Detection in Fuzzy Databases

In general, two tuples are called duplicates, if they are equivalent. However, the interpretation of equivalence depends on the intended goal of the duplicate detection activity. For example, erasing double entries in relational operations such as Union or detecting multiple representations of the same real-world object generally imply two different types of equivalence. Usually for each type of equivalence different detection techniques are required.

A. Types of Equivalence

We define three types of equivalence: data equivalence (D-EQ), information equivalence (I-EQ) and real-world equivalence (RW-EQ):

- *Data and Information Equivalence:* Two tuples are data equivalent, if they are syntactically equal (e.g. the tuples t_{11} and t_{21} from the example in Figure 6 are data equivalent). Two tuples are information equivalent, if all the corresponding values of both tuples represent the same information and hence these tuples are semantically equal. With respect to crisp data, the two values (”06.07.2009”) and (”July 06. 2009”) are information equivalent (but not data equivalent). Two fuzzy values are information equivalent if they have the same possibility distribution. For example, two different linguistic labels representing the same possibility distribution (synonyms) are information equivalent, but not data equivalent. In contrast, in information integration, two sources can link the same linguistic label (homonyms) with different semantics (different possibility distributions). Two homonymous data values are data equivalent but not information equivalent.
Generally, in data preparation (see [27]) representation conflicts as synonyms or homonyms are resolved by standardization (e.g. renaming) and transformation (e.g. data type conversion). Thus, data preparation ensures that equivalent information is mostly represented by equivalent data. As a consequence, w.r.t. prepared data, data- and information equivalence are most often identical. With respect to fuzzy values, data preparation has to include activities to resolve synonyms and homonyms of linguistic labels.
Often data is marginally inconsistent, for example resulting from typos, subjective observations or measurement errors. Thus, to measure the syntactic as well as the semantic resemblance of two values, data equivalence and information equivalence can be considered as similarity measures within the range $[0, 1]$. As a consequence, data (information) equivalence can be interpreted as the extent to which two data values are syntactically (semantically) equivalent. The equality between data equivalence and information equivalence w.r.t. prepared data is limited for the case of absolute equality. Otherwise, both equivalences can extensively differ from each other. For example, the two colors `dark-brown` and `black` are semantically similar to a large extent, but the syntactic equivalence is low. For a counter-example, we consider the three labels `small`, `middle` and `tall`. From the syntactic point of view, `small` and `tall` are more similiar than `small` and `middle`, but the semantics of `small` and `tall` are more contrary to each other and hence more dissimilar than the semantics of `small` and `middle`.
- *Real-World Equivalence:* The generally most considered and in the context of data integration most important type of equivalence is real-world equivalence. Two tuples are

real-world equivalent, if they represent the same real-world object. For example, the two tuples t_{13} and t_{23} (see Figure 6) are real-world equivalent. Usually, identifying two tuples representing the same real-world object is more complex and hence more approximately than identifying two tuples representing the same information. In relational databases, real-world equivalence is reduced to data- and information equivalence by using syntactic (e.g. edit distance) and semantic (e.g. glossaries or ontologies) similarity measures. Two values are assumed to be real-world equivalent, if they are either data equivalent or information equivalent to a large extent. Thus, for the purpose of deriving real-world equivalence from data equivalence and information equivalence, the maximum function can be used:

$$\text{RW-EQ}(A, B) = max(\text{D-EQ}(A, B), \text{I-EQ}(A, B)) \quad (1)$$

In relational databases, a value is either totally known (a crisp value) or totally unknown (a null value). In contrast, in fuzzy databases, data can represent incomplete information in different kinds of degree. Thus, a reduction of real-world equivalence on data equivalence and information equivalence is far from being satisfactory if at least one of the considered fuzzy values is not crisp. For example, two fuzzy values each representing the linguistic label YOUNG are syntactically and semantically equal, but the *true value* of both fuzzy values can be different (e.g. 15 years and 5 years). Thus, in order to measure the real-world equivalence of two non-crisp fuzzy values the degree of incompleteness (e.g. vagueness or imprecision) has to be taken into account.

In general, the equivalence of two tuples follows from the equivalence of their attribute values. Thus, at first we consider the matching of two values before we examine tuple matching techniques.

B. Matching of Fuzzy Values

Since data equivalence is only of syntactic nature, data equivalence of fuzzy values is likely defined as for relational data values. If similarity relationships are defined, in fuzzy databases a more exact measuring of information equivalence of two crisp values is possible. In contrast to relational data, for matching non-crisp fuzzy values w.r.t. information similarity a comparison of different possibility distributions is required. As mentioned above, current methods for value matching w.r.t. real-world equivalence are also not adequate for non-crisp fuzzy values and need further investigation. In the following, we shortly present measuring techniques w.r.t. information equivalence before focusing on techniques for real-world equivalence.

1) Information Equivalence: If for two crisp fuzzy values a similarity score is defined (e.g. in the domain of the attribute haircolor, the similarity between `dark-brown` and `black` is defined as 0.8), this similarity can be used as the information equivalence of these two values. Thus, in such cases no further techniques for measuring the semantic equality are required.

To measure the information equivalence of two non-crisp fuzzy values is by far more difficult. In the literature (e.g. [24],[25]), there are multiple concepts to compare two possibility distributions (see comparison operations on fuzzy sets in [25]). In order to demonstrate the large spectrum of comparison methods, now we briefly present two of the most simple and most representative of them:

- An intuitive measure of information equivalence between two possibility distributions $\Pi_X(A)$ and $\Pi_Y(A)$ is the fraction of domain elements that are possible in both distributions (see equality index $REC(X, Y)$ in [24]):

$$\text{I-EQ(X,Y)} = \frac{Card(\Pi_X(A) \cap \Pi_Y(A))}{Card(\Pi_X(A) \cup \Pi_Y(A))} \quad (2)$$

  This measure is based on the idea that the more domain elements are possible (or impossible) in both fuzzy values, the more similar both values are.
- Another approach [20] is to use the distance between two possibility distributions as their degree of equality. The larger the distance, the smaller is their similarity. For measuring the distance between the possibility distributions of two comparable fuzzy values X and Y common distance functions (e.g. the Minkowski distance [20]) can be used:

$$d(\Pi_X(A), \Pi_Y(A)) = [\sum_A |\pi_X(a) - \pi_Y(a)|^p]^{1/p},\ p > 0$$

  Common specific cases of the Minkowski distance are the Hamming distance (p=1) or the Euclidean distance (p=2). The information equivalence can be derived as the additive inverse of the normalized distance (d_N).

$$\text{I-EQ(X,Y)} = 1 - d_N(\Pi_X(A), \Pi_Y(A)) \quad (3)$$

Unfortunately, with these measures no similarities of different domain elements are considered. Thus, only the information equivalence of error-free and standardized data can be correctly measured. For example, the similarity between two possibility distributions $\{1/\texttt{black}, 1/\texttt{blond}\}$ and $\{1/\texttt{dark-brown}, 1/\texttt{light-brown}\}$ is measured as 0, even though the real information similarity between both distributions is high. As a consequence, these approaches have to be extended in order to consider similarities of individual domain elements in future work.

2) Real-World Equivalence: In matching fuzzy values w.r.t. real-world equivalence, we do not want to know the similarity of two possibility distributions, but we are interested whether both values represent the same real-world phenomenon. Since crisp fuzzy values do not represent incomplete information, real-world equivalence can be measured as for relational data by reducing it to data equivalence and information equivalence. Regarding error-free non-crisp fuzzy values, we consider the real-world equivalence of two values as the probability that both fuzzy values have the same *true value*. Thus, to quantify the equivalence of both values, we transform the possibility distribution into a probabilistic statement assuming a uniform probability distribution on the corresponding

$$P_X(Y \mid \Pi_X(A)) = \frac{P(\{1/y\} \cap \Pi_X(A))}{P(\Pi_X(A))} = \frac{P(\{\pi_X(y)/y\})}{P(\Pi_X(A))} = \frac{\pi_X(y)\rho(y)}{\sum_{a \in A} \pi_X(a)\rho(a)} \quad (4)$$

$$P_X(Y \mid \Pi_X(A), \Pi_Y(A)) = \sum_{a \in A} P_X(a \mid \Pi_X(A)) \cdot P_Y(a \mid \Pi_Y(A)) = \sum_{a \in A} \frac{\pi_X(a)}{Card(\Pi_X(A))} \cdot \frac{\pi_Y(a)}{Card(\Pi_Y(A))} \quad (5)$$

$$\text{RW-EQ(X,Y)} = \sum_{a \in A} \sum_{b \in A} P_X(a \mid \Pi_X(A)) \cdot P_Y(b \mid \Pi_Y(A)) \cdot \theta(a, b) \quad (6)$$

attribute domain. Given a probability function $\rho(a)$ over the domain A, $P_X(\Pi_Y(A))$ is defined as the probability[2] that X is one of the possible values of Y (see probability of fuzzy events in [25]).

$$P_X(\Pi_Y(A)) = \sum_{a \in A} \pi_Y(a)\rho(a) \quad (7)$$

Since the *true value* of X has to be one of its possible values, for a fuzzy value X, the possibility distribution $\Pi_X(A)$ is given as a true event. Therefore, the probability that the *true values* of a non-crisp fuzzy value X and a crisp fuzzy value Y are equal is the conditional probability $P(X = Y|\Pi_X(A))$ (short $P_X(Y|\Pi_X)$) which is defined as follows (the derivation is shown in Equation 4):

$$P_X(Y \mid \Pi_X) = \frac{\pi_X(y)\rho(y)}{\sum_{a \in A} \pi_X(a)\rho(a)} \quad (8)$$

If a uniform distributed domain is assumed ($\rho(a_1) = \rho(a_2),\ \forall a_1, a_2 \in A$), this probability and hence the real-world equivalence RW-EQ(X,Y) result in:

$$P_X(Y|\Pi_X) = \frac{\pi_X(y)}{\sum_{a \in A} \pi_X(a)} = \frac{\pi_X(y)}{Card(\Pi_X(a))} \quad (9)$$

For example, the probability that the person represented by tuple t_{11} is 25 years old is calculated as:

$$P(t_{11}.age = 25) = \frac{\pi_{\text{YOUNG}}(25)}{Card(\Pi_{\text{YOUNG}}(\text{Age}))} = \frac{0.5}{25} = 0.02$$

The probability $P_X(Y|\Pi_X(A), \Pi_Y(A))$ (short $P_X(Y|\Pi_X, \Pi_Y)$) that two real-world phenomena each represented by a non-crisp fuzzy value (X and Y each defined in A which is uniform distributed) are equal is defined as (the derivation is shown in Equation 5):

$$P_X(Y|\Pi_X, \Pi_Y) = \sum_{a \in A} \frac{\pi_X(a)}{Card(\Pi_X(A))} \cdot \frac{\pi_Y(a)}{Card(\Pi_Y(A))} \quad (10)$$

For example, the probability that the persons which are represented by the tuples t_{12} and t_{22} have the same haircolor results in:

$$P(t_{12}.haircolor = t_{22}.haircolor) = \frac{1}{1.9} \cdot \frac{1}{1.5} + \frac{0.7}{1.9} \cdot \frac{0.5}{1.5} = 0.47$$

In order to consider mismatches resulting from incomplete data as well as incorrect data, for calculating the real-world equivalence of two fuzzy values X and Y, syntactic as well as semantic irregularities in real-life data have to be taken into account:

$$\theta(X, Y) = max(\text{D-EQ}(X,Y), \text{I-EQ}(X,Y))$$

In this case, the real-world equivalence of a non-crisp fuzzy value X and a crisp value Y is defined as:

$$\text{RW-EQ(X,Y)} = \sum_{a \in A} P_X(a|\Pi_X) \cdot \theta(a, Y) \quad (11)$$

The real-world equivalence of two non-crisp fuzzy values regarding erroneous data is defined accordingly (see Equation 6).

Note, we assume a uniform probability distribution on the attribute domains. Sometimes, however, other distributions are more suitable (e.g. it is more probable that a person is 20 years than 100 years old). In such cases, the probability $\rho(a)$ cannot be canceled as done in Equation 9.

C. Tuple Matching

From matching the n values of a tuple pair (t_X, t_Y) a comparison vector $\vec{c} = [c_1, ..., c_n]$ results, where each c_i represents the similarity score of the i'th attribute value of these two tuples. As in techniques for the relational model, the comparison vector $\vec{c}$ is the input of a tuple matching method which decides, if the tuples represent the same real-world object or not. In the literature, various tuple matching methods, for instance probabilistic matching models (e.g. Fellegi and Sunter Theory [28]) or distance-based techniques (for more detail see [27]), can be found.

We think, the membership of a tuple to a relation depends on the application context. For example, information on a person can be stored in two different relations ($\mathcal{R}_1$ and $\mathcal{R}_2$): one storing adults, the other storing people having a job. If we assume that the considered person is certainly 34 years old and jobless with a confidence of 90%, then the certainty that a tuple t_1 representing this person belongs to the first relation is $\mu_{\mathcal{R}_1}(t_1) = 1$, but the certainty that a corresponding tuple t_2 belongs to the the second relation is only $\mu_{\mathcal{R}_2}(t_2) = 0.1$. Note that both tuples represent the same person despite the significant difference in certainties. This illustrates that not tuple membership but only uncertainty on attribute value level should influence the duplicate detection process. Thus, for duplicate detection membership degrees are neglected and existing tuple matching methods can also be used in fuzzy relational databases.

[2]This is the probability that the *true value* of X is an element of the fuzzy set $F = (A, \mu),\ \mu(a) = \pi_Y(a)$.

	name	age	haircolor	$\mu_{\mathcal{R}_1}/\mu_{\mathcal{R}_2}$	o-ID
t_{11}	0.8/John,1/Johan	YOUNG	DARK	0.9	1
t_{12}	Johan	≤ 15	1/black,0.7/dark-brown,0.2/brown	1.0	1
t_{21}	0.8/John,1/Johan	YOUNG	DARK	1.0	1
t_{22}	1/Johan,1/Johna	ca. 10	1/black,0.5/dark-brown	0.7	1
t_{13}	1/Tim,1/Tom	ADULT	blond	0.4	2
t_{23}	1/Tim,1/Jim,0.3/Kim	[24,25]	LIGHT	0.8	2
t_{14}	0.7/Mia,1/Mira	33	1/brown,0.8/light-brown	0.5	3
t_{24}	Kira	[32,34]	LIGHT	1.0	3

Fig. 7. Duplicate cluster

V. Data Fusion in Fuzzy Databases

We assume that after duplicate detection all tuples are clustered w.r.t. their corresponding object-IDs (see Figure 7). In order to achieve a concise integration result all the tuples of one cluster have to be fused to a single representation. If duplicate detection was applied w.r.t. real-world equivalence, the tuples of one cluster do not have to be data- or information equivalent. Accordingly, different data conflicts can occur and have to be handled during data fusion.

A. Data Conflicts

Two representations of the same real-world object can be in relation to each other in four different cases [29]: Equality, subsumption, complementation and conflict (the definitions below are formalized in Figure 8):

1) *Equality:* In both, in relational as well as fuzzy relational databases, two tuples t_X and t_Y, specified on the attributes $A_i \in A$ (each defined in D_i), are said to be equal (EQ(t_X, t_Y)), if all their corresponding attribute values are data equivalent. For example, the tuples t_{11} and t_{21} (Figure 7) are equal.
2) *Subsumption:* In relational databases, a tuple t_X is said to subsume a tuple t_Y (SUB(t_X, t_Y)), if for every attribute the value of both tuples are equal or t_Y is a null value. If we abstract from the limited representation capabilities of the relational model, a tuple t_X subsumes a tuple t_Y, if each attribute value of t_X represents the same or more exact information than the associated attribute value of t_Y without any contradictions. Thus, w.r.t. fuzzy databases, a tuple t_X subsumes a tuple t_Y, if for every attribute A each possible domain element of $t_X.A$ is also possible for the fuzzy value $t_Y.A$ with a possibility equal or higher than for $t_X.A$ (the possibility distribution of $t_Y.A$ includes the possibility distribution of $t_X.A$). For example, the tuple t_{12} subsumes the tuple t_{11} (see Figure 7).
3) *Complementation:* In fuzzy databases, two tuples t_X and t_Y are said to be complementary (CMP(t_X, t_Y)), if no subsumption exists, but for every attribute the possibility distribution of one value is included in the possibility distribution of the other value. For example, the tuples t_{12} and t_{22} are complementary.
4) *Conflict:* Two tuples t_X and t_Y are said to be in conflict (CF(t_X, t_Y)), if at least one pair of attribute values represent contrary information. For example, since the possibility distributions of the attribute *name* of the tuples t_{13} and t_{23} are disjoint both tuples are in conflict.

In order to handle situations of equality, subsumption and complementation quite simple concepts can be used: If two tuples are equal, one of them can be omitted. If a tuple subsumes another one, the subsumed tuple can be dropped. In the situation that two tuples are complementary, a new tuple t_N can be created by using the more exact value of both tuples for each attribute. For example, the complementary tuples t_{12} and t_{22} can be fused to:

$$t_N = (Johan,\ ca.10,\ \{1/\texttt{black}, 0.5/\texttt{dark-brown}\})$$

Since t_{12} subsumes t_{11} and t_{21}, all tuples of cluster 1 can be fused to the single tuple t_N. If two tuples are in conflict, fusion is more complex and the handling of such situations has to be traced back to handling the conflicts of the individual attribute values. On attribute value level, two types of data conflicts exist [17]:

1) *Uncertainties:* Two attribute values are in an uncertainty conflict, if it is uncertain whether both *true values* are equal or not. In the relational model, an uncertainty conflict between two values exists, if at least one of them is a null value. With respect to fuzzy databases, an uncertainty conflict between two fuzzy values exists, if at least one of the fuzzy values is not crisp and there is at least one value which is possible for both of them (e.g. the two fuzzy values t_{13}.*haircolor* and t_{23}.*haircolor* or the two fuzzy values t_{13}.*name* and t_{23}.*name* have an uncertainty conflict).
2) *Contradictions:* In contrast, two attribute values are contradictory, if they are certainly inconsistent descriptions of the same real-world property and it can be excluded that both fuzzy values have the same *true value*. In the relational model, two or more values are contradictory if they are distinct and not null values. With respect to fuzzy databases, a contradiction conflict between two fuzzy values exists if there is no value which is possible for both fuzzy values and hence the intersection of the corresponding possibility distributions is empty (e.g. the two fuzzy values t_{14}.*name* and t_{24}.*name* are contradictory).

In order to manage uncertainties as well as contradictions conflict handling strategies are used during data fusion.

$$
\begin{aligned}
\text{EQ}(t_X, t_Y) &= (\forall A_i \in A) : \Pi_{t_X.A_i}(D_i) = \Pi_{t_Y.A_i}(D_i) \\
\text{SUB}(t_X, t_Y) &= (\forall A_i \in A) : \Pi_{t_X.A_i}(D_i) \subseteq \Pi_{t_Y.A_i}(D_i) \\
\text{CMP}(t_X, t_Y) &= (\forall A_i \in A) : \neg(\text{SUB}(t_X, t_Y) \vee \text{SUB}(t_Y, t_X)) \wedge (\Pi_{t_X.A_i}(D_i) \subseteq \Pi_{t_Y.A_i}(D_i) \vee \Pi_{t_Y.A_i}(D_i) \subseteq \Pi_{t_X.A_i}(D_i)) \\
\text{CF}(t_X, t_Y) &= (\exists A_i \in A) : \Pi_{t_X.A_i}(D_i) \not\subseteq \Pi_{t_Y.A_i}(D_i) \wedge \Pi_{t_Y.A_i}(D_i) \not\subseteq \Pi_{t_X.A_i}(D_i)
\end{aligned}
$$

Fig. 8. The four different cases of conflict situations

B. Conflict Handling Strategies

Bleiholder and Naumann classify conflict handling strategies into three main classes [17]: conflict ignorance, conflict avoiding and conflict resolution (see Figure 9). Conflict ignorance strategies are mostly not aware of data conflicts and if so, they do not make a decision as to what to do with such conflicts. A representative of this class is the strategy PASS IT ON which passes conflicts to the user and lets the user decide how to handle possible conflicts. Conflict avoiding strategies acknowledge but do not resolve existing data conflicts. Representatives are TRUST YOUR FRIENDS (takes the data from the most trusted source) or TAKE THE INFORMATION (takes the non-null value). While from conflict ignoring strategies at most complete but not concise data results, conflict avoiding strategies supply at most a concise but not complete result. In contrast, conflict resolution strategies are more adequate to supply complete as well as concise data. These strategies are able to acknowledge and resolve existing data conflicts by regarding instance data as well as metadata. With respect to fuzzy databases, conflict ignoring and conflict avoiding strategies only marginally differ from strategies for relational databases. Thus, in the following, we focus on conflict resolving strategies and how they can resolve conflicts between two or more fuzzy values by applying conflict resolution functions.

C. Conflict Resolution Strategies

Conflict resolution strategies are further classified into deciding and mediating strategies. Both classes either depend on operational data values (instance based) or take additional metadata (metadata based) into account (see Figure 9).

1) Deciding Strategies: Deciding strategies choose one of the present attribute values. Examples for instance based deciding strategies defined for relational databases which can be used also for fuzzy databases are CRY WITH THE WOLVES (takes the most frequent attribute value among the conflicting ones) or ROLL THE DICE (chosen randomly). Furthermore, w.r.t. fuzzy databases additional strategies are possible, e.g., a strategy which takes the most precise fuzzy value (e.g. the fuzzy value whose possibility distribution has the lowest cardinality). In metadata based deciding strategies metadata, e.g., quality values, are used to decide which of the conflicting values is most suitable. An example for a metadata based deciding strategy is KEEP UP TO DATE which chooses the most recent value. In fuzzy databases, additional metadata as fuzzy degrees (e.g. the membership degree of the corresponding tuples) can be additionally taken into account.

2) Mediating Strategies: In contrast to deciding strategies, mediating strategies resolve conflicts by creating a new attribute value (e.g. the average or the median) which is suitable to represent all the conflicting ones. A representative for instance based mediating strategies is MEET IN THE MIDDLE (take an attribute value which is as close as possible to all present ones). Additional instance based mediating strategies for fuzzy databases are possible by using a fuzzy set-based resolution function, as for example one of the three functions `UNION`, `INTERSECTION` or `WEIGHTED UNION` which are presented in the following section. In order to invent a representing attribute value, in metadata based mediating strategies, suitable metadata (e.g. up-to-dateness or other quality values as reliability) is taken into account by weighting the individual attribute values with their corresponding quality.

D. Conflict Resolution Functions

Usually, in order to resolve conflicts on the attribute value level all representations of one real-world object are grouped and fused by applying a conflict resolution function. Since a conflict resolution function aggregates multiple values to a single one, these functions can be seen as a more general case of aggregation as known from the SQL-standard [19]. Every resolution function can be formalized as a function with the conflicting values as input and the resolved value as output. In relational databases, each resolution function defined over a domain D is a mapping of n crisp values on a single crisp value. With respect to fuzzy databases resolution functions can be classified into four classes:

1) *Crisp-to-Crisp:* Resolution functions of this class are of the form

$$
\begin{aligned}
& f : D^n \rightarrow D \\
\Rightarrow \quad & f(c_1, ..., c_n) = s, \;\; c_1, ..., c_n, s \in D
\end{aligned}
$$

if only instance data is used and of the form

$$
\begin{aligned}
& f : D^n \times A \rightarrow D \\
\Rightarrow \quad & f(c_1, ..., c_n, a) = s, \;\; a \in A, c_1, ..., c_n, s \in D
\end{aligned}
$$

if additional meta information (represented by the input parameter A) is regarded [17]. Since non-crisp fuzzy values can be reduced to crisp values by defuzzification (e.g. *center-of-gravity* or the *center-of-area*

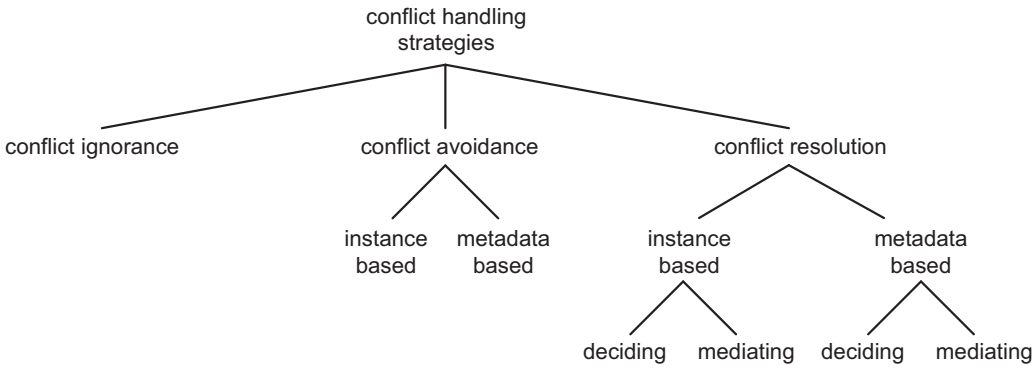

Fig. 9. A classification of strategies for handling inconsistent data [17].

method [24]), resolution functions defined for relational databases can be also used in fuzzy databases.

2) *Fuzzy-to-Fuzzy:* Resolution functions which map n non-crisp values on a single non-crisp value are of the form

$$f : P(D)^n \rightarrow P(D)$$
$$\Rightarrow \quad f(\Pi_{c_1}(D), ..., \Pi_{c_n}(D)) = \Pi_s(D)$$

if only operational data is used and of the form

$$f : P(D)^n \times A \rightarrow P(D)$$
$$\Rightarrow \quad f(\Pi_{c_1}(D), ..., \Pi_{c_n}(D), a) = \Pi_s(D),\ a \in A$$

if metadata is taken into account. Since arithmetic functions (e.g. addition and division) are also defined for fuzzy sets (see the theory of fuzzy numbers [25]), the common resolution functions *min*, *max*, *avg*, *median* and *sum* can be defined and hence applied for all kinds of fuzzy values, too. Altogether, an enormous number of conflict resolution functions of the form $f : P(D)^n \rightarrow P(D)$ is possible (see aggregation operations on fuzzy sets in [25]). In the following, we will take a closer look at two intuitive and suitable mediating conflict resolution functions which are already known from other application domains: `INTERSECTION` and `UNION`.

- `INTERSECTION` ($\cap$) is a function which can be used for resolving uncertainty-conflicts. In relational databases, `INTERSECTION` corresponds to the strategy of using the single non-null value. In fuzzy databases, `INTERSECTION` takes the elements which are possible for all conflicting fuzzy values and excludes those elements which are only possible for a few of them. Let $X_0, X_1, \ldots, X_n$ be n fuzzy values defined in the domain D which have to be fused to a single one. From applying `INTERSECTION` ($\cap(X_0, X_1, ..., X_n)$) the possibility distribution $\Pi_{X_0 \cap X_1 \cap ... \cap X_n}(D)$ result. Its corresponding distribution function is presented in Figure 10. Since `INTERSECTION` requires at least one element which is possible for all fuzzy values the function can only be used for resolving uncertainty-conflicts.
  For example, by using `INTERSECTION` t_{13}.*name* and t_{23}.*name* can be fused to:
  $$t_N.name = \{1/Tim\}$$
- `UNION` ($\cup$) a function which can be used for resolving uncertainty-conflicts as well as contradictions. Let $X_0, X_1, \ldots, X_n$ be n fuzzy values which have to be combined. From applying `UNION` ($\cup(X_0, X_1, ..., X_n)$) the possibility distribution $\Pi_{X_0 \cup X_1 \cup ... \cup X_n}(D)$ result. Its correspodning distribution function is presented in Figure 10.
  For example, from applying the `UNION` function, the two fuzzy values t_{14}.*name* and t_{24}.*name* can be fused to:
  $$t_N.name = \{0.7/Mia, 1/Mira, 1/Kira\}$$

`INTERSECTION` as well as `UNION` can be chosen for resolving uncertainty conflicts. Since by using `INTERSECTION` only the elements which are possible for all fuzzy values are respected, the resulting fuzzy value is more precise than the value resulting from `UNION`. Nevertheless, by omitting some elements, it could be the case, that the actual *true value* of the corresponding object property is dropped. Thus, by using `INTERSECTION` the result is certainly more precise, but likely also more unsound (incorrect) than by resolving the uncertainty-conflict by `UNION`.
Intuitively, an element which is possible for the fuzzy values of multiple duplicates is more plausible to be the *true value* of the corresponding object property than an element which is only possible for the fuzzy value of just a few of these tuples. Thus, in order to combine the benefits of `UNION` and `INTERSECTION`, we introduce the `WEIGHTED UNION` function which is based on a special compensatory operator for fuzzy sets [25].

- `WEIGHTED UNION` ($\Cup$) is a family of functions which consider all elements which are possible for at least one fuzzy value (as `UNION`), but enhance the possibility of these elements which are possible for multiple fuzzy values (similar to `INTERSECTION`).

$$
\begin{array}{rrcl}
\texttt{INTERSECTION}: & \pi_{X_0 \cap X_1 \cap \ldots \cap X_n}(d) & = & \pi_{X_0}(d) \cap \pi_{X_1}(d) \cap \ldots \cap \pi_{X_n}(d) \;=\; min(\pi_{X_0}(d), \ldots, \pi_{X_n}(d)) \\
\texttt{UNION}: & \pi_{X_0 \cup X_1 \cup \ldots \cup X_n}(d) & = & \pi_{X_0}(d) \cup \pi_{X_1}(d) \cup \ldots \cup \pi_{X_n}(d) \;=\; max(\pi_{X_0}(d), \ldots, \pi_{X_n}(d)) \\
\texttt{WEIGHTED UNION}: & \pi_{X_1 \Cup X_2 \Cup \ldots \Cup X_n}(d) & = & \sum_{i=1}^{n} w_i \cdot \pi_{X_i}(d), \quad \text{with} \sum_{i=1}^{n} w_i = 1
\end{array}
$$

Fig. 10. Fuzzy set-based resolution functions

Thus, a compromise between precision and soundness can be achieved. First we consider each `WEIGHTED UNION` as a binary operator which merges the possibility distributions of two fuzzy values. The result of applying a binary `WEIGHTED UNION` ($\Cup(X, Y)$) has the possibility distribution $\Pi_{X \Cup Y}(D)$ with the distribution function:

$$
\begin{aligned}
\pi_{X \Cup Y}(d) &= \frac{1}{2}\pi_X(d) + \frac{1}{2}\pi_Y(d) \\
&= \frac{1}{2}\pi_{X \cup Y}(d) + \frac{1}{2}\pi_{X \cap Y}(d)
\end{aligned}
$$

If we consider each `WEIGHTED UNION` as an n-ary operator, an element is the more plausible to be the *true value*, the more fuzzy values exist for which this element is possible. Thus the result of an n-ary `WEIGHTED UNION` ($\Cup(X_1, X_2, \ldots, X_n)$) has the possibility distribution $\Pi_{X_1 \Cup X_2 \Cup \ldots \Cup X_n}(D)$. Since the reliability of each input value can be different, we introduce a weight for each of the considered fuzzy values. The resulting family of corresponding distribution functions is presented in Figure 10. For example, in strategies only based on instance data, an appropriate weighting of the individual sources is the uniform weighting ($\forall i \in [1, n], w_i = \frac{1}{n}$):

$$
\pi_{X_1 \Cup X_2 \Cup \ldots \Cup X_n}(d) = \sum_{i=1}^{n} \frac{1}{n}\pi_{X_i}(d)
$$

In contrast, in metadata-based strategies quality values as the reliabilities of the corresponding sources or the up-to-dateness of the conflicting values can be used for determining an adequate weighting function.

Since a possibility distribution has to be normalized, the result of $\cap$, $\cup$ or $\Cup$ have to be divided by their highest possibility if necessary.

3) *Crisp-to-Fuzzy:* Using the capabilities of possibility distributions, instead of deriving a crisp value from multiple crisp values additional concepts for conflict resolution are possible (e.g. the union of all conflicting values as it is proposed in [30]). Sometimes, e.g., in order to integrate contradictory crisp data into a target fuzzy schema, it is suitable to aggregate several conflicting crisp values to a non-crisp value. Since every crisp value can be described by a possibility distribution, the resolution functions of this class are special cases of the last class's functions and can be used to integrate data from relational source schemas in a fuzzy target schema.
4) *Fuzzy-to-Crisp:* By concatenating a fuzzy-to-fuzzy resolution function $f_1 : P(D)^n \rightarrow P(D)$ with a function for defuzzification $f_2 : P(D) \rightarrow D$, multiple non-crisp fuzzy values can be mapped on a single crisp value ($f_2 \circ f_1 = f_3 : P(D)^n \rightarrow D$). Such functions are required, if data from several fuzzy databases has to be integrated into a relational schema.

E. Fusing Fuzzy Degrees

In order to resolve conflicts between metadata (e.g. fuzzy degrees) similar strategies as for operational data can be used. For example, two suitable strategies for resolving conflicts between two or multiple degrees of membership is to take the average of all memberships degrees (mediating strategy) or to take the degree representing the most certain information on membership (deciding strategy). A tuple certainly belongs to a relation, if its membership degree is 1 and does certainly not belong to a relation, if its membership degree is 0. As a consequence, the highest uncertainty of membership is modeled by a degree of 0.5. The certainty of the membership of a tuple t to a relation R can be calculated as:

$$
\text{Certainty}(\mu_{\mathcal{R}}(t)) = |2(\mu_{\mathcal{R}}(t) - 0.5)|
$$

If data is integrated by unifying data of multiple sources (see the *Union-Merge* operator $\sqcup$ defined in [31]), a tuple belongs to the integration result, if it belong to one of the source relations. In this case, the maximal tuple membership has to be used. Thus, by using the *Union-Merge* for an integration of the two relations $\mathcal{R}_1$ and $\mathcal{R}_2$, the tuple membership of the integration result $\mathcal{R}_1 \sqcup \mathcal{R}_2$ is defined as:

$$
\mu_{\mathcal{R}_1 \sqcup \mathcal{R}_2}(t) = max(\mu_{\mathcal{R}_1}(t), \mu_{\mathcal{R}_2}(t))
$$

Furthermore, as for conflict resolution in operational data, additional metadata as quality values (e.g. the reliabilities of the corresponding sources) can be taken into account.

F. Tuple Fusion

Two or multiple tuples are fused by merging their attribute values and membership degrees. For instance, if in our example the `WEIGHTED UNION` is used for resolving all existing conflicts, the resulting data is shown in Figure 11. In contrast, the integration result by using `INTERSECTION` for resolving

	name	age	haircolor	$\mu_{\mathcal{R}_1}$
t'_1	Johan	ca.10	1/black,0.5/dark-brown	0.9
t'_2	1/Tim,0.5/Tom,0.5/Jim,0.15/Kim	0.95/24,1/25	1/blond,0.35/light-brown,0.2/brown	0.6
t'_3	0.7/Mia,1/Mira,1/Kira	0.5/32,1/33,0.5/34	0.67/blond,1/light-brown,0.93/brown	0.75

Fig. 11. Integrated relation resulting from using `WEIGHTED UNION` with the weights $w_1 = w_2 = 0.5$

	name	age	haircolor	$\mu_{\mathcal{R}_1}$
t'_1	Johan	ca.10	1/black,0.5/dark-brown	0.9
t'_2	Tim	0.95/24,1/25	blond	0.6
t'_3	0.7/Mia,1/Mira,1/Kira	33	1/light-brown,0.57/brown	0.75

Fig. 12. Integrated relation resulting from using `INTERSECTION` (uncertainties) and `UNION` (contradictions)

uncertainties and `UNION` for resolving contradictions is shown in Figure 12. The result of the second strategy is certainly more precise than the result of the first one, but it is also more unsound, if for example the true age of the person represented by the tuple t'_3 is 34, not 33. Furthermore, the first approach is associative, meaning that the integration result is independent from the fusion order. In conclusion, this example clarifies, choosing an adequate resolution function is generally a trade-off between precision and soundness.

VI. Related Work

Duplicate detection and data fusion are two extensively investigated fields of research. In current approaches duplicate detection is mostly considered w.r.t. the relational data model [32], [28], [33], [18] or any semi-structured data model (e.g. for XML [34]) where approximate duplicates are often denoted as fuzzy duplicates [35], [36]. Nevertheless, uncertain source data, as for example fuzzy data, is not considered in these works. On the other hand, many proposals which focus on data preparation (e.g. [37]), search space reduction (e.g. [33]), decision models (e.g. [28]) or verification (e.g. [27]) can be adopted for duplicate detection in fuzzy databases. Furthermore, existing similarity functions for comparing two attribute values (e.g. [18]) can also be incorporate into techniques for comparing two fuzzy values.

Shahri et al. [36] use the theory of fuzzy logic in order to enhance the detection of approximate duplicates in relational databases. Their fuzzy inference engine enables an handling of uncertainty in deduplication by using matching rules specified in natural language instead of defining certain thresholds. As a consequence, domain experts are unburdened from the requirement of making certain decisions.

The fusion of relational data is considered in [19], [17]. Since a fuzzy relational data model is a generalization of the relational data model, the feasibility of these fusion methods is limited to crisp fuzzy values. In general, for the fusion of non-crisp fuzzy values these methods have to be also considered in a more general way.

Some former proposals handle the uncertainty arising in schema integration [12], duplicate detection [14] and data fusion [30], [13], [14] by using uncertain data models.

DeMichiel [30] and Tseng [13] introduce concepts of modeling uncertainty in order to resolve conflicts between two or more relational values. Thus, in these approaches relational data is fused into a model capable to represent uncertain and incomplete information by using functions for fusing multiple crisp values to a partial value [30] or a probabilistic partial value [13] respectively. Since we additionally focus on the fusion of multiple non-crisp values to a single non-crisp value, if the target schema is a fuzzy schema or on the fusion of multiple non-crisp fuzzy values to a single crisp value, if the target schema is only relational, we consider fusion of imperfect data from a more general point of view.

In [12], the authors propose probabilistic schema mappings for handling the uncertainty which occur during the integration of two or more relational schemas. Mappings of probabilistic schemas or melting uncertain source data are not considered by them so far.

Van Keulen and de Keijzer [38], [39], [14] define a probabilistic XML model in order to manage uncertainties resulting from entity resolution (duplicate detection) and conflict resolution (data fusion) in certain data. However, deduplication of probabilistic source data or fuzzy source data is not covered in their works.

VII. Conclusion

We started from the observation that current techniques of duplicate detection and data fusion are not designed to deal with source data representing fuzzy information. As a consequence, for achieving a concise result from the integration of data originating from different fuzzy databases, we have adapted existing approaches for deduplication to handling imperfect information modelled by possibility distributions.

We have considered duplicate detection w.r.t. different types of equivalence and have presented techniques for measuring information equivalence and real-world equivalence of two non-crisp fuzzy values. In this process, we primarily have focused on real-world equivalence, meaning that two tuples are equivalent, if they represent the same real-world object. If tuples only contain crisp data, an identification of multiple representations of the same real-world object can be traced back to the syntactic and semantic resemblance of the concerned tuples. However, for tuples containing non-crisp fuzzy values such an approach is not adequate. Therefore, we have defined the real-world equivalence of two fuzzy values as the probability that the *true value* of both attribute values are semantically or syntactically similar to a large extent.

Existing definitions and concepts of data fusion, as for example conflict situations between duplicate tuples, data conflicts on attribute value level and conflict handling strategies, are redefined w.r.t. fuzzy values. Furthermore, we have presented four classes of resolution functions which are suitable for an integration of fuzzy source data into a fuzzy target schema or a relational target schema, or for an integration of relational source data into a fuzzy target schema. In this context, we have taken a closer look at three conflict resolution functions, namely intersection, union and priority union, which can be used to fuse multiple conflicting non-crisp fuzzy values to a single one.

In conclusion, this paper gives first ideas for identifying and unifying duplicates in fuzzy databases. Individual subareas have to be investigated in more detail and will be topic of future reflections. For example the proposed functions for measuring information equivalence do not respect syntactic as well semantic similarities between individual domain elements and hence are not suitable for regarding irregularities in real-life data. In addition, a closer examination will reveal the suitability and existing limitations of possible resolution functions w.r.t. different application domains. Furthermore, in order to design deduplication techniques for complex data, fuzzy functional dependencies have to be taken into account. Last but not least schema matching and schema mapping w.r.t. fuzzy databases are also two unexplored fields of research which have to be investigated in future work.

REFERENCES

[1] D. Barbará, H. Garcia-Molina, and D. Porter, "The Management of Probabilistic Data," *IEEE Trans. Knowl. Data Eng.*, vol. 4, no. 5, pp. 487–502, 1992.

[2] E. Wong, "A Statistical Approach to Incomplete Information in Database Systems," *ACM Trans. Database Syst.*, vol. 7, no. 3, pp. 470–488, 1982.

[3] N. Fuhr and T. Rölleke, "A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems," *ACM Trans. Inf. Syst.*, vol. 15, no. 1, pp. 32–66, 1997.

[4] R. Cavallo and M. Pittarelli, "The theory of probabilistic databases," in *VLDB*, 1987, pp. 71–81.

[5] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. U. Nabar, T. Sugihara, and J. Widom, "Trio: A system for data, uncertainty, and lineage," in *VLDB*, 2006, pp. 1151–1154.

[6] B. P. Buckles and F. E. Petry, "A fuzzy representation of data for relational databases," *Fuzzy Sets and Systems*, vol. 7, pp. 213–226, 1982.

[7] H. Prade and C. Testemale, "Generalizing Database Relational Algebra for the Treatment of Incomplete/Uncertain Information and Vague Queries," *Inf. Sci.*, vol. 34, no. 2, pp. 115–143, 1984.

[8] M. Umano and S. Fukami, "Fuzzy relational algebra for possibility-distribution-fuzzy-relational model of fuzzy data," *J. Intell. Inf. Syst.*, vol. 3, no. 1, pp. 7–27, 1994.

[9] A. Y. Halevy, A. Rajaraman, and J. J. Ordille, "Data Integration: The Teenage Years," in *VLDB*, 2006, pp. 9–16.

[10] M. Lenzerini, "Data Integration: A Theoretical Perspective," in *PODS*, 2002, pp. 233–246.

[11] D. Draper, A. Y. Halevy, and D. S. Weld, "The Nimble XML Data Integration System," in *ICDE*, 2001, pp. 155–160.

[12] X. L. Dong, A. Y. Halevy, and C. Yu, "Data integration with uncertainty," *VLDB J.*, vol. 18, no. 2, pp. 469–500, 2009.

[13] F. S.-C. Tseng *et al.*, "Answering Heterogeneous Database Queries with Degrees of Uncertainty," *Distributed and Parallel Databases*, vol. 1, no. 3, pp. 281–302, 1993.

[14] M. van Keulen, A. de Keijzer, and W. Alink, "A Probabilistic XML Approach to Data Integration," in *ICDE*, 2005, pp. 459–470.

[15] M. A. Hernández, R. J. Miller, and L. M. Haas, "Clio: A Semi-Automatic Tool For Schema Mapping," in *SIGMOD Conference*, 2001, p. 607.

[16] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *VLDB J.*, vol. 10, no. 4, pp. 334–350, 2001.

[17] J. Bleiholder and F. Naumann, "Conflict handling strategies in an integrated information system," Humboldt-Universität Berlin, Tech. Rep., 2006.

[18] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate Record Detection: A Survey," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 1, pp. 1–16, 2007.

[19] J. Bleiholder and F. Naumann, "Data fusion," *ACM Comput. Surv.*, vol. 41, no. 1, 2008.

[20] J. Galindo, A. Urrutia, and M. Piattini, *Fuzzy Databases - Modeling, Design and Implementation*. Idea Group Publishing, 2006.

[21] J. M. Medina, O. Pons, and M. A. V. Miranda, "GEFRED: A Generalized Model of Fuzzy Relational Databases," *Inf. Sci.*, vol. 76, no. 1-2, pp. 87–109, 1994.

[22] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[23] Lotfi A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.

[24] D. Dubois and H. Prade, *Fundamentals of Fuzzy Sets*. Kluwer Academic Publishers, 2000.

[25] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets - Analysis and Design*. The MIT Press, 1998.

[26] F. E. Petry, *Fuzzy Databases - Principles and Applications*. Kluwer Academic Publishers, 1996.

[27] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*, ser. Data-Centric Systems and Applications. Springer, 2006.

[28] I. Fellegi and A. Sunter, "A Theory for Record Linkage," *Journal of the American Statistical Association*, vol. 64, pp. 1183–1210, 1969.

[29] J. Bleiholder, S. Szott, M. Herschel, F. Kaufer, and F. Naumann, "Subsumption and complementation as data fusion operators," in *EDBT*, 2010, pp. 513–524.

[30] L. G. DeMichiel, "Resolving Database Incompatibility: An Approach to Performing Relational Operations over Mismatched Domains," *IEEE Trans. Knowl. Data Eng.*, vol. 1, no. 4, pp. 485–493, 1989.

[31] F. Naumann *et al.*, "Completeness of integrated information sources," *Inf. Syst.*, vol. 29, no. 7, pp. 583–615, 2004.

[32] O. Benjelloun, H. Garcia-Molina, D. Menestrina, Q. Su, S. E. Whang, and J. Widom, "Swoosh: a generic approach to entity resolution," *VLDB J.*, vol. 18, no. 1, pp. 255–276, 2009.

[33] M. A. Hernández and S. J. Stolfo, "The Merge/Purge Problem for Large Databases," in *SIGMOD Conference*, 1995, pp. 127–138.

[34] M. Weis and F. Naumann, "Detecting Duplicates in Complex XML Data," in *ICDE*, 2006, p. 109.

[35] S. Chaudhuri, V. Ganti, and R. Motwani, "Robust Identification of Fuzzy Duplicates," in *ICDE*, 2005, pp. 865–876.

[36] H. H. Shahri and A. A. Barforoush, "A Flexible Fuzzy Expert System for Fuzzy Duplicate Elimination in Data Cleaning," in *DEXA*, 2004, pp. 161–170.

[37] H. Müller and J. Freytag, "Problems, methods, and challenges in comprehensive data cleansing," Humboldt Universität Berlin, Tech. Rep., 2003.

[38] A. de Keijzer, M. van Keulen, and Y. Li, "Taming data explosion in probabilistic information integration," http://eprints.eemcs.utwente.nl/7534/, Enschede, Technical Report TR-CTIT-06-05, February 2006.

[39] M. van Keulen and A. de Keijzer, "Qualitative Effects of Knowledge Rules and User Feedback in Probabilistic Data Integration," *The VLDB Journal*, vol. -, no. -, July 2009.