On the Interaction Between ISA and Cardinality Constraints *

D. Calvanese, M. Lenzerini
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
e-mail: {calvanese,lenzerini}@assi.dis.uniroma1.it

Abstract

ISA and cardinality constraints are among the most interesting types of constraints in data models. ISA constraints are used to establish several forms of containment among classes, and are receiving great attention in moving to object-oriented data models, where classes are organized in hierarchies based on a generalization/specialization principle. Cardinality constraints impose restrictions on the number of links of a certain type involving every instance of a given class, and can be used for representing several forms of dependencies beteewn classes, including functional and existence dependencies. While the formal properties of each type of constraints are now well understood, little is known of their interaction. The goal of this paper is to present an effective method for reasoning about a set of ISA and cardinality constraints in the context of a simple data model based on the notions of classes and relationships. In particular, the method allows one both to verify the satisfiability of a schema and to check whether a schema implies a given constraint of any of the two kinds. We prove that the method is sound and complete, thus showing that the reasoning problem for ISA and cardinality constraints is decidable.

1 Introduction

Integrity constraints are used to express conditions that the elements of the database (objects, tuples, values, etc.) must satisfy in order to plausibly represent the underlying domain of interest. There are at least three major problems when dealing with integrity constraints, namely: (a) how to express them, (b) how to use them during schema construction, and (c) how to ensure that they are satisfied by the database. In this paper we are mainly concerned with problem (b).

Two main approaches to problem (a) have been advocated: either using a very general language to express, in principle, all kinds of constraints, or singling out interesting classes of constraints, and adding corresponding ad hoc constructs to the data model.

Problem (b) is related to the idea of reasoning about a collection of constraints in order to derive relevant properties during database schema design. Notable examples of such properties are satisfiability (or consistency) and implication. A set of integrity constraints are said to be satisfiable if there exists at least one database state that satisfies them. A set of constraints is said to imply a given constraint if every database state that satisfies the former satisfies the latter too. Notice that when adopting a general language for expressing constraints, problem (b) becomes very hard: the more powerful the constraint language, the more difficult to devise suitable reasoning methods. On the contrary, using ad hoc constructs for expressing specific classes of constraints often makes it possible to develop specialized reasoning methods, as demonstrated by the large body of research in dependency theory within the relational model (see [1]).

The work reported in this paper has been carried out in the context of the second approach to problem (a) (ad hoc constructs for special classes of constraints), and is concerned with the problem of reasoning about a set of constraints belonging to interesting classes. In particular, we deal with two classes of constraints, namely, ISA and cardinality ratio constraints.

ISA constraints are used to establish several forms of containment between the instances of two or more classes of objects in a database schema. They have always received great attention both in database modeling and in knowledge representation, since they are the basics for representing both general-

*This work has been partially supported by the Italian Research Council (CNR), Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo, Sottoprogetto 7, Linea di ricerca IBRIDI, and by the EEC Esprit Basic Research Action N.6810 (COMPULOG 2).

ization/specialization abstractions and inheritance of properties. Recently, they are becoming even more important in moving to object-oriented data models, where classes are organized in hierarchies based on a containment relation. Formal properties of this type of constraints have been the subject of a lot of research efforts in various contexts, ranging from the relational model to conceptual and semantic data models (see for example [2, 3]).

Cardinality ratio constraints (or simply cardinality constraints) are used to impose restrictions on the number of links of a certain type involving every instance of a given class. They appear in many forms in most of the database models proposed in the last two decades. Functional and existence dependencies are examples of cardinality constraints in the relational model. In [4], a binary data model, that inspired most of the conceptual models subsequently proposed, is presented where cardinality constraints can be stated for binary relationships. Cardinality constraints have always been considered as natural constraints to express in Entity-Relationship (ER) schemas. For example, the ER model proposed in [5, 6, 7, 8] as the target model used in the phase of conceptual database design, includes cardinality constraints in various ways. In [9], a relationship mechanism, together with a form of cardinality constraints, is introduced in the context of the object oriented data model. The use of cardinality constraints is also very common in structured knowledge representation languages (semantic networks and frame-based languages), as pointed out, for example, in [10].

Formal properties of cardinality constraints are studied in several papers. We already mentioned the work on functional and existence constraints in the relational theory (see [11]). In [12], a general form of cardinality constraints (called numerical dependencies), which limit the maximum number of associations between two given sets of attributes, is analyzed in the context of the relational model. Due to the expressive power of such constraints, it turns out that in the general case there is no finite set of inference rules for reasoning about them. In [13], a uniform framework is proposed for expressing cardinality constraints in the ER model, later extended in [14]. In [15], a method is proposed for reasoning about an ER model with cardinality constraints, but without any form of inclusion dependencies. Finally, cardinality constraints in the context of knowledge representation languages are studied in [16, 17, 18].

It is interesting to observe that, while each type of constraints mentioned above is now quite well understood, we know very little of the interaction between ISA and cardinality constraints. In particular, a sound and complete method for satisfiability and implication checking in the presence of constraints of the two kinds is still missing.

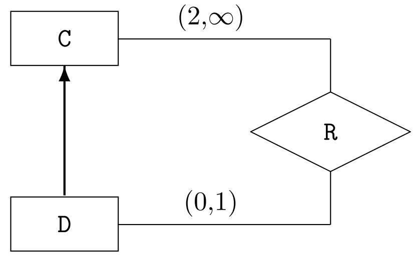

Figure 1: A finitely unsatisfiable ER-diagram

Notice that reasoning about a schema comprising both types of constraints appears to be significantly harder than reasoning about the two types of constraints separately. It is clear, for example, that the interaction between ISA and cardinality constraints may cause a database schema to exhibit undesirable properties. In particular, it may happen that there exists a class in the schema that is necessarily empty (i.e. has no instances) in all finite database states. We show an example of this in Figure 1, where the schema is expressed in the ER notation (classes of objects as boxes and relationships among classes as diamonds). The cardinality constraints in the schema (represented by pairs (min-card, max-card) associated with the connections between classes and relationships) force the number of instances of class `D` to be twice the number of instances of class `C`, while the inclusion dependency (represented by the arrow) forces the extension of `D` to be a subset of the extension of `C`. Obviously, this schema admits no finite database state.

The goal of this paper is to present an effective method for reasoning about a set of ISA and cardinality constraints in the context of a data model based on the notions of classes and relationships. In particular, the method allows one both to verify the satisfiability of a schema and to check whether a schema implies a given constraint of any of the two kinds. We show that the method is sound and complete, thus proving that the reasoning problem for ISA and cardinality constraints is decidable.

Our decision procedure is based on the idea of representing the cardinality constraints in terms of a special system of linear disequations, similarly to what is done in [15]. However, the presence of ISA constraints makes the problem much harder than the one addressed in [15], and requires a different technique for deriving a suitable system of disequations from the schema.

The result presented in this paper provides a solution to a long standing problem in the ER model. However, we point out that the method is not merely confined to this model. Indeed, it is possible to show that we can specialize our technique in order to deal with most of the data models proposed in the literature. For example, by interpreting relationships as attributes, we directly derive a method applicable to object oriented data models. Similarly, by interpreting classes as frames and relationships as slots, we obtain a corresponding decision procedure for several knowledge representation formalisms. We have chosen an abstract simple data model simply because it offers the basic notions for studying the interaction between ISA and cardinality constraints, which is the main focus of our investigation.

The paper is organized as follows. In Section 2, we describe the basic characteristics of the data model we use. In Section 3 we present the method for checking the satisfiability of a schema expressed in this model. In Section 4 we show how to exploit the method in order to check implication. Finally, Section 5 concludes the paper with some remarks about both the proposed method and future related research.

2 The data model

In this section we describe the main characteristics of the data model we use in our investigation, called CR. The basic concepts in this model are those of class and relationship (see [19, 5]).

A *class* denotes a set of individuals, called its instances, representing objects of the domain of interest with common properties.

A *relationship* represents associations between individuals. An instance of a relationship denotes an association between instances of classes connected to the relationship. Since each class can be connected to a relationship more than once, the concept of *role* is introduced to distinguish between different connections of the same class to a relationship.

More precisely, we associate to each relationship R a set of roles, denoted with $role(R)$, such that each role is specific for one relationship. For each relationship R, a particular class is associated to each role $U \in role(R)$. This class is called the *primary class* for U in R. Intuitively, the fact that C is the primary class for U in R means that the U-component of every instance of R is an instance of C.

In the proposed model we make use of the following types of integrity constraints:

- *ISA constraints* between classes, also called *is-a relationships*, force the set of instances of a class to be contained in the set of instances of another class. This implies that the contained class inherits all relevant properties of all its ancestors, including possible connections to relationships and cardinality constraints.

- *Cardinality constraints* specify for a class C, for a relationship R and for a role $U \in role(R)$, the minimum and maximum number of instances of R having the same instance of C in the role U.

We show in Figure 2 an example of CR-schema (represented as diagram, where classes are boxes and relationships are connected to their primary classes), where we model a meeting constituted by a number of talks. For each talk there is exactly one speaker and at least one discussant. Each discussant can participate to at most one talk, whereas there is no limitation on the number of talks a speaker can hold. We further require that each discussant must also be a speaker. Since we do not want to exhaust speakers that are also discussants, they are not allowed to hold more than two talks. This last property is modeled by means of the cardinality constraints between the class `Discussant` and the relationship `Holds`, which are a refinement of the cardinality constraints between `Speaker` (the primary class for `Holds` in the role $\mathtt{U}_1$) and `Holds`. This kind of refinement is represented by means of the dashed edge connecting `Discussant` and `Holds`, labeled with the new constraints.

We observe that the possibility of associating to a class a refinement of the cardinality constraints defined for the superclasses, greatly enhances the expressive power of the model, and is coherent with the notion of property refinement in those data models supporting inheritance.

The following definition formalizes the notion of CR-schema.

Definition 2.1 *A* CR-schema $\mathcal{S}$ *is constituted by:*

- *a set* $\mathcal{C}$ *of* class *symbols, a set* $\mathcal{R}$ *of* relationship *symbols and a set* $\mathcal{U}$ *of* role *symbols*

- *a set* $\mathcal{S}_{isa}$ *of statements of the form* $C_1 \preceq C_2$*, where* C_1 *and* C_2 *are classes. The reflexive transitive closure of* $\preceq$ *is denoted with* $\preceq^*$ *;*

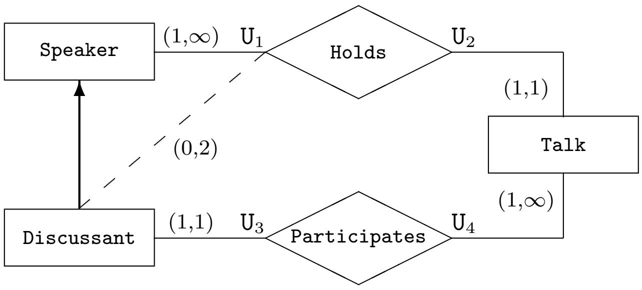

Figure 2: A CR-diagram

- *for each relationship $R \in \mathcal{R}$, a set of roles, denoted by $role(R)$, and for each role $U \in role(R)$ an associated class, called the* primary class for U in R. *The cardinality of $role(R)$ is greater than or equal to 2, and implicitly determines the arity of R. We assume that each role is specific for one relationship, i.e. for any two relationships R and R', $role(R) \cap role(R') = \emptyset$;*
- *for each relationship $R \in \mathcal{R}$, for each role $U \in role(R)$ and for each class $C \in \mathcal{C}$ such that $C \preceq^* C_U$, where C_U is the primary class for U in R, a non negative integer, $minc(C, R, U)$, and a non negative integer or ∞, $maxc(C, R, U)$. If not stated otherwise, $minc(C, R, U)$ is assumed to be 0 and $maxc(C, R, U)$ is assumed to be ∞.*

We define a *labeled tuple* over a generic set $\mathcal{D}$ as a function from a subset of $\mathcal{U}$ to $\mathcal{D}$, denoted as $\langle U_1 \colon d_1, \ldots, U_k \colon d_k \rangle$, where $U_i \in \mathcal{U}$ and $d_i \in \mathcal{D}$, for $i = 1, \ldots, k$. If $T = \langle U_1 \colon d_1, \ldots, U_k \colon d_k \rangle$ is a labeled tuple, we write $T[U_i]$ to denote d_i, where $i \in 1..k$. Notice that a relationship R can be characterized by a labeled tuple over the set $\mathcal{C}$ of classes. We will write $R = \langle U_1 \colon C_1, \ldots, U_K \colon C_K \rangle$ to specify that R is a relationship with $role(R) = \{U_1, \ldots, U_K\}$ and that C_k is the primary class for U_k in R, where $k = 1, \ldots, K$.

Figure 3 shows the schema corresponding to the CR-diagram of Figure 2.

An instance of a CR-schema, corresponding to the notion of database state, is a finite collection of instances of the involved classes and relationships, and is considered acceptable if it satisfies a set of rules inherent in the model and all integrity constraints that are part of the schema. More formally, the semantics of a CR-schema is based on the notion of interpretation. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a set $\Delta^{\mathcal{I}}$ (the *domain* of $\mathcal{I}$), and a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$), that maps

$\mathcal{C} = \{\texttt{Speaker}, \texttt{Discussant}, \texttt{Talk}\}$;
$\mathcal{R} = \{\texttt{Holds}, \texttt{Participates}\}$;
$\mathcal{U} = \{\texttt{U}_1, \texttt{U}_2, \texttt{U}_3, \texttt{U}_4\}$;

$\mathcal{S}_{isa} = \{\texttt{Discussant} \preceq \texttt{Speaker}\}$;

$\texttt{Holds} = \langle \texttt{U}_1 \colon \texttt{Speaker}, \texttt{U}_2 \colon \texttt{Talk} \rangle$;
$\texttt{Participates} = \langle \texttt{U}_3 \colon \texttt{Discussant}, \texttt{U}_4 \colon \texttt{Talk} \rangle$;

$minc(\texttt{Speaker}, \texttt{Holds}, \texttt{U}_1) = 1$;
$maxc(\texttt{Discussant}, \texttt{Holds}, \texttt{U}_1) = 2$;
$minc(\texttt{Talk}, \texttt{Holds}, \texttt{U}_2) = 1$;
$maxc(\texttt{Talk}, \texttt{Holds}, \texttt{U}_2) = 1$;
$minc(\texttt{Discussant}, \texttt{Participates}, \texttt{U}_3) = 1$;
$maxc(\texttt{Discussant}, \texttt{Participates}, \texttt{U}_3) = 1$;
$minc(\texttt{Talk}, \texttt{Participates}, \texttt{U}_4) = 1$.

Figure 3: The CR-schema corresponding to the CR-diagram shown in Figure 2

- every class $C \in \mathcal{C}$ to a subset $C^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and
- every relationship $R \in \mathcal{R}$ to a set $R^{\mathcal{I}}$ of labeled tuples over $\Delta^{\mathcal{I}}$.

The elements of $C^{\mathcal{I}}$ and $R^{\mathcal{I}}$ are called *instances* of C and R respectively.

Definition 2.2 *An interpretation $\mathcal{I}$ of a CR-schema $\mathcal{S}$, is said to be a* model of $\mathcal{S}$ *if it satisfies the following conditions:*

(A) For each statement $C_1 \preceq C_2 \in \mathcal{S}_{isa}$ it holds that $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$.

(B) For each relationship $R = \langle U_1 \colon C_1, \ldots, U_K \colon C_K \rangle \in \mathcal{R}$, all instances of R are of the form $\langle U_1 \colon \tilde{c}_1, \ldots, U_K \colon \tilde{c}_K \rangle$, where $\tilde{c}_k \in C_k^{\mathcal{I}}$, for $k = 1, \ldots, K$.

(C) For each relationship $R = \langle U_1 \colon C_1, \ldots, U_K \colon C_K \rangle \in \mathcal{R}$, for $k = 1, \ldots, K$, for each class $C \in \mathcal{C}$ such that $C \preceq^* C_k$ and for each instance $\tilde{c}$ of C in $\mathcal{I}$, it holds that

$$minc(C, R, U_k) \leq \left| \left\{ \tilde{r} \in R^{\mathcal{I}} \;\middle|\; \tilde{r}[U_k] = \tilde{c} \right\} \right| \leq$$

$$\leq maxc(C, R, U_k),$$

where $|\cdot|$ denotes the cardinality of a set.

3 Satisfiability in CR

A CR-schema is said to be *satisfiable* if it has a nonempty model, and is called *finitely satisfiable* if it has a finite model, i.e. a model with finite domain. Since in both databases and knowledge representation we are generally interested in finite models, in the rest of the paper we refer only to finite satisfiability, and we simply use the term satisfiability for that. Moreover, when we talk about interpretations and models, we implicitly refer to finite interpretations and finite models.

As pointed out in [15], the notion of satisfiability is not sufficient for capturing the relevant properties of a CR-schema. In fact, it is easy to see that every schema is satisfied by any interpretation that assigns an empty set of instances to every class and every relationship. It may happen, however, that the cardinality constraints or the interaction of cardinality constraints and is-a relationships forces a class to be empty in every interpretation. An example of this was shown in Figure 1. The above observation leads us to introduce the concept of class satisfiability, intended to capture the intuition that we should be able to populate a class in a schema without violating any of the constraints. In particular, a class C is said to be *satisfiable* in a CR-schema $\mathcal{S}$, if $\mathcal{S}$ admits a model $\mathcal{I}$ such that $C^{\mathcal{I}}$ is nonempty.

In this section we present a method for verifying the satisfiability of a class C in a CR-schema. Following [15], we model the cardinality constraints that appear in a CR-schema $\mathcal{S}$ by means of an associated system $\Psi_{\mathcal{S}}$ of linear disequations. The system is defined in such a way that the existence of a model of $\mathcal{S}$ where C is nonempty is reflected into the existence of particular solutions of $\Psi_{\mathcal{S}}$.

The unknowns in $\Psi_{\mathcal{S}}$ are intended to represent the number of instances of each class and each relationship in a possible model of $\mathcal{S}$, while the disequations take into account the constraints on the number of instances deriving from the cardinality constraints in $\mathcal{S}$. Unfortunately, because of classes that may have common instances, it is not possible to simply use one unknown for each class and relationship, as done in [15]. We will overcome the problem by introducing the notion of expansion of a CR-schema.

The rest of the section is structured as follows. In paragraph 3.1 we introduce the notion of expansion of a CR-schema; in paragraph 3.2 we use this notion to derive a system of linear disequations with the desired properties; finally, in paragraph 3.3 we present a method for deciding the satisfiability of a class in a CR-schema.

3.1 Expansion of a CR-schema

In order to define the expansion of a CR-schema $\mathcal{S}$, we introduce the notions of compound class and compound relationship (relative to $\mathcal{S}$).

A *compound class* $\bar{C}$ is a nonempty subset of $\mathcal{C}$. Intuitively it represents those individuals that are instances of all classes in $\bar{C}$ and are not instances of all classes in $\mathcal{C} \setminus \bar{C}$ [1]. A compound class $\bar{C}$ is said to be *consistent* if for any two classes $C_1, C_2 \in \mathcal{C}$ such that $C_1 \preceq C_2 \in \mathcal{S}_{isa}$, $C_1 \in \bar{C}$ implies $C_2 \in \bar{C}$.

A *compound relationship* is a labeled tuple over the set of compound classes. In particular, if $R = \langle U_1\colon C_1, \ldots, U_K\colon C_K \rangle \in \mathcal{R}$ is a relationship, a *compound relationship corresponding to* R is a labeled tuple of the form $\langle U_1\colon \bar{C}_1, \ldots, U_K\colon \bar{C}_K \rangle_R$, where $\bar{C}_1, \ldots, \bar{C}_K$ are compound classes. A compound relationship is *consistent* if all compound classes associated to its roles are consistent and contain the primary class for their specific role. A consistent compound relationship represents explicitly for each role the association to a consistent compound class that contains the primary class for that role. With $\bar{\mathcal{R}}_R$ we denote the set of all compound relationships $\langle U_1\colon \bar{C}_1, \ldots, U_K\colon \bar{C}_K \rangle_R$, obtained by associating in all possible ways to each role $U_k \in role(R)$ a compound class $\bar{C}_k$.

It is easy to see that whether a compound class or a compound relationship is consistent can be checked in polynomial time with respect to its size and the size of $\mathcal{S}_{isa}$.

Definition 3.1 *The* expansion $\bar{\mathcal{S}}$ *of a CR-schema* $\mathcal{S}$ *is constituted by:*

- *the set of all compound classes, denoted with* $\bar{\mathcal{C}}$*, and its subset* $\bar{\mathcal{C}}_c$ *of consistent compound classes*
- *the set of all compound relationships, denoted with* $\bar{\mathcal{R}}$*, obtained as the union of all* $\bar{\mathcal{R}}_R$*, where* $R \in \mathcal{R}$*, and its subset* $\bar{\mathcal{R}}_c$ *of consistent compound relationships*
- *for each relationship* $R = \langle U_1\colon C_1, \ldots, U_K\colon C_K \rangle \in \mathcal{R}$*, for* $k = 1, \ldots, K$*, and for each consistent compound class* $\bar{C} \in \bar{\mathcal{C}}_c$ *that contains* C_k*, a non negative integer,* $minc(\bar{C}, R, U_k)$*, and a positive integer or* ∞*,* $maxc(\bar{C}, R, U_k)$*, obtained in the following way:*

$$\begin{aligned} minc(\bar{C}, R, U_k) &= max\,\{m \mid \exists C \in \bar{C}.\ minc(C, R, U_k) = m\} \\ maxc(\bar{C}, R, U_k) &= min\,\{n \mid \exists C \in \bar{C}.\ maxc(C, R, U_k) = n\}\,. \end{aligned}$$

Figure 4 shows the expansion corresponding to the CR-schema of Figure 3, where we abbreviate each class and relationship with its initial letter.

[1]The symbol "\" denotes set difference.

$$
\begin{aligned}
\bar{\mathcal{C}} &= \{\bar{\mathtt{C}}_i \mid 1 \le i \le 7\}, \text{ where}\\
&\quad \bar{\mathtt{C}}_1 = \{\mathtt{S}\},\ \bar{\mathtt{C}}_2 = \{\mathtt{D}\},\ \bar{\mathtt{C}}_3 = \{\mathtt{T}\},\\
&\quad \bar{\mathtt{C}}_4 = \{\mathtt{S},\mathtt{D}\},\ \bar{\mathtt{C}}_5 = \{\mathtt{S},\mathtt{T}\},\ \bar{\mathtt{C}}_6 = \{\mathtt{D},\mathtt{T}\},\\
&\quad \bar{\mathtt{C}}_7 = \{\mathtt{S},\mathtt{D},\mathtt{T}\};\\
\bar{\mathcal{C}}_c &= \{\bar{\mathtt{C}}_1, \bar{\mathtt{C}}_3, \bar{\mathtt{C}}_4, \bar{\mathtt{C}}_5, \bar{\mathtt{C}}_7\};\\
\bar{\mathcal{R}} &= \{\bar{\mathtt{H}}_{ij}, \bar{\mathtt{P}}_{ij} \mid 1 \le i,j \le 7\}, \text{ where}\\
&\quad \bar{\mathtt{H}}_{ij} = \langle \mathtt{U}_1\colon \bar{\mathtt{C}}_i, \mathtt{U}_2\colon \bar{\mathtt{C}}_j\rangle_{\mathtt{H}},\\
&\quad \bar{\mathtt{P}}_{ij} = \langle \mathtt{U}_3\colon \bar{\mathtt{C}}_i, \mathtt{U}_4\colon \bar{\mathtt{C}}_j\rangle_{\mathtt{P}};\\
\bar{\mathcal{R}}_c &= \{\bar{\mathtt{H}}_{ij} \mid i \in \{1,4,5,7\} \wedge j \in \{3,5,7\}\} \cup\\
&\quad \{\bar{\mathtt{P}}_{ij} \mid i \in \{4,7\} \wedge j \in \{3,5,7\}\};
\end{aligned}
$$

$$
\begin{aligned}
& minc(\bar{\mathtt{C}}_1,\mathtt{H},\mathtt{U}_1) = minc(\bar{\mathtt{C}}_4,\mathtt{H},\mathtt{U}_1) = 1;\\
& minc(\bar{\mathtt{C}}_5,\mathtt{H},\mathtt{U}_1) = minc(\bar{\mathtt{C}}_7,\mathtt{H},\mathtt{U}_1) = 1;\\
& maxc(\bar{\mathtt{C}}_4,\mathtt{H},\mathtt{U}_1) = maxc(\bar{\mathtt{C}}_7,\mathtt{H},\mathtt{U}_1) = 2;\\
& minc(\bar{\mathtt{C}}_3,\mathtt{H},\mathtt{U}_2) = minc(\bar{\mathtt{C}}_5,\mathtt{H},\mathtt{U}_2) = 1;\\
& minc(\bar{\mathtt{C}}_7,\mathtt{H},\mathtt{U}_2) = 1;\\
& maxc(\bar{\mathtt{C}}_3,\mathtt{H},\mathtt{U}_2) = maxc(\bar{\mathtt{C}}_5,\mathtt{H},\mathtt{U}_2) = 1;\\
& maxc(\bar{\mathtt{C}}_7,\mathtt{H},\mathtt{U}_2) = 1;\\
& minc(\bar{\mathtt{C}}_4,\mathtt{P},\mathtt{U}_3) = minc(\bar{\mathtt{C}}_7,\mathtt{P},\mathtt{U}_3) = 1;\\
& maxc(\bar{\mathtt{C}}_4,\mathtt{P},\mathtt{U}_3) = maxc(\bar{\mathtt{C}}_7,\mathtt{P},\mathtt{U}_3) = 1;\\
& minc(\bar{\mathtt{C}}_3,\mathtt{P},\mathtt{U}_4) = minc(\bar{\mathtt{C}}_5,\mathtt{P},\mathtt{U}_4) = 1;\\
& minc(\bar{\mathtt{C}}_7,\mathtt{P},\mathtt{U}_4) = 1;
\end{aligned}
$$

Figure 4: The expansion of the CR-schema shown in Figure 3

In order to capture the intuitions given above, we specify the formal semantics of compound classes and compound relationships by extending interpretations to the expansion of a CR-schema. Let $\mathcal{S}$ be a CR-schema, $\mathcal{I}$ an interpretation of $\mathcal{S}$ and $\bar{\mathcal{S}}$ the expansion of $\mathcal{S}$.

If $\bar{C}$ is a compound class of $\bar{\mathcal{S}}$, then its extension $\bar{C}^{\mathcal{I}}$ is defined as follows:

$$\bar{C}^{\mathcal{I}} = \Big\{ d \in \Delta^{\mathcal{I}} \;\Big|\; \begin{array}{l} (\forall C \in \bar{C}. d \in C^{\mathcal{I}}) \ \wedge \\ (\forall C \in \mathcal{C} \setminus \bar{C}. d \notin C^{\mathcal{I}}) \end{array} \Big\}.$$

If $\bar{R} = \langle U_1\colon \bar{C}_1, \ldots, U_K\colon \bar{C}_K\rangle_R$ is a compound relationship corresponding to a relationship R, then its extension $\bar{R}^{\mathcal{I}}$ is defined as follows:

$$\bar{R}^{\mathcal{I}} = \Big\{ \langle U_1\colon \tilde{c}_1, \ldots, U_K\colon \tilde{c}_K\rangle \in R^{\mathcal{I}} \;\Big|\; \tilde{c}_i \in \bar{C}_i^{\mathcal{I}},\ i \in 1..K \Big\}.$$

Conversely, given the extensions of all compound classes and compound relationships of $\bar{\mathcal{S}}$, we can derive the extensions of the classes and relationships of $\mathcal{S}$ in an obvious way. If $C \in \mathcal{C}$ and $R \in \mathcal{R}$, then

$$C^{\mathcal{I}} = \bigcup_{\bar{C} \ni C} \bar{C}^{\mathcal{I}} \qquad \text{and} \qquad R^{\mathcal{I}} = \bigcup_{\bar{R} \in \bar{\mathcal{R}}_R} \bar{R}^{\mathcal{I}}.$$

Notice that the way we interpret compound classes and relationships forces their extensions to be pairwise disjoint in all interpretations. This is crucial for deriving a suitable system of disequations, as shown later. However, the price we have to pay for this property is the exponential number of different compound classes and relationships that form the expansion.

A model of a CR-schema $\mathcal{S}$ will also be called a model of the expansion $\bar{\mathcal{S}}$ of $\mathcal{S}$. The following lemma states the conditions for an interpretation $\mathcal{I}$ of $\mathcal{S}$ to be a model of $\bar{\mathcal{S}}$.

Lemma 3.2 *An interpretation $\mathcal{I}$ is a model of the expansion $\bar{\mathcal{S}}$ of a CR-schema $\mathcal{S}$, if and only if it satisfies the following conditions:*

(A') For each compound class $\bar{C} \in \bar{\mathcal{C}}$ that is not consistent, it holds that $\bar{C}^{\mathcal{I}} = \emptyset$.

(B') For each compound relationship $\bar{R} = \langle U_1\colon \bar{C}_1, \ldots, U_K\colon \bar{C}_K\rangle_R \in \mathcal{R}$, if $\bar{R}$ is consistent, then all labeled tuples of $\bar{R}^{\mathcal{I}}$ are of the form $\langle U_1\colon \tilde{c}_1, \ldots, U_K\colon \tilde{c}_K\rangle$, where $\tilde{c}_k \in \bar{C}_k^{\mathcal{I}}$, for $k = 1, \ldots, K$, and if $\bar{R}$ is not consistent then $\bar{R}^{\mathcal{I}} = \emptyset$.

(C') For each relationship $R = \langle U_1\colon C_1, \ldots, U_K\colon C_K\rangle \in \mathcal{R}$, for $k = 1, \ldots, K$, for each consistent compound class $\bar{C} \in \bar{\mathcal{C}}_c$ containing C_k and for each instance $\tilde{c}$ of $\bar{C}$ in $\mathcal{I}$, it holds that

$$minc(\bar{C}, R, U_k) \le \Big| \Big\{ \tilde{r} \in R^{\mathcal{I}} \;\Big|\; \tilde{r}[U_k] = \tilde{c} \Big\} \Big| \le$$
$$\le maxc(\bar{C}, R, U_k).$$

3.2 System of disequations corresponding to a CR-schema

We recall that we aim at developing a method for checking the satisfiability of a CR-schema $\mathcal{S}$. Lemma 3.2 guarantees that any interpretation $\mathcal{I}$ of the expansion $\bar{\mathcal{S}}$ that satisfies conditions (A') through (C') is a model of $\bar{\mathcal{S}}$ and hence also of $\mathcal{S}$. Therefore, in the following we can concentrate on establishing if the expansion of a CR-schema admits a model.

In order to check this property, we derive a system $\Psi_{\mathcal{S}}$ of linear disequations from the expansion $\bar{\mathcal{S}}$ of $\mathcal{S}$. The unknowns of $\Psi_{\mathcal{S}}$ are obtained as follows:

- For each compound class $\bar{C} \in \bar{\mathcal{C}}$ we introduce an unknown $\mathrm{Var}(\bar{C})$. These unknowns are called *class unknowns*.
- For each compound relationship $\bar{R} \in \bar{\mathcal{R}}$ we introduce an unknown $\mathrm{Var}(\bar{R})$. These unknowns are called *relationship unknowns*.

The disequations of $\Psi_{\mathcal{S}}$ are obtained as follows:

$$
\begin{array}{l}
\mathrm{Var}(\bar{\mathtt{C}}_i) = \mathtt{c}_i, \;\; 1 \le i \le 7; \\
\mathrm{Var}(\bar{\mathtt{H}}_{ij}) = \mathtt{h}_{ij}, \;\; \mathrm{Var}(\bar{\mathtt{P}}_{ij}) = \mathtt{p}_{ij}, \;\; 1 \le i,j \le 7;
\end{array}
$$

$$
\begin{array}{rcll}
\mathtt{c}_i & = & 0, & i \in \{2,6\}; \\
\mathtt{c}_i & \ge & 0, & i \in \{1,3,4,5,7\}; \\
\mathtt{h}_{ij} & = & 0, & i \in \{2,3,6\} \vee j \in \{1,2,4,6\}; \\
\mathtt{p}_{ij} & = & 0, & i \in \{1,2,3,5,6\} \vee j \in \{1,2,4,6\}; \\
\mathtt{h}_{ij} & \ge & 0, & i \in \{1,4,5,7\} \vee j \in \{3,5,7\}; \\
\mathtt{p}_{ij} & \ge & 0, & i \in \{4,7\} \vee j \in \{3,5,7\};
\end{array}
$$

$$
\begin{array}{rcll}
\mathtt{c}_i & \le & \mathtt{h}_{i3} + \mathtt{h}_{i5} + \mathtt{h}_{i7}, & i \in \{1,4,5,7\}; \\
2\mathtt{c}_i & \ge & \mathtt{h}_{i3} + \mathtt{h}_{i5} + \mathtt{h}_{i7}, & i \in \{4,7\}; \\
\mathtt{c}_j & \le & \mathtt{h}_{1j} + \mathtt{h}_{4j} + \mathtt{h}_{5j} + \mathtt{h}_{7j}, & j \in \{3,5,7\}; \\
\mathtt{c}_j & \ge & \mathtt{h}_{1j} + \mathtt{h}_{4j} + \mathtt{h}_{5j} + \mathtt{h}_{7j}, & j \in \{3,5,7\}; \\
\mathtt{c}_i & \le & \mathtt{p}_{i3} + \mathtt{p}_{i5} + \mathtt{p}_{i7}, & i \in \{4,7\}; \\
\mathtt{c}_i & \ge & \mathtt{p}_{i3} + \mathtt{p}_{i5} + \mathtt{p}_{i7}, & i \in \{4,7\}; \\
\mathtt{c}_j & \le & \mathtt{p}_{4j} + \mathtt{p}_{7j}, & j \in \{3,5,7\}.
\end{array}
$$

Figure 5: The system corresponding to the expansion shown in Figure 4

1. for each compound class or compound relationship $\bar{X}$ that is not consistent we introduce

$$\mathrm{Var}(\bar{X}) = 0$$

2. for each relationship $R = \langle U_1 \colon C_1, \ldots, U_K \colon C_K \rangle \in \mathcal{R}$, for $k = 1, \ldots, K$, and for each consistent compound class $\bar{C} \in \bar{\mathcal{C}}_c$ that contains C_k:

   - if $minc(\bar{C}, R, U_k) = m > 0$, we introduce

$$m \cdot \mathrm{Var}(\bar{C}) \le \sum_{\bar{R} \in \bar{\mathcal{R}}_R \mid \bar{R}[U_k] = \bar{C}} \mathrm{Var}(\bar{R})$$

   - if $maxc(\bar{C}, R, U_k) = n \ne \infty$, we introduce

$$n \cdot \mathrm{Var}(\bar{C}) \ge \sum_{\bar{R} \in \bar{\mathcal{R}}_R \mid \bar{R}[U_k] = \bar{C}} \mathrm{Var}(\bar{R})$$

3. for each consistent compound class or compound relationship $\bar{X}$, we introduce

$$\mathrm{Var}(\bar{X}) \ge 0$$

Note that the resulting system $\Psi_{\mathcal{S}}$ of linear disequations is homogeneous (i.e. all of its constant terms are equal to 0) and has integer coefficients. Figure 5 shows the system of disequations corresponding to the expansion of Figure 4.

3.3 Characterization of satisfiability

In order to relate the models of the expansion $\bar{\mathcal{S}}$ of a CR-schema to particular solutions of the associated system $\Psi_{\mathcal{S}}$, we specify an additional condition which must be satisfied by the solutions of $\Psi_{\mathcal{S}}$. First, we need a further definition. If $\bar{R}$ is a compound relationship that associates a compound class $\bar{C}$ to role U, then we say that $\mathrm{Var}(\bar{R})$ *depends on Var*$(\bar{C})$ *via role* U. Also, we say that $\mathrm{Var}(\bar{R})$ *depends on* $\mathrm{Var}(\bar{C})$ if it depends on $\mathrm{Var}(\bar{C})$ via some role U.

Let X be a solution of $\Psi_{\mathcal{S}}$ and $X(y)$ denote the value assigned by X to a class or relationship unknown y. Then, X is said to be *acceptable* if for all relationship unknowns r depending on a class unknown c such that $X(c) = 0$, it holds that $X(r) = 0$.

We are now ready to prove the main result concerning the satisfiability of a class.

Theorem 3.3 *A class C_s of a CR-schema $\mathcal{S}$ is satisfiable, if and only if*

$$\Psi'_{\mathcal{S}} = \Psi_{\mathcal{S}} \bigcup \left\{ \sum_{\bar{C} \ni C_s} Var(\bar{C}) > 0 \right\}$$

admits an acceptable solution.

Coming back to the example of Figure 2, suppose we want to check whether the class `Speaker` is satisfiable. The above theorem tells us that we can add to the system shown in Figure 5 the disequation $\mathtt{c}_1 + \mathtt{c}_4 + \mathtt{c}_5 + \mathtt{c}_7 > 0$, and check whether it admits acceptable nonnegative integer solutions. It turns out that one such solution, named X, is the one given in Figure 6. The way we built the disequation system ensures us that from this solution it is possible to construct a model of the schema where the number of instances of each compound class and compound relationship is exactly the value assigned by the solution to the corresponding unknown. Figure 6 gives a model derived from the solution X.

Now, consider the following additional condition: Each speaker that is allowed to participate in a discussion, must hold at least two talks. It forces us to add to the schema of Figure 3 the constraint $minc(\mathtt{Discussant}, \mathtt{Holds}, \mathtt{U}_1) = 2$, that is reflected by the disequations

$$2\mathtt{c}_i \le \mathtt{h}_{i3} + \mathtt{h}_{i5} + \mathtt{h}_{i7}, \;\; i \in \{4,7\}.$$

If we add them to the system of Figure 5, together with $\mathtt{c}_1 + \mathtt{c}_4 + \mathtt{c}_5 + \mathtt{c}_7 > 0$, then the system becomes unsolvable. Intuitively, this happens because the original constraints on the schema forced each talk to have exactly one discussant and also each speaker to be a discussant and to hold exactly one talk (we will see a formal justification for this in the next section). Therefore the number of talks, speakers and discussants must be the same and the additional constraint gives rise to a contradiction.

$X(\mathtt{c}_i) = \hat{\mathtt{c}}_i, \;\; 1 \leq i \leq 7;$
$X(\mathtt{h}_{ij}) = \hat{\mathtt{h}}_{ij}, \;\; X(\mathtt{p}_{ij}) = \hat{\mathtt{p}}_{ij}, \;\; 1 \leq i, j \leq 7;$

$\hat{\mathtt{c}}_3 = \hat{\mathtt{c}}_4 = 2; \;\; \hat{\mathtt{c}}_1 = \hat{\mathtt{c}}_2 = \hat{\mathtt{c}}_5 = \hat{\mathtt{c}}_6 = \hat{\mathtt{c}}_7 = 0;$
$\hat{\mathtt{h}}_{34} = 2; \;\; \hat{\mathtt{h}}_{ij} = 0, \;\; i \neq 3 \vee j \neq 4;$
$\hat{\mathtt{p}}_{34} = 2; \;\; \hat{\mathtt{p}}_{ij} = 0, \;\; i \neq 3 \vee j \neq 4;$

$\Delta^{\mathcal{I}} = \{\mathtt{John}, \mathtt{Mary}, \mathtt{talk}_J, \mathtt{talk}_M\},$
$\mathtt{Speaker}^{\mathcal{I}} = \{\mathtt{John}, \mathtt{Mary}\},$
$\mathtt{Discussant}^{\mathcal{I}} = \{\mathtt{John}, \mathtt{Mary}\},$
$\mathtt{Talk}^{\mathcal{I}} = \{\mathtt{talk}_J, \mathtt{talk}_M\},$
$\mathtt{Holds}^{\mathcal{I}} = \{\langle \mathtt{U}_1 \colon \mathtt{John}, \mathtt{U}_2 \colon \mathtt{talk}_J \rangle,$
$\langle \mathtt{U}_1 \colon \mathtt{Mary}, \mathtt{U}_2 \colon \mathtt{talk}_M \rangle\},$
$\mathtt{Participates}^{\mathcal{I}} = \{\langle \mathtt{U}_3 \colon \mathtt{John}, \mathtt{U}_4 \colon \mathtt{talk}_M \rangle,$
$\langle \mathtt{U}_3 \colon \mathtt{Mary}, \mathtt{U}_4 \colon \mathtt{talk}_J \rangle\}.$

Figure 6: A solution of the system shown in Figure 5

It remains to show that it is decidable to verify whether a system Ψ of linear homogeneous disequations admits an acceptable solution. Let $\mathcal{V}_{\mathcal{C}}$ and $\mathcal{V}_{\mathcal{R}}$ be respectively the class and relationship unknowns of Ψ. For a generic subset $\mathcal{Z}$ of $\mathcal{V}_{\mathcal{C}}$, let $\Psi^{\mathcal{Z}}$ be the system of linear disequations obtained from Ψ as follows:

$$\begin{aligned} \Psi^{\mathcal{Z}} \;\; = \;\; & \Psi \;\cup\; \left\{ c = 0 \;\middle|\; c \in \mathcal{Z} \right\} \;\cup \\ & \left\{ c > 0 \;\middle|\; c \in \mathcal{V}_{\mathcal{C}} \setminus \mathcal{Z} \right\} \;\cup \\ & \left\{ r \geq 0 \;\middle|\; r \in \mathcal{V}_{\mathcal{R}} \right\} \;\cup \\ & \left\{ r = 0 \;\middle|\; r \in \mathcal{V}_{\mathcal{R}} \wedge \exists c \in \mathcal{Z}. r \text{ depends on } c \right\}. \end{aligned}$$

Theorem 3.4 *Ψ admits an acceptable solution if and only if for some $\mathcal{Z} \subseteq \mathcal{V}_{\mathcal{C}}$, $\Psi^{\mathcal{Z}}$ admits a solution.*

Since it is well known that checking whether a system of linear homogeneous disequations admits a solution, can be done in polynomial time, the above theorem ensures us that satisfiability in CR is decidable. In particular, our method can be turned into an algorithm running in exponential time with respect to the size of the schema. It is also possible to show that the satisfiability problem in CR is polynomially intractable. We will come back to the issues related to computational complexity of the method in the concluding section.

4 Implication in CR

The *implication* problem in our framework can be stated as follows. We consider a CR-schema $\mathcal{S}$, and a constraint statement of one of the following forms:

$$\begin{aligned} C & \preceq D \\ minc(C, R, U) & = m \\ maxc(C, R, U) & = n \end{aligned}$$

where m is a positive and n a nonnegative integer, C and D are classes of $\mathcal{S}$, R is a relationship of $\mathcal{S}$ and $U \in role(R)$. An interpretation $\mathcal{I}$ *satisfies*

- $C \preceq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$;
- $minc(C, R, U) = m$ if for any instance $\tilde{c}$ of C in $\mathcal{I}$, the number of instances of R in $\mathcal{I}$ that assign $\tilde{c}$ to U is greater than or equal to m;
- $maxc(C, R, U) = n$ if for any instance $\tilde{c}$ of C in $\mathcal{I}$, the number of instances of R in $\mathcal{I}$ that assign $\tilde{c}$ to U is less than or equal to n.

Analogously to satisfiability, we are interested in implication in finite models. We say that a CR-schema $\mathcal{S}$ *implies* a constraint statement $\mathcal{K}$, written $\mathcal{S} \models \mathcal{K}$, if every finite model of $\mathcal{S}$ satisfies $\mathcal{K}$. We show that the proposed method for satisfiability checking can be easily adapted to solve the implication problem.

ISA constraints: To decide whether $\mathcal{S} \models (C \preceq D)$, where C and D are classes of $\mathcal{S}$, we take the following system of disequations:

$$\Psi'_{\mathcal{S}} = \Psi_{\mathcal{S}} \cup \left\{ \sum_{\bar{C} \ni C \,\wedge\, \bar{C} \not\ni D} \mathrm{Var}(\bar{C}) > 0 \right\}.$$

It is easy to see that $\mathcal{S} \models (C \preceq D)$ if and only if $\Psi'_{\mathcal{S}}$ admits no solution. In fact, if $\Psi'_{\mathcal{S}}$ admits a solution, by applying the method described above we can construct a model $\mathcal{I}$ of $\mathcal{S}$ with at least one instance of C that is not an instance of D. Therefore $C \preceq D$ does not follow from $\mathcal{S}$. On the other hand, if there is a model $\mathcal{I}$ of $\mathcal{S}$ such that $C^{\mathcal{I}} \not\subseteq D^{\mathcal{I}}$, for the completeness of the proposed method, $\Psi_{\mathcal{S}}$ must admit a solution X in which at least one unknown corresponding to a compound class containing C and not containing D gets a positive value. Therefore X is also a solution of $\Psi'_{\mathcal{S}}$.

Cardinality constraints: To decide whether $\mathcal{S} \models (minc(C, R, U) = m)$, where $m > 0$, we introduce a new class C_{exc} and construct the following CR-schema $\mathcal{S}'$:

- $\mathcal{C}' = \mathcal{C} \cup \{C_{exc}\}$;
- $\mathcal{R}' = \mathcal{R}$ and $\mathcal{U}' = \mathcal{U}$;
- $\mathcal{S}'_{isa} = \mathcal{S}_{isa} \cup \{C_{exc} \preceq C\}$;

$$\mathcal{S} \models \texttt{Speaker} \preceq \texttt{Discussant}$$
$$\mathcal{S} \models maxc(\texttt{Talk}, \texttt{Participates}, \texttt{U}_4) = 1$$
$$\mathcal{S} \models maxc(\texttt{Speaker}, \texttt{Holds}, \texttt{U}_1) = 1$$

Figure 7: Inferences from the CR-diagram shown in Figure 2

- $maxc(C_{exc}, R, U) = (m-1)$ and for all other $C' \in \mathcal{C}$, $R' \in \mathcal{R}$ and $U' \in \mathcal{U}$, $minc$ and $maxc$ are defined as in $\mathcal{S}$.

It is easy to see that $\mathcal{S} \models (minc(C, R, U) = m)$ if and only if C_{exc} is not satisfiable in $\mathcal{S}'$. In fact, if C_{exc} is satisfiable in $\mathcal{S}'$, each instance of C_{exc} in $\mathcal{S}'$ is also an instance of C in $\mathcal{S}$ which is assigned by at most $m-1$ instances of R to role U. On the other hand, if C_{exc} is not satisfiable in $\mathcal{S}'$, then there is no model of $\mathcal{S}'$, and hence of $\mathcal{S}$, with an instance of C that is assigned by less than m instances of R to role U, because this instance would also have to be an instance of C_{exc}.

With regard to deciding whether $\mathcal{S} \models (minc(C, R, U) = n)$, we can proceed similarly to the previous case.

Referring again to the example of Figure 2, some of the inferences that can be drawn are shown in Figure 7, where $\mathcal{S}$ denotes the schema of Figure 3.

5 Conclusions

Reasoning about database schemas is an important task in database design. Typically, when the data model is formally defined, reasoning essentially means being able to check both satisfiability and logical implication. For example, one of the most important issues addressed in the relational theory of data has been to devise effective methods for dependency implication in order to derive interesting properties of the schema. More recently, great attention has been devoted to studying suitable algorithms for subtyping computation in object-oriented data models, which is the main reasoning task to be performed in those data models supporting inheritance. Also, some basic reasoning capabilities are embedded in several CASE tools, most of which are based on an ER data model. It is important to note that, although these tools often include some form of ISA and cardinality constraints, they do not offer any complete method for deriving interesting properties of schemas comprising such constraints.

In this paper, we have presented an effective method for reasoning about a set of ISA and cardinality constraints in the context of a simple data model based on the notions of classes and relationships. The method allows one both to verify the satisfiability of a schema and to check whether a schema implies a given constraint of any of the two kinds, and shows that the reasoning problem for ISA and cardinality constraints is decidable.

There are at least three problems related to our method that we aim at investigating in the near future.

The first problem concerns the extension of the method in order to capture more expressive modeling features. Our first investigation shows that we can directly consider an extension which takes into account disjointness statements between classes, covering constraints [20], and qualification conditions which restrict the participation of classes in relationships.

The second problem concerns the efficiency of the method. Although we did not discuss this issue in the paper, there are many possible criteria for decreasing the complexity of the method. For example, the knowledge about the structure of the schema (for example, existence of cycles) allows several forms of simplifications on the system of disequations. Also, disjointness constraints between classes not only enhance the expressive power of the model, but can also lead to a dramatic reduction of the size of the resulting system, by limiting the number of compound classes and compound relationships to be considered. Taking as an example the diagram of Figure 2, the natural restriction that talks and speakers be disjoint leads to a system of disequations with just a few unknowns.

Finally, we are studying an extension of the method in order to assist the designer when a schema is found unsatisfiable. The idea is to equip our method with a technique that provides the designer with a minimum number of constraints that are unsatisfiable, thus supporting her in schema debugging.

References

[1] D. Maier, *The Theory of Relational Databases*. Computer Science Press, Potomac, Maryland, 1983.

[2] R. Hull and R. King, "Semantic database modelling: Survey, applications and research issues," *ACM Computing Surveys*, vol. 19, pp. 201–260, Sept. 1987.

[3] S. Cosmadakis, P. Kanellakis, and M. Vardi, "Polynomial-time implication problems for unary inclusion dependencies," *Journal of the ACM*, vol. 37, pp. 15–46, Jan. 1990.

[4] J. Abrial, "Data semantics," in *Data Base Management* (J. Klimbie and K. Koffeman, eds.), pp. 1–59, North-Holland Publ. Co., Amsterdam, 1974.

[5] C. Batini, S. Ceri, and S. Navathe, *Conceptual Database Design, an Entity-Relationship Approach*. Benjamin and Cummings Publ. Co., Menlo Park, California, 1992.

[6] J. Ullman, *Principles of Database and Knowledge Base Systems*, vol. 1. Computer Science Press, Potomac, Maryland, 1988.

[7] H. Korth and A. Silberschatz, *Database Systems Concepts*. McGraw-Hill, New York, 1986.

[8] R. ElMasri and S. Navathe, *Fundamentals of Database Systems*. Benjamin and Cummings Publ. Co., Menlo Park, California, 1988.

[9] A. Albano, G. Ghelli, and R. Orsini, "A relationship mechanism for strongly typed Object-Oriented database programming languages," in *Proc. of the 17th Int. Conf. on Very Large Data Bases VLDB-91*, (Barcelona), pp. 565–575, 1991.

[10] R. Fikes and T. Kehler, "The role of frame-based representation in reasoning," *Communications of the ACM*, vol. 28, no. 9, pp. 904–920, 1985.

[11] S. Cosmadakis and P. Kanellakis, "Functional and inclusion dependencies - a graph theoretical approach," in *Advances in Computing Research, Vol. 3* (P. Kanellakis and F. Preparata, eds.), pp. 163–184, JAI Press, 1986.

[12] J. Grant and J. Minker, "Numerical dependencies," in *Advances in Database Theory II* (H. Gallaire, J. Minker, and J.-M. Nicolas, eds.), Plenum Publ. Co., New York, 1984.

[13] M. Lenzerini and G. Santucci, "Cardinality constraints in the entity-relationship model," in *Proc. of the Int. Conf. on the Entity-Relationship Approach to Software Engineering* (C. Davis and others, eds.), pp. 529–549, North-Holland Publ. Co., Amsterdam, 1983.

[14] B. Thalheim, "Fundamentals of cardinality constraints," in *Proc. of the 11th Int. Conf. on the Entity-Relationship Approach ER-92*, pp. 7–23, Springer-Verlag, 1992.

[15] M. Lenzerini and P. Nobili, "On the satisfiability of dependency constraints in entity-relationship schemata," *Information Systems*, vol. 15, no. 4, pp. 453–461, 1990.

[16] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt, "The complexity of concept languages," in *Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning KR-91* (J. Allen, R. Fikes, and E. Sandewall, eds.), pp. 151–162, Morgan Kaufman, Los Altos, 1991.

[17] M. Buchheit, F. M. Donini, and A. Schaerf, "Decidable reasoning in terminological knowledge representation systems," in *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence IJCAI-93*, (Chambery, France), pp. 704–709, Morgan Kaufman, Los Altos, 1993.

[18] B. Nebel, "Computational complexity of terminological reasoning in BACK," *Artificial Intelligence*, vol. 34, no. 3, pp. 371–383, 1988.

[19] P. Chen, "The Entity-Relationship model: Toward a unified view of data," *ACM Trans. on Database Syst.*, vol. 1, pp. 9–36, Mar. 1976.

[20] M. Lenzerini, "Covering and disjointness constraints in type networks," in *Proc. of the 3rd IEEE Int. Conf. on Data Engineering*, (Los Angeles), pp. 386–393, IEEE Computer, 1987.