Proceedings of the 15th International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2015

Keys for the fusion of heterogeneous information

Benito-Picazo, F.[1], Cordero, P.[1], Enciso, M.[1] and Mora, A.[1]

[1] *Universidad de Málaga, Andalucia Tech, Spain*

emails: fbenito@lcc.uma.es, pcordero@uma.es, enciso@lcc.uma.es, amora@ctima.uma.es

Abstract

The management of heterogeneous information is a current topic which demands the use of intelligent techniques to deal with data semantics. In this work we approach this problem by using Simplification Logic. It has a sound and complete inference system conceived to treat implications and functional dependencies. The automatic processing of functional dependencies allows to develop methods and tools to tackle most classical problems in database and information processing. In this work, we use Simplification Logic to design a method to enumerate all minimal keys of a data repository inferring them from a set of functional dependencies. We also illustrates how this method provides a successful way to solve some outstanding problems in data processing in linked data.

Key words: Heterogenous information, integration, keys, logic.

1 Introduction

The notion of a key is fundamental to any data model, including Codd's relational model of data [Codd, 1970]. A key of a relation schema is composed by a subset of attributes playing the role of a *domain* in a given function whose *image* is the whole set of attributes. This way, the table is viewed as its extensional definition. These functions are described by means of a *Functional Dependency (FD)* which specifies a constraint between two subset of attributes, denoted $A \to B$, ensuring us that for any two tuples in a table, if they agree on A, they also agree on B.

Besides primary keys, unique constraints (candidate keys) provide a more complete understanding of the model. Keys are not only fundamental to data design, they are also considered powerful tools to solve several problems related to a lot of fields of information management as mentioned before. Indeed, in [Sismanis et al., 2006] the authors affirm

that "*identification of keys is a crucially important task in many areas of modern data management, including data modeling, query optimization (provide a query optimizer with new access paths that can lead to substantial speedups in query processing), indexing (allow the database administrator to improve the efficiency of data access via physical design techniques such as data partitioning or the creation of indexes and materialized views), anomaly detection, and data integration*".

Identifying properly the keys of a relation schema is a crucial task for a lot of modern areas of information management (data modeling [Simsion and Witt, 2005], query optimization [Kemper and Moerkotte, 1991], indexing [Manolopoulos et al., 1999], etc).

A very outstanding characteristic of keys is their minimality. We denote a key as minimal when every attribute contained in its attribute set is necessary to keep the property of key, i.e. keys with no superfluous attributes. Thus, in the literature, the key finding problem is focused on minimal keys. Giving an attribute set A, the cardinality of the set 2^A forces us to consider in applying special techniques that lead the search of the sets being candidates to become minimal keys.

Keys constraints specify the sets of attributes of a relation such that their projection univocally identifies each tuple of the relation. The key finding problem consists in finding all the attribute subsets which make up a minimal key as of a set of FDs occurring within a schema of a relational model table.

In Table 1, we illustrate its semantics by the following basic example.

Table 1: Movie table

Title	Year	Country	Director	Nationality	Star
Pulp Fiction	1994	USA	Quentin Tarantino	USA	John Travolta
Pulp Fiction	1994	USA	Quentin Tarantino	USA	Uma Thurman
Pulp Fiction	1994	USA	Quentin Tarantino	USA	Samuel L. Jackson
King Kong	2005	New Zealand	Peter Jackson	New Zealand	Naomi Watts
King Kong	2005	New Zealand	Peter Jackson	New Zealand	Jack Black
King Kong	1976	USA	De Laurentiis	IT	Jessica Lange
King Kong	1976	USA	De Laurentiis	IT	Jeff Bridges
Django Unchained	2012	USA	Quentin Tarantino	USA	Jamie Foxx
Django Unchained	2012	USA	Quentin Tarantino	USA	Samuel L. Jackson

From the information in Table 1, we may ensure that the following FDs are satisfied: $Title, Year \rightarrow Country$; $Title, Year \rightarrow Director$; $Director \rightarrow Nationality$. Moreover, the table has only one minimal key: $\{Title, Year, Star\}$

It is imperative to state that though we are dealing with FDs, it is not the responsibility of this work to extract them from a relation schema. There are several techniques that provide us this work [Huhtala et al., 1999], [Yao et al., 2002]. Therefore, we will begin with given sets of FDs and then go ahead finding minimal keys within them.

In the same way, it is necessary to clarify that what we are researching is not a matter of data mining techniques [Fayyad et al., 1996]. Giving an overall view, data mining could be considered as a computational process of discovering patterns in large data sets and its

goal goes to extract information from a data set and transform it into an understandable structure for further use [Witten et al., 2011]. Nevertheless, what we are studying and developing are mechanisms and algorithms for deriving all minimal keys so they can help us performing a more intelligent and efficient way to manage the information stored in relational schemes.

In this work we will concentrate our efforts on those algorithms guided by logic, and most specifically, those using Tableaux paradigm [Morgan, 1992] [Risch and Schwind, 1992] for deriving keys of a relation schema using inference systems.

The problem of the key finding methods arises when we try to deal with a huge amount of information since the tableaux's building mechanism produces such an explosion of the search space that we go beyond the machine capabilities, even with small problems.

2 Background

Before going further, we need to introduce the necessary terms from relational database theory. Due to space limitation, we refer those readers non familiar with the basic notions of FDs and relational databases to previously visit [Elmasri and Navathe, 2010]. The notion of FDs are well-known in other areas as Formal Concept Analysis with the concept of implications.

Definition 1 (Attribute) *An attribute a is an identifier for an element of some domain D. We use letters $a, b, c, d, ...$ for attributes. Let U be a set of attributes. An attribute set X over U is a subset of U. We use capital letters X, Y, Z, V, ... for attribute sets.*

Definition 2 (Functional dependency) *Let U be a set of attributes. A functional dependency (FD) over U is an expression of the form $X \to Y$, where X, Y are attribute sets. It is satisfied in a table R if for every two tuples of R, if they agree on X, then they agree on Y.*

Definition 3 (Relation schema) *A relation schema $R = \langle U, F \rangle$ is an ordered pair consisting of an attribute set U and a set F of FDs over U.*

A key of a relational table is a subset of attributes that allows us to uniquely characterize each row. It may be defined by means of FDs as follows:

Definition 4 (Key) *Given a table R over the set of attributes U, we say that K is a key in R if the functional dependency $K \to U$ holds in R.*

Definition 5 (Minimal Key) *Given the table R, the attribute set $K \subset U$ is said to be a minimal key if it is a key of R and for all attribute $k \in K$ the subset $K - \{k\}$ is not a key of R.*

 ISBN: 978-84-617-2230-3

3 Simplification Logic for finding-key problem

3.1 SL_{FD}

Now, we summarize the axiomatic system of SL_{FD} [Cordero et al., 2013]. The inference system for SL_{FD} is equivalent to the well-known Armstrong's axioms as the first complete inference system for functional dependencies. It avoids the use of transitivity and is guided by the idea of simplifying the set of FDs by removing redundant attributes efficiently.

SL_{FD} is defined as the pair (L_{FD}, S_{FD}) where S_{FD} has the following axiom scheme and inference rules. The third rule is named Simplification rule and it is the core of SL_{FD}:

$$[Axiom] \quad \frac{Y \subseteq X}{X \to Y}$$

$$[Frag] \quad \frac{X \to Y}{X \to Y'}, \quad [Comp] \quad \frac{X \to Y \quad U \to V}{XU \to YV}$$

$$[Simp] \quad \frac{X \to Y \quad U \to V}{(U - Y) \to (V - Y)'} \quad X \subseteq U, X \cap Y \neq \emptyset$$

3.2 SST Method

The method we will use to find all minimal keys in schemes provided from two heterogeneous sources is SST method (see [Cordero et al., 2014] for more details). The input of the tableaux method is a set of attributes Ω and a set of formulas F. We build a tree as follows:

1. The root of the tree will be $(\Omega \to \Omega, F)$.

2. For each node $(U \to \Omega, F)$ with $U \neq \emptyset$ and each minimal formula $A \to B \in F$, a new children node $(U' \to \Omega, F')$ is added where:

   - $U' \to \Omega$ is obtained applying [lSimp] to $A \to B$ and $U \to \Omega$. That is, $U' = A \cup (U \setminus B)$.
   - F' is computed by applying [sSimp] to $A \to B$ and every formula in F. Moreover, in the new set of formulas, Union equivalence is applied and degenerated formulas are removed.

3. The method renders $Minimal\{U | (U \to \Omega, \emptyset)$ is a leaf of the tree$\}$.

 ISBN: 978-84-617-2230-3

4 Applications of keys in Linked Data

There are a lot of fields of knowledge in computer science where counting on efficient techniques for data management is crucial. Generally, it is an engineering work to establish and choose the keys as a part of the normalization process of the schema. The challenge is to figure out those attributes of the schema that allow identifying univocally each tuple of the relation. Lets show an easy example.

Example 1 *Assume that we have stored in a table the main data of an enterprise crew such as: Name, Age, ID Number, Phone Number. A priori, we notice several alternatives in order to identify each employee.*

As a key we could consider: ID Number, the pair (Name, Phone Number) or even (ID Number, Name). However, from all of that possibilities, ID Number arises as the best one since the other ones contain information that is not absolutely necessary to identify each person. Two people with different names could share the same phone number (members of a same home will share the same phone number). Pairs (ID Number, Name) will also identify properly each person, but the Name attribute is not indispensable since ID Number will be definitely enough as identification.

So, even in this trivial example we can easily notice that deriving keys is a task with an indisputable importance. Lets go deeper in details with a bit more realistic problems.

Linked Data is about connecting pieces of related data and information coming from different sources. In computing, linked data describes a method of publishing structured data so that it can be interlinked and become more useful. It builds upon standard Web technologies such as HTTP, RDF [1] and URIs, but rather than using them to serve web pages for human readers, it extends them to share information in a way that can be read automatically by computers. This enables data from different sources to be connected and queried [Bizer et al., 2009].

A typical case of a large Linked Dataset is DBPedia, which, essentially, makes the content of Wikipedia available in RDF. The importance of DBPedia is not only that it includes Wikipedia data, but also that it incorporates links to other datasets on the Web, e.g., to Geonames. By providing those extra links (in terms of RDF triples) applications may exploit the extra (and possibly more precise) knowledge from other datasets when developing an application; by virtue of integrating facts from several datasets, the application may provide a much better user experience.

For instance, the DBpedia resource for Brussels (http://dbpedia.org/resource/Brussels) can be linked to the one maintained by the Statistics Belgium

[1] RDF, the Resource Description Framework, is one of the key ingredients of Linked Data, and provides a generic graph-based data model for describing things, including their relationships with other things. RDF data can be written down in a number of different ways, known as serialisations. Examples of RDF serialisations include RDF/XML, Notation-3 (N3), Turtle, N-Triples, RDFa, and RDF/JSON.

(http://location.testproject.eu/so/au/AdministrativeUnit/STATBEL/24000).

Linking these two data resources allows us to get richer information about Brussels. Besides that, the attributes of the data entity for the city of Brussels is country. This attribute reveals that a city is positioned in/belongs to a country. In our case the value for country is Belgium. There are different options for encoding this information.

One way would be to include the value for country as text, e.g. a literal or a string. This option however cannot take us too far and can suffer from different writings, different languages and even spelling errors. The Linked Data approach in this case opts for replacing the text value with a URI pointing to the specific country, i.e. to Belgium (the URI of DBpedias resource for Belgium is http://dbpedia.org/resource/Belgium). The Linked Data option allows us to unambiguously refer to Belgium and also navigate through the links in order to collect more information about Brussels.

Due to these considerations, we need to find out the minimal set of properties necessary to make up the appropriate connections between the data in both schemes. Therefore, we need the minimal keys from each one of the sets so we can decide later which element will be our joining one.

In the following, we outline how the heterogeneous information is linked in two dataset with information about films.

Example 2 *The repositories* `http://www.imdb.com` *and* `https://www.filmaffinity.com` *contain information about films with different structure. The goal is to obtain the keys in order to connect the knowledge stored in both datasets.*

The method to make the fusion of information is summarized in the next items:

- *To select a group of films.*
- *To take the topics from* IMDB repository*.*
- *To extract the implications using the tool for Formal Concept Analysis named* ConceptExplorer*.*
- *To calculate the minimal keys for IMBD schema.*
- *To take the topics from* Filmafinity repository*.*
- *To extract the implications using the tool for Formal Concept Analysis named* ConceptExplorer*.*
- *To calculate the minimal keys for Filmafinity schema.*
- *To repeat the same with the joining of both implication sets.*

The result of applying SST Method to this two heterogeneous dataset is the following:

```
 IMDB
Topics: Action Comedy Crime Drama Fantasy Romance Thriller

Little City (1998) Comedy Romance
Driver, The (1978)  Action Crime Thriller
Father of the Bride (1950) Comedy Romance
Bio-Dome (1996) Comedy
Fast Runner, The (2001) Drama Fantasy
Overboard (1987) Comedy Romance
Get Rich or Die Tryin? (2005) Crime Drama

Implications:
Romance -> Comedy
Thriller -> Action Crime
Action -> Crime Thriller
Keys:
Romance Thriller
Romance Action

Filmaffinity
Topics: Action Comedy Crime Drama Family Film-noir Nature Romance Survival

Little City (1998) Comedy Drama
Driver, The (1978)  Action Crime Film-noir
Father of the Bride (1950) Comedy Romance Family
Bio-Dome (1996) Comedy
Fast Runner, The (2001) Drama Nature Survival
Overboard (1987) Comedy
Get Rich or Die Tryin? (2005) Drama

Implications:
Action -> Crime Filmnoir
Crime -> Action Filmnoir
Family -> Comedy Romance
Filmnoir -> Action Crime
Romance -> Comedy Family
Survival -> Drama Nature
Nature -> Drama Survival
```

 ISBN: 978-84-617-2230-3

```
Keys:
Action Family Survival
Action Family Nature
Action Romance Survival
Action Romance Nature
Family Filmnoir Survival
Family Filmnoir Nature
Filmnoir Romance Survival
Filmnoir Romance Nature
Crime Family Survival
Crime Family Nature
Crime Romance Survival
Crime Romance Nature

The joining of both topics:
Topics: Action Comedy Crime Drama Family Fantasy Film-noir Nature Romance Survival Thrill

Little City (1998) Comedy Drama Romnace
Driver, The (1978)  Action Crime Film-noir Thriller
Father of the Bride (1950) Comedy Romance Family
Bio-Dome (1996) Comedy
Fast Runner, The (2001) Drama Fantasy Nature Survival
Overboard (1987) Comedy Romance
Get Rich or Die Tryin? (2005) Crime Drama

Implications:
Comedy Drama ->  Romance
Family -> Comedy Romance
Fantasy ->  Drama Nature Survival
Nature ->  Drama Fantasy Survival
Romance ->  Comedy
Survival ->  Drama Fantasy Nature
Thriller ->  Action Crime Filmnoir
Action ->  Crime Filmnoir Thriller
Filmnoir ->  Action Crime Thriller

Keys:
Family Filmnoir Fantasy
Family Action Fantasy
```

 ISBN: 978-84-617-2230-3

```
Family Thriller Fantasy
Family Filmnoir Nature
Family Action Nature
Family Thriller Nature
Family Filmnoir Survival
Family Action Fantasy
Family Thriller Survival
```

5 Conclusions

The fusion of heterogeneous information is an emergent problem in which the use of Logic is adequate in order to incorporate automated reasoning mechanism. We have proposed the use of Simplification Logic to manipulate functional dependencies (implications).

SST Method is based of Simplification Logic and allow us to enumerate all minimal keys of a data repository inferring them from a set of functional dependencies. We illustrate how the method can be used to solve problems related with the integration/fusion of heterogeneous information in linked data.

For a extension of this work, we are envolved in the introduction of parallelism techniques in the implementation of the algorithms that could help us dealing with problems containing a substantial amount of information at the input. We think that the tableaux paradigm used in logic matches perfectly the design of the parallel versions of the algorithms. This strategy could be used in order to resolve complex input problems. We truly need to go further searching for strategies in order to find out good BOVs for the partial implementation process.

Besides, it is necessary to go ahead in the design of a *benchmark* that takes into account the different aspects and nature of these problem and algorithms in order to direct us searching the best strategies.

Acknowledgements

Supported by grant TIN11-28084 of the Science and Innovation Ministry of Spain.

References

[Bizer et al., 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22.

[Codd, 1970] Codd, E. F. (1970). A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387.

[Cordero et al., 2013] Cordero, P., Enciso, M., and Mora, A. (2013). Automated reasoning to infer all minimal keys. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 817–823. AAAI Press.

[Cordero et al., 2014] Cordero, P., Enciso, M., Mora, A., and de Guzmán, I. P. (2014). A tableaux-like method to infer all minimal keys. *Logic Journal of the IGPL*, 22(6):1019–1044.

[Elmasri and Navathe, 2010] Elmasri, R. and Navathe, S. (2010). *Fundamentals of Database Systems*. Prentice Hall International, 6 edition.

[Fayyad et al., 1996] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, pages 37–54.

[Huhtala et al., 1999] Huhtala, Y., Krkkinen, J., Porkka, P., and Toivonen, H. (1999). Tane: An efficient algorithm for discovering functional and approximate dependencies. *Comput. J.*, 42(2):100–111.

[Kemper and Moerkotte, 1991] Kemper, A. and Moerkotte, G. (1991). Query optimization in object bases: Exploiting relational techniques. In *Query Processing for Advanced Database Systems, Dagstuhl*, pages 63–98. Morgan Kaufmann.

[Manolopoulos et al., 1999] Manolopoulos, Y., Theodoridis, Y., and Tsotras, V. J. (1999). *Advanced Database Indexing*, volume 17 of *Advances in Database Systems*. Kluwer.

[Morgan, 1992] Morgan, C. G. (1992). An automated theorem prover for relational logic (abstract). In Fronhfer, B., Hhnle, R., and Kufl, T., editors, *TABLEAUX*, pages 56–58.

[Risch and Schwind, 1992] Risch, V. and Schwind, C. (1992). Tableaux-based theorem proving and non-standard reasoning. In Fronhfer, B., Hhnle, R., and Kufl, T., editors, *TABLEAUX*, pages 76–78.

[Simsion and Witt, 2005] Simsion, G. C. and Witt, G. C. (2005). *Data modeling essentials*. Amsterdam; Boston, 3rd edition.

[Sismanis et al., 2006] Sismanis, Y., Brown, P., Haas, P. J., and Reinwald, B. (2006). Gordian: efficient and scalable discovery of composite keys. In *In Proc. International Conference on Very Large Data Bases (VLDB*, pages 691–702.

[Witten et al., 2011] Witten, I., Frank, E., and Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

[Yao et al., 2002] Yao, H., Hamilton, H. J., and Butz, C. J. (2002). Fdmine: Discovering functional dependencies in a database using equivalences. In *ICDM*, pages 729–732. IEEE Computer Society.

 ISBN: 978-84-617-2230-3