Information Systems 35 (2010) 352–374

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Empirical evidence for the usefulness of Armstrong relations in the acquisition of meaningful functional dependencies

Warren-Dean Langeveldt, Sebastian Link *

School of Information Management, The Victoria University of Wellington, Wellington, New Zealand

A R T I C L E   I N F O

Article history:
Received 2 November 2009
Received in revised form
22 November 2009
Accepted 23 November 2009
Recommended by: D. Shasha

Keywords:
Knowledge acquisition
Armstrong relation
Functional dependency
Soundness
Completeness

A B S T R A C T

Armstrong relations satisfy precisely those data dependencies that are implied by a given set of data dependencies. A common perception is that Armstrong relations are useful in the acquisition of data semantics, in particular since errors during the requirements elicitation have the most expensive consequences.

We report on some first empirical evidence for this perception regarding the class of functional dependencies (FDs). For this purpose, we investigate the usefulness of Armstrong relations with respect to various measures. Soundness measures how many of the as meaningful perceived FDs are actually meaningful. Completeness measures how many of the actually meaningful FDs are also perceived as meaningful.

Our experiment determines what and how much design teams learn about the application domain in addition to what they know prior to using Armstrong relations. The data analysis suggests that in using Armstrong relations it is not more likely to recognize meaningless FDs which are incorrectly perceived as meaningful, but it is more likely to recognize meaningful FDs that are incorrectly perceived as meaningless.

Our measures assess the quality of an FD set with respect to a target FD set, and therefore qualify naturally for the use in automated assessment tools, e.g. for database course exams or assignments.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Armstrong relations are of interest in database theory and practice. Let $\Sigma \cup \{\varphi\}$ denote a set of functional dependencies (FDs). We say that Σ *implies* φ, if every relation that satisfies every FD in Σ also satisfies φ. That is, there is no counterexample relation that satisfies all FDs in Σ and violates φ. We write $\Sigma \models \varphi$ to denote that Σ implies φ (and $\Sigma \not\models \varphi$ to denote that Σ does not imply φ). For a set Σ of FDs, let Σ^* denote the set of all FDs implied by Σ. For every FD φ that is not in Σ^*, there is a counterexample relation r_φ that satisfies all FDs in Σ and violates φ. As a consequence of a result by Armstrong [1], there is a single counterexample relation that satisfies all FDs in Σ^* and violates all FDs not in Σ^*. Following common terminology we call such a relation an Armstrong relation for Σ. The following example illustrates the potential benefits of utilizing Armstrong relations for the acquisition of meaningful FDs.

Let us assume that in developing an information system for some manufacturer of electrical goods we identify the processing of orders by retail sellers as a domain of interest. In particular, we define the relation schema ORDER that consists of the attributes *Order#*, *Product#*, *Description*, *Qty* and *Total*. These show for an order (identified by its order number *Order#*), a product in that order (identified by its unique product number *Product#*), a description *Description* of that product, the quantity *Qty* of that product in that order, and the total value *Total* (in some fixed currency) of that product in that order.

* Corresponding author. Tel.: +64 44636813; fax: +64 44635446.
E-mail address: sebastian.link@vuw.ac.nz (S. Link).

0306-4379/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2009.11.002

Table 1
An Armstrong relation for the empty FD set.

Order#	Product#	Description	Qty	Total
00723	389	Microwave	10	5000
00724	389	Microwave	10	5000
00724	521	Microwave	10	5000
00724	521	Oven	10	5000
00724	521	Oven	20	5000
00724	521	Oven	20	8000

Suppose the designers of our information system have not been able yet to identify any meaningful FDs for the schema Order, i.e., $\Sigma=\emptyset$. Therefore, they decide to inspect a relation that faithfully represents the initial design draft of an empty FD set. The relation they decide to examine is the one in Table 1. This relation is Armstrong for the empty FD set Σ.

By inspecting the Armstrong relation the designers simply notice that the *Oven* with Product# *521* is associated with the different quantities of *10* and *20* in the order with Order# *00724*. This observation causes the design team to specify the FD

$$\text{Order}\sharp, \text{Product}\sharp \rightarrow \text{Qty}$$

which states that the schema Order records a unique quantity for the same product in the same order. A similar observation causes the design team to specify the FD

$$\text{Order}\sharp, \text{Product}\sharp \rightarrow \text{Total}$$

which states that the order number and the product number together uniquely determine the total of the product in the order. Moreover, the design team observes that the product with Product# *521* has two different descriptions *Microwave* and *Oven*. This observation causes the design team to ask the domain experts whether different descriptions can be given to any product. Since the experts agree that this cannot be the case, the design team responds by specifying the FD

$$\text{Product}\sharp \rightarrow \text{Description}$$

which states that the description of a product is uniquely determined by the product number. We can see that, by inspecting the Armstrong relation above, the designers have successfully identified three meaningful FDs for the application domain. Furthermore, these three FDs together imply the FD

$$\text{Order}\sharp, \text{Product}\sharp \rightarrow \text{Description}, \text{Qty}, \text{Total}.$$

Therefore, the design team recommends the attribute set {Order#, Product#} as a candidate key for the schema Order.

In general, a relation that satisfies an FD set Σ but which is not Armstrong for Σ will satisfy some FD that is not in Σ^*. Therefore, relations that are not Armstrong for a given FD set may not be able to reveal problems with the current design. For example, the relation in Table 2 is not Armstrong for the empty FD set Σ. While this relation satisfies Σ (as every other relation does in this case), it gives the false impression that the current design, i.e. $\Sigma=\emptyset$, is acceptable. Specifically, the relation is not a faithful representation of the FD set Σ. For example, the relation does not violate the FD Order#, Product$\sharp\rightarrow$Qty, nor the FD Order#, Product$\sharp\rightarrow$Total, nor does it violate the FD Product$\sharp\rightarrow$Description, even though they are not in Σ^*. Intuitively, an inspection of the relation in Table 2 does neither seem to encourage a design team to specify the FDs

Table 2
A relation not Armstrong for the empty FD set.

Order#	Product#	Description	Qty	Total
00723	389	Microwave	10	5000
00724	521	Oven	20	8000

$$\text{Order}\sharp, \text{Product}\sharp \rightarrow \text{Qty},$$

$$\text{Order}\sharp, \text{Product}\sharp \rightarrow \text{Total}$$

nor does it seem to encourage the team to ask the domain experts whether different descriptions can be associated with the same product number.

This simple example illustrates the potential benefit of using Armstrong relations in the process of identifying the complete set of FDs that are meaningful for the underlying application domain. Failure to identify such a complete set means that the output of the requirements analysis is afflicted with errors.

Empirical studies show that more than half the errors which occur during systems development are requirements errors [2–4]. Requirements errors are also the most common cause of failure in systems development projects [2,5,6]. The cost of errors increases exponentially over the development life cycle: it is more than 100 times more costly to correct a defect post-implementation than it is to correct it during requirements analysis [7]. This suggests that it would be more effective to concentrate quality assurance efforts in the requirements analysis stage, in order to catch requirements errors as soon as they occur, or to prevent them from occurring altogether [8]. Hence, Armstrong relations appear to be a valuable tool for the requirements analysis of the target database. However, the question remains in what precise sense they are valuable.

Research gap and research questions: In previous work, Armstrong relations were called "user-friendly representations" of sets of data dependencies [9], and it was stated that they are "useful for database design" [9,10]. However, the phrase "useful for database design" was exclusively justified in terms of the structural and algorithmic properties of Armstrong relations. For instance, this may refer to the fact that FDs enjoy Armstrong relations, i.e., for every set Σ of FDs there is an Armstrong relation for Σ. Note that it is everything but self-evident that a given class of data dependencies enjoys Armstrong relations [11]. Other interpretations of "useful" may refer to either the size of an Armstrong relation for an FD set Σ, e.g. the minimal number of tuples required for a relation to be Armstrong for Σ, or the existence/efficiency of algorithms to compute such an Armstrong relation. These

Download English Version:

https://daneshyari.com/en/article/396745

Download Persian Version:

https://daneshyari.com/article/396745

Daneshyari.com