Non-destructive Integration of Form-based Views

Jan Hidders[1], Jan Paredaens[1], Philippe Thiran[2], Geert-Jan Houben[2], and
Kees van Hee[2]

[1] University of Antwerp, Belgium
[2] Eindhoven University of Technology, The Netherlands

Abstract. Form documents or screen forms bring essential information
on the data manipulated by an organization. They can be considered
as different but often overlapping views of its whole data. This paper
presents a non-destructive approach of their integration. The main idea
of our approach is to keep the original views intact and to specify con-
straints between overlapping structures. For reasoning over constraints,
we provide a set of inference rules that allows not only to infer implied
constraints but also to detect conflicts. These reasoning rules are proved
to be sound and complete. Although the form-based views are hierarchi-
cal structures, our constraints and reasoning rules can also be used in
non-hierarchical data models.

1 Introduction

In the design process for data-intensive applications the design of the global
data model is a crucial step. Often this step involves the integration of different
data models that each describe the information need of different groups of end
users. In the case of workflow and case management systems these data models
or views are usually defined as a form, hence *form-based views* and the tasks that
are managed by the system are typically manipulations of these forms. For large
and complex workflows the task of modeling the forms is often split according
to the different case types. The consequence is often that we obtain a set of
different data models that contain synonyms (different class names that refer to
the same class) and homonyms (the same class name is used in different models
with a different meaning).

A classical solution for resolving this problem is to integrate the different
views into a single global schema [1]. However, in this paper we will integrate
the different views by taking a disjoint union of them and adding constraints that
express semantic relationships between the classes and relations in the different
data models. Since the original views remain part of the global data model we
call this *non-destructive integration*. The fact that the original views remain part
of the global data model is important in the case of workflow systems because the
the views are part of the description of the execution of the workflow. However,
even for other types of data-intensive information systems such a design has

benefits. Since the original class and relation names from the views are kept in the global data model this will make communication easier with the end-users for which the views described an information need. Moreover, since the relationships between the different views are made explicit it will be easy to see how changes to the global data model affect the different views and vice versa.

The main contribution of this paper is the presentation of a small but powerful set of semantical relationships between classes and relations in different views, and a sound and complete set of inference rules that allows us to derive implied relationships and in particular whether there is a conflict in the resulting global schema.

The paper is organized as follows. Section 2 develops a small example that allows us to informally present our non-destructive approach of form-based view integration. In Section 3, we formally specify the problem. The schema and their instances are defined as well as the different constraints we consider. Sections 4, 5 and 6 present different sets of inference rules for deriving constraints and detecting conflicts. For each set, we prove their soundness and their completeness. In Section 7, we discuss related works. We give our concluding remarks in Section 8.

2   Informal problem definition

We assume that the process of data integration starts with so-called *form-based views* which are essentially hierarchical data structures that describe complex values which can be roughly thought of as tree-shaped graphs. Three examples of such views are given in Figure 1. Each view has the form of a tree which defines all the data that is shown in the view. The nodes of these graphs can be interpreted as classes that contain sets of objects, and the edges can be interpreted as binary relationships between these classes. The root node indicates for which class the view is defined as well as the name of the view. The nodes directly below a node define the attributes of this class. For example, in the Patient view we see that for a patient we have the patient's names, diseases, rooms and treating doctors. At the next level in the view we see that for a disease of a patient we have its types and its names. For the purpose of this paper we will assume that all attributes are set-valued, i.e., they can contain zero, one or more objects.

In the three views in the example we see that there is an overlap in the sense that some objects such as those in the Doctor class in the Patient view and those in the Doctor class in the Doctor view are in fact the same object. In a similar fashion it holds that some of the pairs of the Department-Doctor-Manager relationship in the Doctor view will also be pairs in the Department-Manager relationship. This type of redundancy can be solved by integrating the views into a single new schema, but we propose to leave the original views intact and explicitly specifies such constraint between the different views as illustrated in Figure 2.

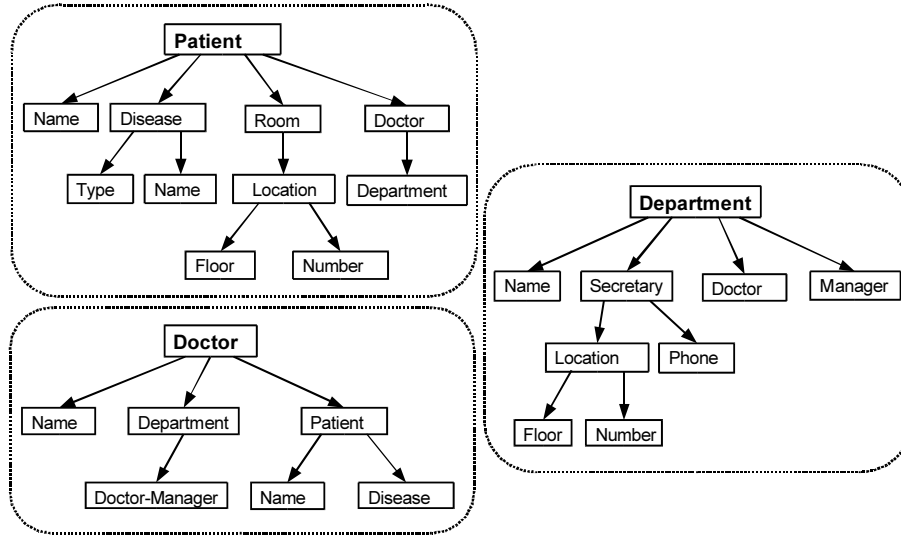There are 8 types of constraints that we will consider:

Fig. 1. Three form-based views

ISA The ISA constraint between classes is indicated by an edge that is labeled with $\Rightarrow$. An example is the edge between the Department class in the Doctor view and the Department class in the Department view. The ISA constraint indicates that the objects in one class must also be objects of the other class.

Relational ISA The relational ISA constraint between relationships is indicated by an edge that is labeled with $\Rightarrow^{\downarrow}$. An example is the edge between the Department-Doctor-Manager relationship in the Doctor view and the Department-Manager relationship in the Department view. This constraint indicates that all pairs of the first relationship are also pairs of the second relationship.

Inverse Relational ISA The inverse relational ISA constraint between relationships is indicated by an edge that is labeled with $\Rightarrow^{\uparrow}$. An example is the edge between the Patient-Doctor relationship in the Patient view and the Doctor-Patient relationship in the Doctor view. This constraint indicates that all the inverse pairs of the first relationship are also pairs of the second relationship.

Disjointness The disjointness constraint between classes is indicated by an edge that is labeled with $\not\approx$. An example is the edge between the Location class in the Patient view and the Location class in the Department view. This constraint indicates that the two classes cannot have common objects.

Relational Disjointness The relational disjointness constraint between relationships is indicated by an edge that is labeled with $\not\approx^{\downarrow}$. This constraint indicates that the two relationships cannot have common pairs.

Fig. 2. Three integrated form-based views

Inverse Relational Disjointness The inverse relational disjointness constraint between relationships is indicated by an edge that is labeled with $\wr^{\uparrow}$. This constraint indicates that there cannot be a pair in one relationship such that the inverse pair is in the other relationship.

Totalness The totalness constraint of a relationship r is indicated by a solid dot at the beginning of the edge of r. It indicates that the relationship r is total, i.e.,. that for every object o in the source class of r there is a pair of r whose first component is o. For example, every patient has a name, a room and a doctor, but probably has no disease.

Surjectivity constraint The surjectivity constraint of a relationship r is indicated by a solid dot at the end of the edge of r. It indicates that the relationship r is surjective, i.e., for every object o in the target class of r there is a pair of r whose second component is o.

The schema of Figure 2 has a straightforward interpretation that is similar to that of FDM [2], binary ORM [3] and the data models that are used in descriptive logics. Note that in all these models the instances of a schema are essentially graphs that somehow match the schema. Since the original views are still present in the schema it is also clear that for each view we can define a projection on the instances of this schema. Although this projection will usually define a graph it can always be transformed into a forest by splitting nodes with two incoming edges. This means that the general approach is here that of the *local as view* (LAV) approach as defined in [4].

When the constraints are added to the views it is possible that conflicts appear. For example, if there is an ISA constraint between the classes A and B

and at the same time a disjointness constraint between them then the class A nor B can never be populated.

In the remainder of this paper, we discuss the problem of reasoning over schemas with such constraints in order to infer implied such constraints and to detect conflicts.

3 Formal problem definition

For formally specifying the form-based views, we use a graph representation instead of a tree representation as presented in the previous section. We use this simplified data representation since our aim is to provide some constraints that can be used in a context broader than the forms. As such, we are now considering these two types of data models:

– *Frame:* graph structure related to an original form-based view,
– *Schema:* graph structure related to the union of disjoint form-based views, with constraints among them.

In the following paragraphs, we present these types by giving their schema definition and their instance definition.

A *frame* is a multigraph where the nodes represent classes and the edges relationships. More formally:

Definition 1 (Frame). *A* frame *is a tuple $F = (C, R, s, t)$ with C a set of classes, R a set of relationships, $s : R \to C$ a function that indicates the source class of a relationship, and $t : R \to C$ a function that indicates the target class of a relationship.*

Definition 2 (Instance). *An* instance *of a frame $F = (C, R, s, t)$ is a tuple $I = (O, \llbracket \cdot \rrbracket)$ with O a set of objects, and $\llbracket \cdot \rrbracket$ the interpretation function that maps classes $c \in C$ to a subset of O, denoted as $\llbracket c \rrbracket$, and relationships $r \in R$ to subsets of $O \times O$, denoted as $\llbracket r \rrbracket$, such that, for all relationships $r \in R$, it holds that:*

$$\llbracket r \rrbracket \subseteq \llbracket s(r) \rrbracket \times \llbracket t(r) \rrbracket \tag{1}$$

A *schema* is a frame over which some constraints are specified.

Definition 3 (Constraint). *Given a frame $F = (C, R, s, t)$ a* constraint *is one of the following:*

subclass $c_1 \Rightarrow c_2$, $r_1 \Rightarrow^{\downarrow} r_2$, $r_1 \Rightarrow^{\uparrow} r_2$
cardinality $r\cdot$, $\cdot r$
disjointness $c_1 \nsim c_2$, $r_1 \nsim^{\downarrow} r_2$, $r_1 \nsim^{\uparrow} r_2$

with $r, r_1, r_2 \in R$ and $c_1, c_2 \in C$. We let $I \vdash k$ denote that constraint k holds for instance $I = (O, \llbracket \cdot \rrbracket)$. Then we have:

$$I \vdash c_1 \Rightarrow c_2 \; \textit{iff} \; [\![c_1]\!] \subseteq [\![c_2]\!] \tag{2}$$

$$I \vdash r_1 \Rightarrow^\downarrow r_2 \; \textit{iff} \; [\![r_1]\!] \subseteq [\![r_2]\!] \tag{3}$$

$$I \vdash r_1 \Rightarrow^\uparrow r_2 \; \textit{iff} \; [\![r_1]\!]^{-1} \subseteq [\![r_2]\!] \tag{4}$$

$$I \vdash \cdot r \; \textit{iff} \; [\![s(r)]\!] = \{o_1 | (o_1, o_2) \in [\![r]\!]\} \tag{5}$$

$$I \vdash r \cdot \; \textit{iff} \; [\![t(r)]\!] = \{o_2 | (o_1, o_2) \in [\![r]\!]\} \tag{6}$$

$$I \vdash c_1 \nsim c_2 \; \textit{iff} \; [\![c_1]\!] \cap [\![c_2]\!] = \varnothing \tag{7}$$

$$I \vdash r_1 \nsim^\downarrow r_2 \; \textit{iff} \; [\![r_1]\!] \cap [\![r_2]\!] = \varnothing \tag{8}$$

$$I \vdash r_1 \nsim^\uparrow r_2 \; \textit{iff} \; [\![r_1]\!]^{-1} \cap [\![r_2]\!] = \varnothing \tag{9}$$

Definition 4 (Schema). *A* schema *is a tuple $S = (F, K)$ with F a frame and K a finite set of constraints over F.*

Definition 5 (Instance). *An* instance of schema S *is an instance I of frame F such that $I \vdash k$ for all constraints $k \in K$.*

4 Inference rules for deriving subclass constraints

In this section, we present the set of inference rules M_1 that only derive constraints of the forms $c_1 \Rightarrow c_2$, $r_1 \Rightarrow^\downarrow r_2$, $r_1 \Rightarrow^\uparrow r_2$. In Figure 3, we give the set of rules $M1$. We assume that the inference rules are defined given a frame $F = (C, R, s, t)$ and variables $c, c_1, c_2, \dots$ range over C and $r, r_1, r_2, \dots$ range over R. If for a relationship $r \in R$, it holds that $s(r) = c_1$ and $t(r) = c_2$ then this is denoted as $c_1 \xrightarrow{r} c_2$. We will also assume that K^* is the closure of K under the rules in M_1.

4.1 Instance construction

For proving the completeness of the inference rules in M_1, we construct the instances I^{cl} and I^{rel} given a schema $S = (F, K)$ with $F = (C, R, s, t)$.

Informally, the instance I^{cl} is constructed as follows. It is assumed that $d_1 \Rightarrow d_2 \notin K^*$. We introduce two objects, o_1 and o_2 where o_1 is in only the superclasses of d_1 and o_2 is simply in all classes of the schema. Then we fill the relations with pairs that contain o_1 to satisfy the surjectivity and totalness constraints. This leads to the following formal definition.

Definition 6 (Instance I^{cl}). *Given a schema S with $d_1 \Rightarrow d_2 \notin K^*$, we define $I^{cl} = (O^{cl}, [\![\cdot]\!]^{cl})$ such that $O^{cl} = \{o_1, o_2\}$ and $[\![\cdot]\!]^{cl}$ the smallest function[3] that satisfies the following rules for all classes c:*

$$o_1 \in [\![c]\!] \; \textit{if} \; d_1 \Rightarrow c \in K^* \tag{10}$$

$$o_2 \in [\![c]\!] \tag{11}$$

---

[3] The ordering over set-valued functions over the same domain is defined such that f is smaller than g iff $f(x) \subseteq g(x)$ for all x in the domain.

$$\text{REFL} \ \frac{}{c \Rightarrow c} \qquad\qquad \text{TRANS} \ \frac{c_1 \Rightarrow c_2 \qquad c_2 \Rightarrow c_3}{c_1 \Rightarrow c_3} \qquad\qquad \text{RELRFL} \ \frac{}{r \Rightarrow^\downarrow r}$$

$$\text{RELTR1} \ \frac{r_1 \Rightarrow^\downarrow r_2 \qquad r_2 \Rightarrow^\downarrow r_3}{r_1 \Rightarrow^\downarrow r_3} \qquad\qquad \text{RELTR2} \ \frac{r_1 \Rightarrow^\downarrow r_2 \qquad r_2 \Rightarrow^\uparrow r_3}{r_1 \Rightarrow^\uparrow r_3}$$

$$\text{RELTR3} \ \frac{r_1 \Rightarrow^\uparrow r_2 \qquad r_2 \Rightarrow^\downarrow r_3}{r_1 \Rightarrow^\uparrow r_3} \qquad\qquad \text{RELTR4} \ \frac{r_1 \Rightarrow^\uparrow r_2 \qquad r_2 \Rightarrow^\uparrow r_3}{r_1 \Rightarrow^\downarrow r_3}$$

$$\text{ISAPR1} \ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad r_1\cdot \qquad r_1 \Rightarrow^\downarrow r_2}{c_2 \Rightarrow c_4} \qquad \text{ISAPR2} \ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad r_1\cdot \qquad r_1 \Rightarrow^\uparrow r_2}{c_2 \Rightarrow c_3}$$

$$\text{ISAPR3} \ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad \cdot r_1 \qquad r_1 \Rightarrow^\downarrow r_2}{c_1 \Rightarrow c_3} \qquad \text{ISAPR4} \ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad \cdot r_1 \qquad r_1 \Rightarrow^\uparrow r_2}{c_1 \Rightarrow c_4}$$

Fig. 3. Set of inference rules M_1

and the following rules for all relationships r:

$$(o_1, o_2) \in [\![r]\!] \ \text{if} \cdot r_1 \in K^* \wedge d_1 \Rightarrow s(r_1) \in K^* \wedge r_1 \Rightarrow^\downarrow r \in K^* \qquad (12)$$

$$(o_1, o_2) \in [\![r]\!] \ \text{if} \ r_1 \cdot \in K^* \wedge d_1 \Rightarrow t(r_1) \in K^* \wedge r_1 \Rightarrow^\uparrow r \in K^* \qquad (13)$$

$$(o_2, o_1) \in [\![r]\!] \ \text{if} \cdot r_1 \in K^* \wedge d_1 \Rightarrow s(r_1) \in K^* \wedge r_1 \Rightarrow^\uparrow r \in K^* \qquad (14)$$

$$(o_2, o_1) \in [\![r]\!] \ \text{if} \ r_1 \cdot \in K^* \wedge d_1 \Rightarrow t(r_1) \in K^* \wedge r_1 \Rightarrow^\downarrow r \in K^* \qquad (15)$$

$$(o_2, o_2) \in [\![r]\!] \qquad (16)$$

Lemma 1. *Given a schema $S = (F, K)$ such that $d_1 \Rightarrow d_2 \notin K^*$, then the corresponding I^{cl} is an instance of S.*

Proof. We first show that I^{cl} is an instance of F (i.e., the proposition (1) holds). We then show that all constraints in K will hold for I^{cl}. It means that we have to verify that the constraints from (2) to (6) are satisfied.

- Rule (1): $[\![r]\!] \subseteq [\![s(r)]\!] \times [\![t(r)]\!]$. We consider each of the rules that define $[\![r]\!]^{cl}$:
  - Assume that $(o_1, o_2) \in [\![r]\!]^{cl}$ because $\cdot r_1 \in K^*$, $d_1 \Rightarrow s(r_1) \in K^*$ and $r_1 \Rightarrow^\downarrow r \in K^*$ (rule (12)). By ISAPR3, it follows that $s(r_1) \Rightarrow s(r) \in K^*$ and therefore that $d_1 \Rightarrow s(r) \in K^*$, because of rule (TRANS). By the rule (10), it then follows that $o_1 \in [\![s(r)]\!]^{cl}$ and $o_2 \in [\![t(r)]\!]^{cl}$.
  - Assume that $(o_1, o_2) \in [\![r]\!]^{cl}$ because $r_1 \cdot \in K^*$, $d_1 \Rightarrow t(r_1) \in K^*$ and $r_1 \Rightarrow^\uparrow r \in K^*$ (rule (13)). With the same argument as in the previous item, we use ISAPR2 and TRANS to derive that $(o_1, o_2) \in [\![s(r)]\!]^{cl} \times [\![t(r)]\!]^{cl}$.

- • Similar arguments can be made for the rules (14) and (15) using IsAPR1, IsAPR4 and TRANS).
    - • For $(o_2, o_2) \in [\![r]\!]^{cl}$ (rule (16)) it follows from rule (11).
- – Rule (2): if $c_1 \Rightarrow c_2 \in K^*$ then $[\![c_1]\!] \subseteq [\![c_2]\!]$. For the rule (10) that defines $[\![c_1]\!]^{cl}$ it holds that if the right-side holds for $c = c_1$, then it holds for $c = c_2$ if $c_1 \Rightarrow c_2 \in K^*$ because of TRANS.
- – Rule (3): if $r_1 \Rightarrow^{\downarrow} r_2 \in K^*$ then $[\![r_1]\!] \subseteq [\![r_2]\!]$. For each of the rules that define $[\![r]\!]^{cl}$, it holds that if the right-hand side holds for a certain r and $r \Rightarrow^{\downarrow} r' \in K^*$, then by using RELTR1 or RELTR3, it also holds for r'.
- – Rule (4): if $r_1 \Rightarrow^{\uparrow} r_2 \in K^*$ then $[\![r_1]\!]^{-1} \subseteq [\![r_2]\!]$. With the same argument as the previous item, we use the rules RELTR2 and RELTR4.
- – Rule (5): if $\cdot r \in K^*$ then $[\![s(r)]\!] = \{o_1 | (o_1, o_2) \in [\![r]\!]\}$. The $\subseteq$-part is ensured by Prop. (1) and $\supseteq$-part is ensured by RELRFL and rule (12).
- – Rule (6): if $r \cdot \in K^*$ then $[\![t(r)]\!] = \{o_2 | (o_1, o_2) \in [\![r]\!]\}$. Similar to the previous item except we use rule (15).

$\square$

The instance I^{rel} is constructed under the assumption that $r_1 \Rightarrow^{\downarrow} r_2 \notin K^*$. We introduce two objects o_1 and n_2 that are in all classes. Then we add the pair (o_1, o_2) to r_1 and all super-relations of r_1 and the pair (o_2, o_1) to all inverse super-relation of r_1. This leads to the following formal definition:

Definition 7 (Instance I^{rel}). *Given a schema S with $r_1 \Rightarrow^{\downarrow} r_2 \notin K^*$, we define $I^{rel} = (O^{rel}, [\![\cdot]\!]^{rel})$ of S such that $O^{rel} = \{o_1, o_2\}$ and $[\![\cdot]\!]^{rel}$ the smallest function such that $[\![c]\!]^{rel} = \{o_1, o_2\}$ for all classes $c \in C$ and for all relationships $r \in R$:*

$$(o_1, o_2) \in [\![r]\!] \; if \, r_1 \Rightarrow^{\downarrow} r \in K^* \tag{17}$$

$$(o_2, o_1) \in [\![r]\!] \; if \, r_1 \Rightarrow^{\uparrow} r \in K^* \tag{18}$$

$$(o_1, o_1) \in [\![r]\!] \tag{19}$$

$$(o_2, o_2) \in [\![r]\!] \tag{20}$$

Lemma 2. *Given a schema $S = (F, K)$ such that $r_1 \Rightarrow^{\downarrow} r_2 \notin K^*$, then the corresponding I^{rel} is an instance of S.*

Proof. As for the Lemma 1, we prove that I^{rel} is an instance of F and that all constraints in K also hold for I^{rel}.

- – Rule (1): $[\![r]\!] \subseteq [\![s(r)]\!] \times [\![t(r)]\!]$. Since $[\![s(r)]\!]^{rel} = [\![t(r)]\!]^{rel} = \{o_1, o_2\}$ it always holds that $[\![r]\!]^{rel} \subseteq [\![s(r)]\!]^{rel} \times [\![t(r)]\!]^{rel}$.
- – Rule (2): if $c_1 \Rightarrow c_2 \in K^*$ then $[\![c_1]\!] \subseteq [\![c_2]\!]$. $[\![c_1]\!]^{rel} \subseteq [\![c_2]\!]^{rel}$ always holds.
- – Rule (3): if $r_3 \Rightarrow^{\downarrow} r_4 \in K^*$ then $[\![r_3]\!] \subseteq [\![r_4]\!]$. We consider each of the rules that define $[\![r]\!]^{rel}$:
    - • Assume that $(o_1, o_2) \in [\![r_3]\!]^{rel}$ because of $r_1 \Rightarrow^{\downarrow} r_3 \in K^*$ (rule (17)). By rule RELTR1 it follows that $r_1 \Rightarrow^{\downarrow} r_4 \in K^*$ and therefore that $(o_1, o_2) \in [\![r_4]\!]^{rel}$.

- A similar argument can be made for $(o_2, o_1) \in [\![r_3]\!]^{rel}$ because of $r_1 \Rightarrow^\uparrow$ $r_3 \in K^*$ (rule (18)) by using RELTR3.
- For $(o_1, o_1) \in [\![r_3]\!]^{rel}$ and $(o_2, o_2) \in [\![r_3]\!]^{rel}$ because of rule (19) and rule (19), respectively, it follows by the same rule that they are also in $[\![r_4]\!]$.

- Rule (4): if $r_1 \Rightarrow^\uparrow r_2 \in K^*$ then $[\![r_1]\!]^{-1} \subseteq [\![r_2]\!]$. A similar argument as for the previous item is made by using the rules RELTR2 and RELTR4.
- Rule (5): if $\cdot r \in K^*$ then $[\![s(r)]\!] = \{o_1 | (o_1, o_2) \in [\![r]\!]\}$. By definition $[\![s(r)]\!]^{rel} = \{o_1, o_2\}$ and by rules 19 and 20 there are pairs in $[\![r]\!]$ that start with o_1 and o_2.
- Rule (6): similar to the previous item.

□

4.2 Soundness and completeness of rules

In this section we establish the soundness and completeness of the inference rules presented in the preceding sections. By soundness we mean here that if a certain constraint can be derived with the inference rules from K, e.g., $a \Rightarrow b \in K^*$, then it should hold that if an instance I satisfies all constraints in K then it also satisifies $a \Rightarrow b$, i.e., $[\![a]\!] \subseteq [\![b]\!]$. By completeness we mean that if it holds in all instances that satisfy K that a constraint holds, e.g., $[\![a]\!] \subseteq [\![b]\!]$, then it must follow from the inference rules that $a \Rightarrow b \in K^*$. To demonstrate completeness we show the complementary implication: if $a \Rightarrow b \notin K^*$ then we can find an instance that satisifes all constraints in K, but for which $[\![a]\!] \not\subseteq [\![b]\!]$.

Theorem 1. *Given a schema $S = (F, K)$ with K containing only subclass constraints and cardinality constraints and K^* the closure of K under the rules in M_1 then*

1. *$c_1 \Rightarrow c_2 \in K^*$ iff $I \vdash c_1 \Rightarrow c_2$ for all instances I of S,*
2. *$r_1 \Rightarrow^\downarrow r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^\downarrow r_2$ for all instances I of S, and*
3. *$r_1 \Rightarrow^\uparrow r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^\uparrow r_2$ for all instances I of S.*

Proof. The *only-if* part of all the propositions is easily proved by verifying that all the inference rules in M_1 are sound which follows from the semantics of the constraints as defined in Definition 3. The *if* part is proven hereafter for all the propositions.

We consider each constraint of type subclass and show that if it is not in K^*, then it does not hold in at least one of I^{cl} and I^{rel} (which by Lemmas 1 and 2 are instances of S):

1. If $d_1 \Rightarrow d_2 \notin K^*$ then $I^{cl} \nvdash d_1 \Rightarrow d_2$ because $[\![d_1]\!]^{cl} = \{o_1, o_2\}$ and $[\![d_2]\!]^{cl} = \{o_2\}$.
2. If $r_1 \Rightarrow^\downarrow r_2 \notin K^*$ then $I^{cl} \vdash r_1 \Rightarrow^\downarrow r_2$ because $(o_1, o_2) \in [\![r_1]\!]^{rel}$ and $(o_1, o_2) \notin [\![r_2]\!]^{rel}$ and by considering RELRFL.
3. For $r_1 \Rightarrow^\uparrow r_2 \notin K^*$ the proof proceeds similar to that of the previous item.

□

5   Inference rules for deriving cardinality constraints

We are now considering a new set of inference rules M_2 that derive constraints of the forms $\cdot r$ and $r\cdot$. In Figure 4, we give the set of rules M_2. We will assume that K^* is the closure of K under the rules in $M_1 \cup M_2$.

$$\text{RELPR1} \quad \frac{c_1 \xrightarrow{r_1} c_2 \quad c_3 \xrightarrow{r_2} c_4 \quad c_3 \Rightarrow c_1 \quad \cdot r_1 \quad r_1 \Rightarrow^{\downarrow} r_2}{\cdot r_2} \qquad \text{RELPR2} \quad \frac{c_1 \xrightarrow{r_1} c_2 \quad c_3 \xrightarrow{r_2} c_4 \quad c_4 \Rightarrow c_1 \quad \cdot r_1 \quad r_1 \Rightarrow^{\uparrow} r_2}{r_2 \cdot}$$

$$\text{RELPR3} \quad \frac{c_1 \xrightarrow{r_1} c_2 \quad c_3 \xrightarrow{r_2} c_4 \quad c_4 \Rightarrow c_2 \quad r_1 \cdot \quad r_1 \Rightarrow^{\downarrow} r_2}{r_2 \cdot} \qquad \text{RELPR4} \quad \frac{c_1 \xrightarrow{r_1} c_2 \quad c_3 \xrightarrow{r_2} c_4 \quad c_3 \Rightarrow c_2 \quad r_1 \cdot \quad r_1 \Rightarrow^{\uparrow} r_2}{\cdot r_2}$$

Fig. 4. Set of inference rules M_2

5.1   Instance construction

For proving the completeness of inference rules in $M_1 \cup M_2$, we construct the instances I^{tot}, I^{surj} given a schema $S = (F, K)$ with $\cdot r \notin K^*$ and $r\cdot \notin K^*$, respectively. These instances are defined by the same rules as I^{cl} (def. 6) except that we replace d_1 with $s(r)$ for I^{tot} and d_1 with $t(r)$ for I^{surj}.

5.2   Soundness and completeness of rules

Theorem 2. *Given a schema $S = (F, K)$ with K containing only subclass constraints and cardinality constraints and K^* the closure of K under the rules in $M_1 \cup M_2$ then*

1. *$c_1 \Rightarrow c_2 \in K^*$ iff $I \vdash c_1 \Rightarrow c_2$ for all instances I of S,*
2. *$r_1 \Rightarrow^{\downarrow} r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^{\downarrow} r_2$ for all instances I of S,*
3. *$r_1 \Rightarrow^{\uparrow} r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^{\uparrow} r_2$ for all instances I of S,*
4. *$\cdot r \in K^*$ iff $I \vdash \cdot r$ for all instances I of S, and*
5. *$r\cdot \in K^*$ iff $I \vdash r\cdot$ for all instances I of S.*

Proof. The *only-if* part of all the propositions is easily proved by verifying that all the inference rules in $M_1 \cup M_2$ are sound which follows from the semantics of the constraints as defined in Definition 3.

The *if* part proceeds similar to that of Theorem 1. For the subclass constraints the proof proceeds identical. For the cardinality constraints we show that if it is not in K^*, then it does not hold in at least one of I^{tot} and I^{surj}. Because I^{tot} and I^{surj} are equal to I^{cl}, they are instances of S by Lemma 1. What remains to be shown is that if $\cdot r \notin K^*$ ($r \cdot \notin K^*$) then this constraint is not satisfied by I^{tot} (I^{surj}):

- If $\cdot r \notin K^*$ then $I^{tot} \not\vdash \cdot r$. We prove this by contradiction. Assume that $\cdot r \notin K^*$ and $d_1 = s(r)$. Clearly $o_1 \in [\![s(r)]\!]^{tot}$. We consider the only two rules that might add a pair (o_1, o') to $[\![r]\!]^{tot}$:
  - Assume that $(o_1, o_2) \in [\![r]\!]^{tot}$ because $\cdot r_1 \in K^*$, $d_1 \Rightarrow s(r_1) \in K^*$ and $r_1 \Rightarrow^{\downarrow} r \in K^*$. By rule RELPR1, it holds that $\cdot r \in K^*$ which contradicts the assumption.
  - Assume that $(o_1, o_2) \in [\![r]\!]^{tot}$ because $r_1 \cdot \in K^*$, $d_1 \Rightarrow t(r_1) \in K^*$ and $r_1 \Rightarrow^{\uparrow} r \in K^*$. By rule RELPR4, it holds that $\cdot r \in K^*$ which also contradicts the assumption.

  Hence there is no pair $(o_1, o') \in [\![r]\!]^{tot}$ and therefore $I^{tot} \not\vdash \cdot r$.
- If $r \cdot \notin K^*$ then $I^{surj} \not\vdash r \cdot$. The proof proceeds similar to the previous items but with RELPR2 and RELPR3.

$\square$

6   Inference rules for deriving disjointness constraints

In this section we will also consider constraints of the forms $c_1 \not\approx c_2$, $r_1 \not\approx^{\downarrow} r_2$ and $r_1 \not\approx^{\uparrow} r_3$. We give three sets of these rules, namely M_3 (Figure 5) and M_4 (Figure 6). We will assume from now on that K^* is the closure of K under the rules in $M_1, \ldots, M_4$.

With disjointness constraints it is possible to define schemas in which certain classes and relations cannot be populated. To find such conflicts we introduce the following syntactical notion of conflict.

Definition 8 (conflict). *A conflict is a constraint of the form $c \not\approx c$ or $r \not\approx^{\downarrow} r$. If a set of constraints K does not contain such a conflict then it is said to be conflict-free.*

Note that $r \not\approx^{\uparrow} r$ is not a conflict since there are non-empty relations for which it holds.

6.1   Instance construction

For proving the completeness of the inference rules in $M_1, \ldots, M_4$, we construct the instances I^{base}, $I^{\not\approx}$, $I^{\not\approx^{\uparrow}}$ and $I^{\not\approx^{\downarrow}}$ given a schema $S = (F, K)$ with $F = (C, R, s, t)$.

$$\textsc{DsjSym}\ \frac{c_1 \nsim c_2}{c_2 \nsim c_1} \qquad\qquad \textsc{DsjDnSym}\ \frac{r_1 \nsim^\downarrow r_2}{r_2 \nsim^\downarrow r_1}$$

$$\textsc{DsjUpSym}\ \frac{r_1 \nsim^\uparrow r_2}{r_2 \nsim^\uparrow r_1} \qquad\qquad \textsc{DsjInh}\ \frac{c_1 \nsim c_2 \qquad c_3 \Rightarrow c_2}{c_1 \nsim c_3}$$

$$\textsc{DsjInh1}\ \frac{r_1 \nsim^\downarrow r_2 \qquad r_3 \Rightarrow^\downarrow r_2}{r_1 \nsim^\downarrow r_3} \qquad\qquad \textsc{DsjInh2}\ \frac{r_1 \nsim^\downarrow r_2 \qquad r_3 \Rightarrow^\uparrow r_2}{r_1 \nsim^\uparrow r_3}$$

$$\textsc{DsjInh3}\ \frac{r_1 \nsim^\uparrow r_2 \qquad r_3 \Rightarrow^\downarrow r_2}{r_1 \nsim^\uparrow r_3} \qquad\qquad \textsc{DsjInh4}\ \frac{r_1 \nsim^\uparrow r_2 \qquad r_3 \Rightarrow^\uparrow r_2}{r_1 \nsim^\downarrow r_3}$$

$$\textsc{DsjPr1}\ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad c_2 \nsim c_4}{r_1 \nsim^\downarrow r_2} \qquad\qquad \textsc{DsjPr2}\ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad c_2 \nsim c_3}{r_1 \nsim^\uparrow r_2}$$

$$\textsc{DsjPr3}\ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad c_1 \nsim c_3}{r_1 \nsim^\downarrow r_2} \qquad\qquad \textsc{DsjPr4}\ \frac{c_1 \xrightarrow{r_1} c_2 \qquad c_3 \xrightarrow{r_2} c_4 \qquad c_1 \nsim c_4}{r_1 \nsim^\uparrow r_2}$$

Fig. 5. Set of inference rules M_3

Informally we can describe the construction of I^{base} as follows. For each class c we introduce a distinct object o_c that is in c and all its super-classes. For each relation r we introduce the objects o_r^1 (and o_r^2) that are in the source (target) class of r and all its (implied) super-classes. Next we add the pair (o_r^1, o_r^2) to relation r and all its super-relations, and the inverse to all its inverse super-relations. Finally, to satisfy the totalness and surjectivity constraints we add for relations q with such a constraint and each object o in a class c a pair with o and either o_q^1 or o_q^2 to q and its inverse and normal sub-relations. This leads to the following formal definition:

Definition 9 (Instance I^{base}). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ we define $I^{base} = (O^{base}, \llbracket \cdot \rrbracket^{base})$ such that $O^{base} = \{o_c | c \in C\} \cup \{o_r^1, o_r^2 | r \in R\}$ and the interpretation function is defined as the smallest interpretation function that satisfies the following rules for all classes c:*

$$o_d \in \llbracket c \rrbracket \ \text{if}\, d \Rightarrow c \in K^* \tag{21}$$

$$o_r^1 \in \llbracket c \rrbracket \ \text{if}\, r \Rightarrow^\downarrow q \in K^* \wedge s(q) \Rightarrow c \in K^* \tag{22}$$

$$o_r^1 \in \llbracket c \rrbracket \ \text{if}\, r \Rightarrow^\uparrow q \in K^* \wedge t(q) \Rightarrow c \in K^* \tag{23}$$

$$o_r^2 \in \llbracket c \rrbracket \ \text{if}\, r \Rightarrow^\downarrow q \in K^* \wedge t(q) \Rightarrow c \in K^* \tag{24}$$

$$o_r^2 \in \llbracket c \rrbracket \ \text{if}\, r \Rightarrow^\uparrow q \in K^* \wedge s(q) \Rightarrow c \in K^* \tag{25}$$

$$\text{CnflPr1}\ \frac{c_1 \xrightarrow{r} c_2 \quad r \nsim^{\downarrow} r \quad \cdot r}{c_1 \nsim c_1}$$

$$\text{CnflPr2}\ \frac{c_1 \xrightarrow{r} c_2 \quad r \nsim^{\downarrow} r \quad r\cdot}{c_2 \nsim c_2}$$

$$\text{IsaCnfl}\ \frac{c_1 \nsim c_1}{c_1 \Rightarrow c_2}$$

$$\text{IsaDnCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \Rightarrow^{\downarrow} r_2}$$

$$\text{IsaUpCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \Rightarrow^{\uparrow} r_2}$$

$$\text{TotCnfl}\ \frac{c_1 \xrightarrow{r} c_2 \quad c_1 \nsim c_1}{\cdot r}$$

$$\text{SurjCnfl}\ \frac{c_1 \xrightarrow{r} c_2 \quad c_2 \nsim c_2}{r\cdot}$$

$$\text{DisjCnfl}\ \frac{c_1 \nsim c_1}{c_1 \nsim c_2}$$

$$\text{DisjDnCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \nsim^{\downarrow} r_2}$$

$$\text{DisjUpCnfl}\ \frac{r_1 \nsim^{\downarrow} r_1}{r_1 \nsim^{\uparrow} r_2}$$

Fig. 6. Set of inference rules M_4

and the following rules for all relationships r:

$$(o_q^1, o_q^2) \in [\![r]\!] \ if\ q \Rightarrow^{\downarrow} r \in K^* \tag{26}$$

$$(o_q^2, o_q^1) \in [\![r]\!] \ if\ q \Rightarrow^{\uparrow} r \in K^* \tag{27}$$

$$(o, o_q^2) \in [\![r]\!] \ if\ o \in [\![s(q)]\!] \wedge q \Rightarrow^{\downarrow} r \in K^* \wedge \cdot q \in K^* \tag{28}$$

$$(o_q^1, o) \in [\![r]\!] \ if\ o \in [\![t(q)]\!] \wedge q \Rightarrow^{\downarrow} r \in K^* \wedge q\cdot \in K^* \tag{29}$$

$$(o, o_q^1) \in [\![r]\!] \ if\ o \in [\![t(q)]\!] \wedge q \Rightarrow^{\uparrow} r \in K^* \wedge q\cdot \in K^* \tag{30}$$

$$(o_q^2, o) \in [\![r]\!] \ if\ o \in [\![s(q)]\!] \wedge q \Rightarrow^{\uparrow} r \in K^* \wedge \cdot q \in K^* \tag{31}$$

Lemma 3. *Given a schema $S = (F, K)$ such that K^* is conflict-free, then the corresponding I^{base} is an instance of S.*

Proof. We first show that I^{base} is an instance of F (i.e., the proposition (1) holds). We then show that all constraints in K will also hold for I^{base}. It means that we have to verify that the constraints from (2) to (9)) are satisfied.

- Rule (1): $[\![r]\!] \subseteq [\![s(r)]\!] \times [\![t(r)]\!]$. We consider each of the rules that defines $[\![r]\!]^{base}$:
  - Assume that $(o_q^1, o_q^2) \in [\![r]\!]^{base}$ because $q \Rightarrow^{\downarrow} r \in K^*$ (rule (26)). By rule Refl it holds that $s(r) \Rightarrow s(r) \in K^*$ and $t(r) \Rightarrow t(r) \in K^*$. By the rules (22) and (24) it then follows that $o_q^1 \in [\![s(r)]\!]^{base}$ and $o_q^2 \in [\![t(r)]\!]^{base}$, respectively. A similar argument can be made for the pairs (o_q^2, o_q^1) added by rule (27).
  - Assume that $(o, o_q^2) \in [\![r]\!]^{base}$ because $o \in [\![s(q)]\!]^{base}$, $q \Rightarrow^{\downarrow} r \in K^*$ and $\cdot q \in K^*$ (rule (28)). With the same argument as in the previous item it

follows that $o_q^2 \in \llbracket t(r) \rrbracket^{base}$. Similar arguments can be made for the rules (29), (30) and (31).

– Rule (2): If $c_1 \Rightarrow c_2 \in K^*$ then $\llbracket c_1 \rrbracket \subseteq \llbracket c_2 \rrbracket$. For each of the rules that defines $\llbracket c_1 \rrbracket^{base}$ it holds that if the right-hand side holds for $c = c_1$ then it also holds for $c = c_2$ if $c_1 \Rightarrow c_2 \in K^*$, because of rule TRANS.

– Rule (3): If $r_1 \Rightarrow^\downarrow r_2 \in K^*$ then $\llbracket r_1 \rrbracket \subseteq \llbracket r_2 \rrbracket$. For each of the rules that defines $\llbracket r_1 \rrbracket^{base}$ it holds that if the right-hand side holds for a certain r and $r \Rightarrow^\downarrow r' \in K^*$ then, because of RELTR1 and RELTR3, it also holds for r'.

– Rule (4): If $r_1 \Rightarrow^\uparrow r_2 \in K^*$ then $\llbracket r_1 \rrbracket^{-1} \subseteq \llbracket r_2 \rrbracket$. Assume that $(o_q^1, o_q^2) \in \llbracket r_1 \rrbracket^{base}$ because $q \Rightarrow^\downarrow r_1 \in K^*$ (rule (26)). By rule RELTR2 it follows that $q \Rightarrow^\uparrow r_2 \in K^*$ and therefore $(o_q^2, o_q^1) \in \llbracket r_2 \rrbracket^{base}$ by rule (27). A similar argument can be made for each of the rules for $\llbracket r \rrbracket^{base}$, i.e., if the right-hand side is true then by applying RELTR2 or RELTR4 we can derive the right-hand side of the rule that adds the inverse pair.

– Rule (5): If $\cdot r \in K^*$ then $\llbracket s(r) \rrbracket = \{o_1 | (o_1, o_2) \in \llbracket r \rrbracket\}$. The $\subseteq$-part is ensured by Prop. (1) and $\supseteq$-part is ensured by RELRFL and rule (28).

– Rule (6): If $r \cdot \in K^*$ then $\llbracket t(r) \rrbracket = \{o_2 | (o_1, o_2) \in \llbracket r \rrbracket\}$. Similar to the previous item except we use rule (29).

– Rule (7): If $c_1 \nsim c_2 \in K^*$ then $\llbracket c_1 \rrbracket \cap \llbracket c_2 \rrbracket = \varnothing$. We assume the intersection is non-empty and consider each of the rules for $\llbracket c \rrbracket^{base}$:

  • Assume that $o_d \in \llbracket c_1 \rrbracket^{base}$ because $d \Rightarrow c_1 \in K^*$ and $o_d \in \llbracket c_2 \rrbracket^{base}$ because $d \Rightarrow c_2 \in K^*$ (rule (21) and (21)). By DSJSYM and DSJINH it follows that $d \nsim d \in K^*$, which contradicts the assumption that K^* is conflict-free.

  • Assume that $o_r^1 \in \llbracket c_1 \rrbracket^{base}$ because $r \Rightarrow^\downarrow q \in K^*$ and $s(q) \Rightarrow c_1 \in K^*$, and that $o_r^1 \in \llbracket c_2 \rrbracket^{base}$ because $r \Rightarrow^\downarrow q' \in K^*$ and $s(q') \Rightarrow c_1 \in K^*$ (rule (22) and (22)). By DSJINH and DSJPR3 it follows that $q \nsim^\downarrow q' \in K^*$, and by DSJINH1 that $r \nsim^\downarrow r \in K^*$ which contradicts the assumption that K^* is conflict-free. For the combination of rule (22) and (23) we can use DSJPR2, DSJINH3 and DSJINH4 to derive a contradiction. For the combination of rule 23 and 23 we can use DSJPR1, DSJINH2 and DSJINH4 to derive a contradiction. Similar arguments can be made for all the combinations of rule (24) and (25) using DSJPR1, DSJPR2, DSJPR3, DSJINH1, DSJINH2, DSJINH3 and DSJINH4.

Since all other combinations cannot add the same object, it holds that the assumption that the intersection is not empty is false.

– Rule (8): If $r_1 \nsim^\downarrow r_2 \in K^*$ then $\llbracket r_1 \rrbracket \cap \llbracket r_2 \rrbracket = \varnothing$. We assume the intersection is non-empty and consider each of the rules for $\llbracket r \rrbracket^{base}$:

  • For all pairs from the set of rule (26), (28) and (29) it must hold that $q \Rightarrow^\downarrow r_1$ and $q \Rightarrow^\downarrow r_2$. With DSJINH1 we can then derive that $q \nsim^\downarrow q \in K^*$ which contradicts the assumption that K^* is conflict-free.

  • For all pairs from the set of rule (27), (30) and (31) it must hold that $q \Rightarrow^\uparrow r_1$ and $q \Rightarrow^\uparrow r_2$. With DSJINH2 and DSJINH4 we can then derive that $q \nsim^\downarrow q \in K^*$ which contradicts the assumption that K^* is conflict-free.

Since all other combinations cannot add the same pair of objects, it holds that the assumption that the intersection is not empty is false.

– Rule (9): If $r_1 \not\sim^\uparrow r_2 \in K^*$ then $[\![r_1]\!]^{-1} \cap [\![r_2]\!] = \varnothing$. We assume the intersection is non-empty and consider each of the rules for $[\![r]\!]^{base}$. For all combinations that add a pair to $[\![r_1]\!]^{base}$ and its inverse to $[\![r_2]\!]^{base}$ it holds that either $q \Rightarrow^\downarrow r_1, q \Rightarrow^\uparrow r_2 \in K^*$ or $q \Rightarrow^\uparrow r_1, q \Rightarrow^\downarrow r_2 \in K^*$. In both cases we can derive with DsjInh3 and DsjInh4 that $q \Rightarrow^\downarrow q \in K^*$ which contradicts the assumption that K^* is conflict-free.

$\square$

Informally we can describe the construction of $I^\not\sim$ as follows. We assume that $a \not\sim b \notin K^*$. Then we construct the instance as for I^{base} except that we introduce a special object o_{ab} that is placed both in class a and in class b and in all their super-classes. This leads to the following formal definition:

Definition 10 (Instance $I^\not\sim$). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ and $a \not\sim b \notin K^*$ we define $I^\not\sim = (O^\not\sim, [\![\cdot]\!]^\not\sim)$ such that $O^\not\sim = O^{base} \cup \{o_{ab}\}$ and the interpretation as for I^{base} but with the following additional rule:*

$$o_{ab} \in [\![c]\!] \text{ if } a \Rightarrow c \in K^* \vee b \Rightarrow c \in K^* \tag{32}$$

Lemma 4. *Given a schema $S = (F, K)$ such that K^* is conflict-free and $a \not\sim b \notin K^*$, then the corresponding $I^\not\sim$ is an instance of S.*

Proof. The proof proceeds similar to that of Lemma 3 except that for some propositions we need to consider extra cases:

– Rule (2): If $o_{ab} \in [\![c_1]\!]^\not\sim$ because of rule (32) then by Trans it follows that $o_{ab} \in [\![c_2]\!]^\not\sim$ because of the same rule.
– Rule (7): Assume that $o_{ab} \in [\![c_1]\!]^\not\sim$ because $a \Rightarrow c_1 \in K^* \vee b \Rightarrow c_1 \in K^*$ and $o_{ab} \in [\![c_2]\!]^\not\sim$ because $a \Rightarrow c_2 \in K^* \vee b \Rightarrow c_2 \in K^*$. We consider all combinations:
  • If $a \Rightarrow c_1 \in K^*$ and $a \Rightarrow c_2 \in K^*$ then $a \not\sim a \in K^*$ by DsjInh, which contradicts the assumption that K^* is conflict-free. The case where a is replaced with b is similar.
  • If $a \Rightarrow c_1 \in K^*$ and $b \Rightarrow c_2 \in K^*$ then $a \not\sim b \in K^*$ by DsjInh, which contradicts the assumption that $a \not\sim b \notin K^*$. The case where a and b are swapped is similar.

$\square$

Informally we can describe the construction of $I^{\not\sim^\downarrow}$ as follows. We assume that $p \not\sim^\downarrow q \notin K^*$. Then we construct the instance as for I^{base} except that we introduce a special pair (o_{pq}^1, o_{pq}^2) that is placed both in the relation p and in the relation q and in all their super-relations, and the inverse is placed in all the inverse super-relations. This leads to the following formal definition:

Definition 11 (Instance $I^{\not\sim^\downarrow}$). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ and $p \not\sim^\downarrow q \notin K^*$ we define $I^{\not\sim^\downarrow} = (O^{\not\sim^\downarrow}, [\![\cdot]\!]^{\not\sim^\downarrow})$ such that $O^{\not\sim^\downarrow} = O^{base} \cup \{o_{pq}^1, o_{pq}^2\}$*

and the interpretation function as for I^{base} but with the following additional rules for all classes c:

$$o^1_{pq} \in [\![c]\!] \; if \, (p \Rightarrow^\downarrow r \in K^* \vee q \Rightarrow^\downarrow r \in K^*) \wedge s(r) \Rightarrow c \in K^* \qquad (33)$$

$$o^1_{pq} \in [\![c]\!] \; if \, (p \Rightarrow^\uparrow r \in K^* \vee q \Rightarrow^\uparrow r \in K^*) \wedge t(r) \Rightarrow c \in K^* \qquad (34)$$

$$o^2_{pq} \in [\![c]\!] \; if \, (p \Rightarrow^\downarrow r \in K^* \vee q \Rightarrow^\downarrow r \in K^*) \wedge t(r) \Rightarrow c \in K^* \qquad (35)$$

$$o^2_{pq} \in [\![c]\!] \; if \, (p \Rightarrow^\uparrow r \in K^* \vee q \Rightarrow^\uparrow r \in K^*) \wedge s(r) \Rightarrow c \in K^* \qquad (36)$$

and for all relationships r:

$$(o^1_{pq}, o^2_{pq}) \in [\![r]\!] \; if \, p \Rightarrow^\downarrow r \in K^* \vee q \Rightarrow^\downarrow r \in K^* \qquad (37)$$

$$(o^2_{pq}, o^1_{pq}) \in [\![r]\!] \; if \, p \Rightarrow^\uparrow r \in K^* \vee q \Rightarrow^\uparrow r \in K^* \qquad (38)$$

Informally we can describe (o^1_{pq}, o^2_{pq}) as the typical pair that is both in the relationship p and q.

Lemma 5. *Given a schema $S = (F, K)$ such that K^* is conflict-free and $p \not\sim^\downarrow q \notin K^*$, then the corresponding $I^{\not\sim^\downarrow}$ is an instance of S.*

Proof. The proof proceeds similar to that of Lemma 4 and considers the extra cases for Prop. (1), Constr. (7), Constr. (8) and Constr. (9) using the assumption that $p \not\sim^\downarrow q \notin K^*$. □

Definition 12 (Instance $I^{\not\sim^\uparrow}$). *Given a schema $S = (F, K)$ with $F = (C, R, s, t)$ and $p \not\sim^\uparrow q \notin K^*$ we define $I^{\not\sim^\uparrow}$ similar to $I^{\not\sim^\downarrow}$ but here we add a pair $(o^{12}_{pq}, o^{21}_{pq})$ such that it is in $[\![p]\!]^{\not\sim^\uparrow}$ and its inverse, $(o^{21}_{pq}, o^{12}_{pq})$, is in $[\![q]\!]^{\not\sim^\uparrow}$.*

Lemma 6. *Given a schema $S = (F, K)$ such that K^* is conflict-free and $p \not\sim^\uparrow q \notin K^*$, then the corresponding $I^{\not\sim^\uparrow}$ is an instance of S.*

Proof. The proof proceeds similar to that of Lemma 5. □

6.2 Soundness and completeness of rules

Theorem 3. *Given a schema $S = (F, K)$ with K^* the closure of K under the rules in $M_1, \ldots, M_4$ then*

1. *$c_1 \Rightarrow c_2 \in K^*$ iff $I \vdash c_1 \Rightarrow c_2$ for all instances I of S,*
2. *$r_1 \Rightarrow^\downarrow r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^\downarrow r_2$ for all instances I of S,*
3. *$r_1 \Rightarrow^\uparrow r_2 \in K^*$ iff $I \vdash r_1 \Rightarrow^\uparrow r_2$ for all instances I of S,*
4. *$\cdot r \in K^*$ iff $I \vdash \cdot r$ for all instances I of S,*
5. *$r \cdot \in K^*$ iff $I \vdash r \cdot$ for all instances I of S,*
6. *$c_1 \not\sim c_2 \in K^*$ iff $I \vdash r_1 \not\sim r_2$ for all instances I of S,*
7. *$r_1 \not\sim^\downarrow r_2 \in K^*$ iff $I \vdash r_1 \not\sim^\downarrow r_2$ for all instances I of S, and*
8. *$r_1 \not\sim^\uparrow r_2 \in K^*$ iff $I \vdash r_1 \not\sim^\uparrow r_2$ for all instances I of S.*

Proof. The *only-if* part of all the propositions is easily proved by verifying that all the inference rules in $M_1, \ldots, M_4$ are sound, which follows straightforwardly from the semantics of the constraints as defined in Definition 3.

The *if* part is proven in two steps. We first show that for each type of constraint that if K^* is conflict-free then it holds, and then we show that from this it follows that it holds for any K.

Under the assumption that K^* is conflict-free we consider each type of constraint and show that if it is not in K^* then in at least one of I^{base}, $I^{\nsim}$, $I^{\nsim^\downarrow}$ and $I^{\nsim^\uparrow}$ (which by Lemmas 3, 4, 5 and 6 are instances of S) it does not hold:

- If $a \Rightarrow b \notin K^*$ then $I^{base} \nVdash a \Rightarrow b$ because $o_a \in [\![b]\!]^{base}$ iff $a \Rightarrow b \in K^*$.
- If $p \Rightarrow^\downarrow q \notin K^*$ then consider the pair $(o_p^1, o_p^2) \in [\![p]\!]^{base}$. The rules that might add this pair to $[\![q]\!]^{base}$ are (26), (28) and (29), but these all require that $p \Rightarrow^\downarrow q \in K^*$, and therefore $I^{base} \nVdash p \Rightarrow^\downarrow q$.
- If $p \Rightarrow^\uparrow q \notin K^*$ then consider the pair $(o_p^1, o_p^2) \in [\![p]\!]^{base}$. The rules that might add its inverse, i.e.,(o_p^2, o_p^1), to $[\![q]\!]^{base}$ are (27), (30) and (31), but these all require that $p \Rightarrow^\uparrow q \in K^*$, and therefore $I^{base} \nVdash p \Rightarrow^\uparrow q$.
- If $\cdot p \notin K^*$ and $d = s(p)$ then assume there is a pair $(o_d, o) \in [\![p]\!]^{base}$. This pair can only be added by rules (28) and (30). For rule (28) it would follow that $o_d \in [\![s(q)]\!]^{base}$, $q \Rightarrow^\downarrow p \in K^*$ and $\cdot q \in K^*$. Since only rule (21) can put o_d in $[\![s(q)]\!]^{base}$, it follows also that $d \Rightarrow s(q) \in K^*$ and then RELPR1 derives that $\cdot p \in K^*$ which contradicts the assumption. A similar argument holds if the pair is added by rule (30) using RELPR4. Hence there is no pair $(o_d, o) \in [\![p]\!]^{base}$ and therefore $I^{base} \nVdash \cdot p$.
- If $p \cdot \notin K^*$ and $d = t(p)$ then we can argue similarly as in the previous item that a pair $(o, o_d) \in [\![p]\!]^{base}$ cannot exist by considering the rules (29) and 31 and RELPR3 and RELPR2, respectively.
- If $a \nsim b \notin K^*$ then $I^{\nsim} \nVdash a \nsim b$ because $o_{ab} \in [\![a]\!]^{\nsim} \cap [\![b]\!]^{\nsim}$.
- If $p \nsim^\downarrow q \notin K^*$ then $I^{\nsim^\downarrow} \nVdash p \nsim^\downarrow q$ because $(o_{pq}^1, o_{pq}^2) \in [\![p]\!]^{\nsim^\downarrow} \cap [\![q]\!]^{\nsim^\downarrow}$.
- If $p \nsim^\uparrow q \notin K^*$ then $I^{\nsim^\uparrow} \nVdash p \nsim^\uparrow q$ because $(o_{pq}^{21}, o_{pq}^{12}) \in ([\![p]\!]^{\nsim^\uparrow})^{-1} \cap [\![q]\!]^{\nsim^\uparrow}$.

In the next step, we show that the same holds for any K^*. We define the schema $S^{ok} = (F^{ok}, K^{ok})$ as the restriction of S to classes and relationships for which there is no conflict in K^*, i.e., $C^{ok} = \{c \in C | c \nsim c \notin K^*\}$, $R^{ok} = \{r \in R | r \nsim^\downarrow r \in K^*\}$ and K^{ok} is the subset of K^* that mentions only classes in C^{ok} and relationships R^{ok}. This indeed defines a valid schema because if we assume that there is a relationship r in R^{ok} but $s(r)$ (or $t(r)$) is not in C^{ok} then by DSJPR3 (DSJPR1) it follows that $s(r) \nsim s(r) \in K^*$ ($t(r) \nsim t(r) \in K^*$). Moreover, it holds that K^{ok} is closed under the inference rules because it as a subset of K^* and all inference rules are monotone.

As shown in the previous step, it then follows that for each constraint k over F^{ok} that is not in K^{ok} (and therefore also not in K^*) there is an instance of S^{ok} that does not satisfy this constraint. Then we can construct from this an instance I^k of S by letting $[\![c]\!] = \varnothing$ and $[\![r]\!] = \varnothing$ for all classes c and relationships r with conflicts in K^*. To show that this is is indeed an instance of S we have to show that it satisfies all constraints in K. Clearly this holds for those in K^{ok}

so we consider all constraints that mention a class or relationship with a conflict in K^*:

- Consider $a \Rightarrow b \in K$. If $a \nsim a \in K^*$ then this holds trivially. If $b \nsim b \in K^*$ then $a \nsim a \in K^*$ by DSJINH. A similar argument can be made for $p \Rightarrow^\downarrow q \in K$ and $p \Rightarrow^\uparrow q \in K$ using DSJINH1 for the first and DSJINH2 and DSJINH4 for the second.
- Consider $\cdot p \in K$. If $p \nsim^\downarrow p \in K^*$ then by CNFLPR1 we derive $s(p) \nsim s(p) \in K^*$ and it holds trivially. A similar argument holds for $p\cdot$ using CNFLPR2.
- Consider $a \nsim b \in K$. If $a \nsim a \in K^*$ or $b \nsim b \in K^*$ then this holds trivially. A similar argument can be made for $p \nsim^\downarrow q \in K$ and $p \nsim^\uparrow q \in K$.

Clearly the constraint k will not be satisfied by I^k.

In the preceding step we did not consider the constraints that mention a class or relationship that have a conflict in K^*:

- Assume that $a \Rightarrow b \notin K^*$ and $a \nsim a \in K^*$ or $b \nsim b \in K^*$. By DSJINH it follows that always $a \nsim a \in K^*$ and by ISACNFL that $a \Rightarrow b \in K^*$. A similar argument can be made for $p \Rightarrow^\downarrow q$ and $p \Rightarrow^\uparrow q$ using DSJINH1 for the first and DSJINH2 and DSJINH4 for the second to derive that $p \nsim^\downarrow q$ and ISADNCNFL and ISAUPCNFL to derive that have to be in K^*.
- Consider $\cdot p \notin K^*$ and $p \nsim^\downarrow p \in K^*$. Then by CNFLPR1 we derive $s(p) \nsim s(p) \in K^*$ and it follows by TOTCNFL that $\cdot p \in K^*$. A similar argument holds for $p\cdot$ using CNFLPR2 and SURJCNFL.
- Consider $a \nsim b \notin K^*$ and $a \nsim a \in K^*$ or $b \nsim b \in K^*$. If $a \nsim a \in K^*$ then by DISJCNFL and DSJSYM we derive $a \nsim b \in K^*$. A similar argument can be made for $p \nsim^\downarrow q$ and $p \nsim^\uparrow q$ using DISJDNCNFL and DISJDNSYM for the first, and DISJUPCNFL and DISJUPSYM for the second.

$\square$

Corollary 1. *Deciding whether a certain constraint k holds for all instances of a schema S is in PTIME.*

Proof. If there are n classes and m relationships in S then the size of the set of all constraints over S is in $\mathcal{O}(n^2 + m^2)$. Since K^* is always a subset of this and each inference rule can be computed in polynomial time, K^* can be computed in PTIME. $\square$

Corollary 2. *The rules in $M_1, \ldots, M_3$ are complete if the closure of K under $M_1, \ldots, M_3$ is conflict-free.*

Proof. All the rules in M_4 have a conflict in the premise. So if the closure of K under $M_1, \ldots, M_3$ does not contain a conflict then it will be equal to the closure of K under $M_1, \ldots, M_4$. $\square$

Corollary 3. *The rules in $M_1, \ldots, M_3$ are sufficient to detect if K^* is conflict-free.*

Proof. From the preceding corollary it follows that if the closure of K under $M_1, \ldots, M_3$ does not contain a conflict then the closure under $M_1, \ldots, M_4$ will also not contain a conflict. It is also clear that if the first closure contains a conflict then so does the second closure. Therefore the first closure contains a conflict iff the second closure does. $\quad\square$

7   Related work

There has already been a large amount of research on the topic of data integration [4] and reasoning about taxonomies in general [5] and database schemas in particular [6]. As is argued in [7] the two subjects are closely linked together since the ability to reason over the views can be used to check the representation for inconsistencies and redundancies, and to maintain the system in response to changes in the data needs. In particular, [8] presents a reasoning approach for automating a significant part of the schema integration process and [9] relies on a reasoning support to improve the quality of data.

Description Logics (DL) are a well-known family of knowledge representation formalisms that descend from KL-ONE [10]. Long since, they have been applied to data management [11] and information integration [12]. The basic idea is to express database schemas as DL knowledge bases so that DL reasoning techniques can be used to reason about the schema. Although this approach can be restricted to useful fragments where reasoning is still tractable, e.g. [13] and [14], it often already becomes intractable for relatively small fragments [15] and even more so when the concept of *inverse role* is added [16]. It was to the best of our knowledge not yet known that the fragment that is proposed in this paper, which can express such inverse roles, has a relatively simple set of inference rules that is sound and complete and allows tractable reasoning.

8   Conclusion

In this paper we have proposed a view integration method that leaves the original views intact and allows their relationships to be defined by constraints that explicitly express semantic relationships between the components of the different views. Although the motivation of this approach comes from workflow and case management systems where the original views are important for the description of the workflow, this approach can also be beneficial for data integration in more general settings. To support the integration process we have proposed a set of inference rules that allows us to derive implied semantic relationships and especially whether there are conflicts in the integrated schema. We have shown that these sets of rules are sound and complete for all proposed types of constraints, and that subsets of these rules can be already complete for certain subsets of the constraints. Finally it was shown that the inference rules provide in all cases a tractable inference mechanism.

References

1. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. ACM Comput. Surv. **18** (1986) 323–364
2. Shipman, D.W.: The functional data model and the data language DAPLEX. ACM Trans. Database Syst. **6** (1981) 140–173
3. Halpin, T.: Information modeling and relational databases: from conceptual analysis to logical design. Morgan Kaufmann Publishers (2001)
4. Lenzerini, M.: Data integration: A theoretical perspective. In: PODS. (2002) 233–246
5. Bergamaschi, S., Sartori, C.: On taxonomic reasoning in conceptual design. ACM Trans. Database Syst. **17** (1992) 385–422
6. Formica, A., Missikoff, M.: Inheritance processing and conflicts in structural generalization hierarchies. ACM Comput. Surv. **36** (2004) 263–290
7. Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Information integration: Conceptual modeling and reasoning support. In: Proc. of the 6th Int. Conf. on Cooperative Information Systems (CoopIS'98). (1998) 280–291
8. Kashyap, V., Sheth, A.P.: Semantic and schematic similarities between database objects: A context-based approach. VLDB J. **5** (1996) 276–304
9. Calvanese, D., De Giacomo, G., Lenzerini, M., Nardi, D., Rosati, R.: Source integration in data warehousing. In: Proc. of the 9th Int. Workshop on Database and Expert Systems Applications (DEXA'98), IEEE Computer Society Press (1998) 192–197
10. Brachman, R., Schmolze, J.: An overview of the KL-ONE knowledge representation system. Cognitive Science (1985) 171–216
11. Kirk, T., Levy, A.Y., Sagiv, Y., Srivastava, D.: The Information Manifold. In Knoblock, C., Levy, A., eds.: Information Gathering from Heterogeneous, Distributed Environments, Stanford University, Stanford, California (1995)
12. Calvanese, D., Giacomo, G.D., Lenzerini, M., Nardi, D., Rosati, R.: Description logic framework for information integration. In: KR. (1998) 2–13
13. Brachman, R., Levesque, H.: The tractability of subsumption in frame-based description languages. In: AAAI-84, Austin, Texas (1984) 34–37
14. Borgida, A., Brachman, R.J., McGuinness, D.L., Resnick, L.A.: CLASSIC: a structural data model for objects. In: Proc. of the ACM SIGMOD International Conference on Management of Data, Portland, Oregon (1989) 58–67
15. Nutt, W., Donini, F.M., Lenzerini, M., Nardi, D.: The complexity of concept languages. Inf. Comput. **134** (1997) 1–58
16. Horrocks, I., Sattler, U.: A description logic with transitive and inverse roles and role hierarchies. Journal of Logic and Computation **9** (1999) 385–410