Bridging heterogeneous representations of binary relations: first results

Chiara Ghidini[1], Luciano Serafini[1], and Sergio Tessaris[2]

[1] FBK-irst. Via Sommarive 18 Povo, 38050,Trento, Italy
[2] Free University of Bozen - Bolzano. Piazza Domenicani 3. 39100 Bolzano, Italy
ghidini@fbk.eu serafini@fbk.eu tessaris@inf.unibz.it

1 Introduction

In previous work we argued that semantic discrepancies between different ontologies do not reduce to the fact that a concept (relation) in an ontology is somehow related to another concept (relation) in another ontology. On the contrary, semantic discrepancies are likely to be more articulate; e.g. the same "real world" entity can be modelled as a concept or a relation. For example, consider an ontology modelling orders by means of a `PurchaseOrder` class with attributes (i.e. functional roles) holding the details of the actual order (buyer, good, date, delivery option, etc.). In a second ontology not all the details on the purchase orders are modelled, and the information is summarised into the `BoughtBy` role ranging from `Taxable` to `Person`.

This kind of differences are among the so-called *schematic differences*, well studied in schema integration (see, e.g., [1]) but still not deeply investigated in ontology integration. To address this problem we introduced in [3] a Distributed Description Logic (DDL for short) able to represent heterogeneous mappings involving a class and a relation, in addition to mappings between homogeneous components (i.e. concepts into concepts, and roles into roles). This formalism enables the modeller to express the semantic relation between concepts and roles (and vice-versa). For example, the modeller can map the concept `PurchaseOrder` of Ontology 1 and the role `BoughtBy` of Ontology 2 via, so-called, *heterogeneous bridge rules* like

$$1 : \texttt{PurchaseOrder} \xrightarrow{\sqsupseteq}\!\!\!\!\rightarrow 2 : \texttt{BoughtBy} \qquad (1)$$

stating that the class `PurchaseOrder` of Ontology 1 is *more specific* than the role `BoughtBy` of Ontology 2.

In this paper we extend the formalism of DDL further in order to represent additional mappings involving concepts and relations. In particular we study how to extend heterogeneous mappings in order to express the fact that a concept in an ontology is the *reification* of a relation contained in another ontology. The main idea is the following: let us assume for the moment that `PurchaseOrder` and `BoughtBy` belong to the same ontology. If the class `PurchaseOrder` reifies the relation `BoughtBy`, by means of two (functional) roles `Buyer` and `Good`, then we have that:

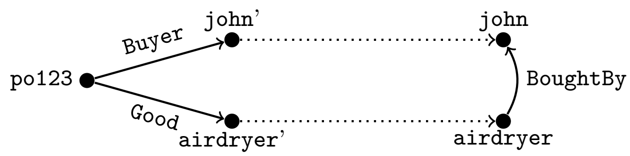

Fig. 1. Reification in distributed ontologies.

(a) instances d of the concept PurchaseOrder represent pairs of objects (d_1, d_2) of the role BoughtBy; and additionally
(b) roles Buyer and Good exist, and connect the PurchaseOrder d representing the BoughtBy pair (d_1, d_2) to the two components d_1 and d_2 of the pair.

Let us examine the case in which PurchaseOrder and BoughtBy belong to different ontologies. If we want to model the fact that the class PurchaseOrder in Ontology 1 reifies the relation BoughtBy in Ontology 2, then we can maintain condition (a), which is captured in DDL by heterogeneous bridge rules like (1), but we have to modify condition (b) slightly. In fact, assume that po123 is the purchase order in Ontology 1 reifying the pair (airdryer, john) of BoughtBy in Ontology 2. Since the two ontologies are interpreted over different domains, we cannot require (or force) that the two objects connected to po123 via Buyer and Good are exactly airdryer and john, as they belong to a different domain of interpretation. Nevertheless domains can be related, as represented by the dotted arrows in Figure 1. We therefore "relax" the property (b) into the following

(b') two (functional) roles exist, and connect the PurchaseOrder d representing the BoughtBy pair (d_1, d_2) to two objects d'_1 and d'_2 which "correspond" (in a sense to be specified) to the objects d_1 and d_2 in Ontology 2 (see Figure 1).

In this paper we develop this intuition into a logic framework, providing a DDL enriched with new bridge rules, called *reification bridge rules*, able to formalise the aspects (a) and (b') above. We provide a semantics for the reification bridge rule, different examples of how to use them to model heterogeneous representations of binary relations, and a sound and complete axiomatisation of the effects of all mappings from a source ontology to a target ontology. This is the crucial step towards the proof of completeness for an arbitrary network of ontologies in the style as the one presented in [6].

2 DDL Syntax

Given a non empty set I of indexes, used to identify ontologies, let $\{\mathcal{D}L_i\}_{i \in I}$ be a collection of description logics. For each $i \in I$ let us denote a T-box of $\mathcal{D}L_i$ as $\mathcal{T}_i$. In this paper, we assume that each $\mathcal{D}L_i$ is weaker or at most equivalent to $\mathcal{SHIQ}$, enriched with role union and difference. Namely, we allow complex role expressions of the form $R \sqcup S$, and $P \sqcap \neg(Q_1 \sqcup Q_2 \sqcup \ldots \sqcup Q_n)$. In [8] it is shown as this language can be encoded in $\mathcal{ALCQI}_b$, which corresponds to $\mathcal{ALCQI}$ with role union, conjunction and difference (see [8]).

We call $\mathbf{T} = \{\mathcal{T}_i\}_{i \in I}$ a family of T-Boxes indexed by I. To make every description distinct, we prefix it with the index of ontology it belongs to. For instance, the concept C that occurs in the i-th ontology is denoted as $i : C$.

Semantic mappings between different ontologies are expressed via collections of *bridge rules*. In the following we use A, B, C and D as place-holders for concepts and R, S, P and Q as place-holders for roles. We instead use X and Y to denote both concepts and roles.

Definition 1 (Bridge rule). *A bridge rule from i to j is defined as follows:*

– homogeneous bridge rule

$$i : X \xrightarrow{\sqsubseteq} j : Y \qquad \text{(into bridge rule)} \qquad (2)$$

$$i : X \xrightarrow{\sqsupseteq} j : Y \qquad \text{(onto bridge rule)} \qquad (3)$$

– heterogeneous bridge rule

$$i : C \overset{\sqsubseteq}{\twoheadrightarrow} j : R \qquad \text{(concept-into-role bridge rule)} \qquad (4)$$

$$i : C \overset{\sqsupseteq}{\twoheadrightarrow} j : R \qquad \text{(concept-onto-role bridge rule)} \qquad (5)$$

$$i : R \overset{\sqsubseteq}{\twoheadrightarrow} j : C \qquad \text{(role-into-concept bridge rule)} \qquad (6)$$

$$i : R \overset{\sqsupseteq}{\twoheadrightarrow} j : C \qquad \text{(role-onto-concept bridge rule)} \qquad (7)$$

– reification bridge rule

$$i : P \twoheadrightarrow j : R^{\#k} \qquad \text{(attribute to role bridge rule)} \qquad (8)$$

$$i : R^{\#k} \twoheadrightarrow j : P \qquad \text{(role to attribute bridge rule)} \qquad (9)$$

where X and Y in a bridge rule are either both concepts, or both roles, C is a concept, R and P are atomic roles and $k \in \{1, 2\}$.

Bridge rule (2) states that, from the j-th point of view the concept (role) X in i is less general than its local concept (role) Y. Similarly, the onto bridge rule (3) expresses the fact that, according to j, concept (role) X in i is more general than its own concept (role) Y. Bridge rules (4)–(7) express similar mappings, but involve heterogeneous elements of the ontologies. For instance, bridge rule (4) states that, from the j-th point of view the concept C in i is less general than its local role R. Bridge rules (8) and (9) are novel bridge rules introduced in this paper. (8) is used to express the fact that, from the j-th point of view, the role P in ontology i corresponds to the k-th argument of its own relation R. Vice-versa bridge rule (9) expresses the fact that, from the j-th point of view, the k-th argument of the relation R in ontology i correspond to role P in j. Bridge rules (8) and (9) can be used, together with the heterogeneous bridge rules to express reification mappings between different ontologies. For instance, the bridge rules

$$1 : \texttt{PurchaseOrder} \overset{\equiv}{\twoheadrightarrow} 2 : \texttt{BoughtBy} \qquad (10)$$

$$1 : \texttt{Good} \twoheadrightarrow 2 : \texttt{BoughtBy}^{\#1} \qquad (11)$$

$$1 : \texttt{Buyer} \twoheadrightarrow 2 : \texttt{BoughtBy}^{\#2} \qquad (12)$$

will be used in Section 4 to express the reification relation between `PurchaseOrder` and `BoughtBy` introduced in Section 1. Notationally, $i : X \stackrel{\equiv}{\longrightarrow} j : Y$ indicates the existence of both an into and an onto bridge rule between $i : X$ and $j : Y$.

A *distributed T-box* (DTB) $\mathfrak{T} = \langle \{\mathcal{T}_i\}_{i \in I}, \mathfrak{B} \rangle$ consists of a collection $\{\mathcal{T}_i\}_{i \in I}$ of T-boxes, and a collection $\mathfrak{B} = \{\mathfrak{B}_{ij}\}_{i \neq j \in I}$ of bridge rules between them.[3]

3 DDL Semantics

The semantic of DDL assigns to each ontology $\mathcal{T}_i$ a *local interpretation domain*. The first component of an interpretation of a DTB is a family of interpretations $\{\mathcal{I}_i\}_{i \in I}$, one for each T-box $\mathcal{T}_i$. Each $\mathcal{I}_i$ is called a *local interpretation* and consists of a *possibly empty* domain $\Delta^{\mathcal{I}_i}$ and a valuation function $\cdot^{\mathcal{I}_i}$, which maps every concept to a subset of $\Delta^{\mathcal{I}_i}$, and every role to a subset of $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_i}$. The interpretation on the empty domain is used to provide a semantics for distributed T-boxes in which some of the local T-boxes are inconsistent. The reader interested in this aspect of DDL can refer to [6].

The second component of the DDL semantics are families of domain relations. Domain relations define how the different T-box interact and are necessary to define the satisfiability of bridge rules.

Definition 2 (Domain relation). *A* domain relation r_{ij} *from* i *to* j *is a subset of* $\Delta^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$.

Domain relations are used to interpret homogeneous bridge rules and are illustrated in detail in [6], but do not provide sufficient information to interpret heterogeneous bridge rules. As an example, in order to evaluate the heterogeneous bridge rule (10) we would like to map a purchase order, say order `po123`, of ontology 1 into a triple of the form

$$\texttt{BoughtBy}[\texttt{airdryer}, \texttt{john}]$$

composed of elements of Ontology 2, with the intuitive meaning that `po123` is the purchase order in ontology 1 which correspond to the bought by relation holding between `airdryer` and `john` in Ontology 2.

Let us formally introduce a triple $R[d_1, d_2]$. Let $\mathcal{I}_i$ be a *local interpretation* for $\mathcal{DL}_i$. Let $\mathcal{R}$ be the set of all atomic roles of $\mathcal{DL}_i$. We indicate with $[\mathcal{R}]^{\mathcal{I}_i}$ the set of all triples $R[d_1, d_2]$ such that $R \in \mathcal{R}$ and $(d_1, d_2) \in R^{\mathcal{I}_i}$. We call $[\mathcal{R}]^{\mathcal{I}_i}$ the set of *admissible triples* for $\mathcal{I}_i$. Given a role $R \in \mathcal{R}$, we write $[R]^{\mathcal{I}_i}$ to denote all the admissible triples in $[\mathcal{R}]^{\mathcal{I}_i}$ of the form $R'[d_1, d_2]$ with $R' \sqsubseteq R$.

Definition 3 (Concept-role and role-concept domain relation). *A* concept-role domain relation cr_{ij} *from* i *to* j *is a subset of* $\Delta^{\mathcal{I}_i} \times [\mathcal{R}]^{\mathcal{I}_j}$ *such that if* $(d, R'[d_1, d_2]) \in cr_{ij}$ *and* $R'^{\mathcal{I}_j} \subseteq R^{\mathcal{I}_j}$ *with* R *atomic role, then* $(d, R[d_1, d_2]) \in cr_{ij}$. *A* role-concept domain relation rc_{ij} *from* i *to* j *is a subset of* $[\mathcal{R}]^{\mathcal{I}_i} \times \Delta^{\mathcal{I}_j}$ *such that if* $(R'[d_1, d_2], d) \in rc_{ij}$ *and* $R'^{\mathcal{I}_i} \subseteq R^{\mathcal{I}_i}$ *with* R *atomic role, then* $(R[d_1, d_2], d) \in rc_{ij}$.

[3] As in [3] we require that for every bridge rule between roles $i : P \longrightarrow j : R$ in $\mathfrak{B}_{ij}$, also $i : inv(P) \longrightarrow j : inv(R)$ is in $\mathfrak{B}_{ij}$ (where $inv(X)$ is the inverse of X).

The domain relation cr_{ij} represents a possible way of mapping elements of $C^{\mathcal{I}_j}$ into pairs of $R^{\mathcal{I}_i}$, seen from j's perspective. For instance,

$$(\texttt{po123}, \texttt{BoughtBy}[\texttt{airdryer}, \texttt{john}]) \in cr_{12} \tag{13}$$

represents the fact that `po123` is an object in ontology 1 corresponding to the `BoughtBy` relation between `airdryer` and `john` in ontology 2. The additional condition on $R^{\mathcal{I}_j} \subseteq R'^{\mathcal{I}_j}$ above is used to ensure that cr_{ij} is consistent with the hierarchy of roles. Analogously for rc_{ij}.

Definition 4 (Distributed interpretation). *A* distributed interpretation $\mathfrak{I}$ *of a DTB* $\mathfrak{T}$ *consists of the 4-tuple* $\langle \{\mathcal{I}_i\}_{i \in I}, \{r_{ij}\}_{i \neq j \in I}, \{cr_{ij}\}_{i \neq j \in I}, \{rc_{ij}\}_{i \neq j \in I} \rangle$.

In order to define the satisfiability of bridge rules we introduce some functional notation for domain relations and for roles. Given a (regular, concept-role, role-concept) domain relation dr_{ij}, we write $dr_{ij}(t)$ to denote the set of objects (elements of the domains or triples) t' such that (t, t') is in dr_{ij}. Analogously, given a set $T = \{t_1, t_2, \ldots\}$, we write $dr_{ij}(T)$ to denote the union of all $dr_{ij}(t_i)$ with $t_i \in T$ (we use the same notation with the inverse dr_{ij}^{-} of the relation). Finally, given a role P, we use $P^{\mathcal{I}_i}(t)$ to denote the set of objects t' such that (t, t') belongs to $P^{\mathcal{I}_i}$.

Definition 5 (Satisfiability of bridge rules). *A distributed interpretation* $\mathfrak{I}$ *satisfies a bridge rule* br*, written as* $\mathfrak{I} \models br$*, when*

– *homogeneous bridge rules*

$$\mathfrak{I} \models i : X \xrightarrow{\sqsubseteq} j : Y \quad \textit{if} \quad r_{ij}(X^{\mathcal{I}_i}) \subseteq Y^{\mathcal{I}_j} \tag{14}$$

$$\mathfrak{I} \models i : X \xrightarrow{\sqsupseteq} j : Y \quad \textit{if} \quad r_{ij}(X^{\mathcal{I}_i}) \supseteq Y^{\mathcal{I}_j} \tag{15}$$

– *heterogeneous bridge rules*

$$\mathfrak{I} \models i : C \overset{\sqsubseteq}{\twoheadrightarrow} j : R \quad \textit{if} \quad cr_{ij}(C^{\mathcal{I}_i}) \subseteq [R]^{\mathcal{I}_j} \tag{16}$$

$$\mathfrak{I} \models i : C \overset{\sqsupseteq}{\twoheadrightarrow} j : R \quad \textit{if} \quad cr_{ij}(C^{\mathcal{I}_i}) \supseteq [R]^{\mathcal{I}_j} \tag{17}$$

$$\mathfrak{I} \models i : R \overset{\sqsubseteq}{\twoheadrightarrow} j : C \quad \textit{if} \quad rc_{ij}([R]^{\mathcal{I}_i}) \subseteq C^{\mathcal{I}_j} \tag{18}$$

$$\mathfrak{I} \models i : R \overset{\sqsupseteq}{\twoheadrightarrow} j : C \quad \textit{if} \quad rc_{ij}([R]^{\mathcal{I}_i}) \supseteq C^{\mathcal{I}_j} \tag{19}$$

– *reification bridge rules*

$$\mathfrak{I} \models i : P \twoheadrightarrow j : R^{\#k} \ \textit{if for all} \ (d, R[d_1, d_2]) \in cr_{ij}, P^{\mathcal{I}_i}(d) \subseteq r_{ij}^{-1}(d_k) \tag{20}$$

$$\mathfrak{I} \models i : R^{\#k} \twoheadrightarrow j : P \ \textit{if for all} \ (R[d_1, d_2], d) \in rc_{ij}, P^{\mathcal{I}_j}(d) \subseteq r_{ij}(d_k) \tag{21}$$

Satisfiability of into bridge rules forces the appropriate domain relation to map objects of the left hand side element of the bridge rule into objects of the right hand side element. Analogously all the onto bridge rules ensure that each

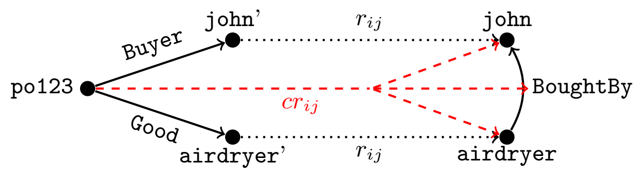

Fig. 2. Semantics of reification rule (8).

object of the right hand side element has at least a pre-image, via the appropriate domain relation, which is in the left hand side element of the rule.

Satisfiability of reification bridge rules establishes how concept-role (role-concept) domain relations interact with the regular domain relation r_{ij}. For instance, if object `po123` in Ontology 1 reifies the pair (`airdryer`, `john`) in Ontolgy 2 (via cr_{12}), then the bridge rule (11) forces the domain relation r_{12} to relate the object(s) connected `po123` via the attribute `Good`, with the first component of the pair (`airdryer`, `john`). Analogously with the reification rule (11), which relates the attribute `Buyer` to the second component of the pair, as shown in Figure 2. The meaning of the reification rule (9) is similar.

A distributed interpretation $\mathfrak{I}$ *satisfies* DTB $\mathfrak{T}$ if all the T-boxes $\mathcal{T}_i$ are satisfied by their local interpretation $\mathcal{I}_i$, and if $\mathfrak{I}$ satisfies all the bridge rules in $\mathfrak{B}_{ij}$. Entailment and satisfiability of a single concept are defined in the usual way by means of the satisfiability of a distributed T-Box. The reader interested in the formal definitions can refer to [3].

4 Reification and Undecidability results

Among the driving motivations to enrich the expressiveness of bridge rules by means of the introduction of the so called *reification* rules there is the possibility of fully capture reification. In fact, heterogeneously relating roles and concepts doesn't capture the key reification modelling practice of using attributes to represent the participants in a given tuple (e.g. see [5]).

As described in the previous sections, the reification bridge rules serve exactly to this purpose. In fact, they relate roles and "projections" of relations (in this case just binary, since they are roles) along the first or the second components. Note that the given semantics does more than simply relating domains and ranges of the involved roles. This is enforced by the appropriate involvement of the concept to role and role to concept relations.

In this section we show, by means of reification examples and by showing that the unconstrained language leads to undecidability, that the proposed semantics genuinely increase the expressiveness of the language.

Let us consider the purchase order example introduced in Section 1. In Ontology 1 we assume also the "reification" axiom $\texttt{PurchaseOrder} \sqsubseteq {=}1\texttt{Buyer}.\texttt{Customer} \sqcap {=}1\texttt{Good}.\texttt{Hardware}$ which states that `PurchaseOrder` has two mandatory functional roles and specifies their range, while in Ontology 2 we still have the two concepts $2:$ `Person` and $2:$ `Taxable` but we have no information on the domain and range of $2 : \texttt{BoughtBy}$. Let us connect the two ontologies via bridge rule (10), which maps concept $1 : \texttt{PurchaseOrder}$ into role $2 : \texttt{BoughtBy}$, plus the following bridge rules:

$$1 : \texttt{Customer} \xrightarrow{\sqsubseteq} 2 : \texttt{Person} \qquad\qquad 1 : \texttt{Hardware} \xrightarrow{\sqsubseteq} 2 : \texttt{Taxable}$$

which map concepts $1 : \texttt{Customer}$, $1 : \texttt{Hardware}$ into concepts $2 : \texttt{Person}$ and $2 : \texttt{Taxable}$ respectively. With these rules only, nothing can be said about the domain and range of $2 : \texttt{BoughtBy}$. This because the combination of homogeneous and heterogeneous bridge rules alone does not generate any effect, since the domain relation and the concept-role domain relations do not affect each other. However, the addition of the reification rules (11) and (12) makes the different domain relations interact and forces the domain and range of `BoughtBy` to `Taxable` and `Person`, respectively.

An analogous situation that shows the flexibility of the framework is when the role, on one side of the heterogeneous bridge rule, has additional constraints incompatible with the concept representing its reification. Let us assume an ontology in which a marriage is represented by means of a concept $1 : \texttt{Marriage}$ with two attributes $1 : \texttt{Husband}$ and $1 : \texttt{Wife}$ restricted to two disjoint concepts $1 : \texttt{Male}$ and $1 : \texttt{Female}$. On the other side we have a *symmetric* role $2 : \texttt{MarriedTo}$ whose domain and range are the union of the disjoint concepts $2 : \texttt{Male}$ and $2 : \texttt{Female}$. If we proceed as in the last example with the rules below, $2 : \texttt{MarriedTo}$ is going to be empty in any model for the distributed system.

$$1 : \texttt{Marriage} \xrightarrow{\equiv}\!\!\!\!\rightarrow 2 : \texttt{MarriedTo} \tag{22}$$

$$1 : \texttt{Wife} \twoheadrightarrow 2 : \texttt{MarriedTo}^{\#1} \qquad 1 : \texttt{Female} \xrightarrow{\sqsubseteq} 2 : \texttt{Female} \tag{23}$$

$$1 : \texttt{Husband} \twoheadrightarrow 2 : \texttt{MarriedTo}^{\#2} \qquad 1 : \texttt{Male} \xrightarrow{\sqsubseteq} 2 : \texttt{Male} \tag{24}$$

The reason is that if a pair (x, y) is in the local interpretation of $2 : \texttt{MarriedTo}$, then the bridge rules (22) and (23) force x to be in the interpretation of $2 : \texttt{Female}$. But because of symmetry also the pair (y, x) is in the local interpretation of $2 : \texttt{MarriedTo}$ and the bridge rules (22) and (24) force x to be also in the interpretation of $2 : \texttt{Male}$, which is disjoint from $2 : \texttt{Female}$.

The problem is that the concept $1 : \texttt{Marriage}$ does not represent exactly the reification of $2 : \texttt{MarriedTo}$, but of a non-symmetric sub-role of $2 : \texttt{MarriedTo}$ in which domain and range are disjoint. The bridge rules, as stated in (22)–(24), do not capture correctly this situation. Obviously, there are several ways of "fixing" the problem: the first is to add an atomic role $2 : \texttt{MarriedTo}_{\texttt{Female}}$ indicating the domain restriction of `MarriedTo` over the concept `Female` and use this in rules (22)–(24).[4] However, in the spirit of DDL we should adopt a method which doesn't require modifications in the local ontologies. In particular, the flexibility of bridge rules enables the relaxation of rule (22) into the following

$$1 : \texttt{Marriage} \xrightarrow{\sqsubseteq} 2 : \texttt{MarriedTo}$$

[4] We can encode the domain restriction of R over C, often expressed as $R \upharpoonright C$, as $R \upharpoonright C \sqsubseteq R$, $C \sqsubseteq \neg\exists(R \sqcap \neg R \upharpoonright C)$, and $\exists R \upharpoonright C \sqsubseteq C$.

which states that $1:$ `Marriage` is more specific that $2:$ `MarriedTo`. This rule, in fact, does not force the existence of an object in $1:$ `Marriage` for each pair in $2:$ `MarriedTo`.

A strong indication that the proposed semantics properly captures the reification mechanism can be understood in the fact that it enables the encoding of role axioms of the form $R \circ S \sqsubseteq T$. Note that the adopted Description Logics doesn't allow axioms of that sort. The encoding can be done by means the following set of bridge rules, which is equivalent to of the axiom $R \circ S \sqsubseteq T$ asserted in the second ontology.

$$1 : T'^{\#1} \twoheadrightarrow 2 : R^- \qquad\qquad 1 : T'^{\#2} \twoheadrightarrow 2 : S$$

$$1 : T' \xrightarrow{\sqsubseteq} 2 : T \qquad\qquad 1 : T' \overset{\exists}{\twoheadrightarrow} 2 : \exists R^-.\top \sqcap \exists S.\top$$

To show that these axioms imply the above role axiom we should consider the case in which the local interpretation for 2 contains both $R^-(x, x_1)$ and $S(x, x_2)$. Because of the fourth bridge rule there should be a pair $(T'[y_1, y_2], x)$ in rc_{12}, where $T'[y_1, y_2]$ is an admissible triple (i.e. $T'(y_1, y_2)$ is in the local interpretation of 1). The reification rules ensure that x_1 and x_2 are in the image of y_1 and y_2 respectively (i.e. $x_i \in r_{1,2}(y_i)$). Finally, the third (homogeneous) rule implies that each pair in $r_{1,2}(y_1) \times r_{1,2}(y_2)$ is in the interpretation of T; this ends the proof, showing that $T(x_1, x_2)$ is satisfied in the local interpretation of 2.

This expressiveness shows that the semantics really captures the intuition behind reification; however, it shows also that the satisfiability problem for the resulting DDL is undecidable. In fact, even simple Description Logics which allow unrestricted axioms of the form $R \circ S \sqsubseteq T$ are undecidable (see [9]). In order to guarantee decidability, we should restrict the interaction of the bridge rules.

5 The effects of bridge rules

Bridge rules can be thought of as inter-theory axioms, which constrain the models of the theories representing the different ontologies. An important characteristic of mappings specified by DDL bridge rules is that they are *directional*, in the sense that they are defined from a source ontology to a target ontology, and they allow to transfer knowledge only from the source to the target, without any undesired back-flow effect. Here we start by characterising the effects of mappings of a simple DTB $\langle \mathcal{T}_i, \mathcal{T}_j, \mathfrak{B}_{ij} \rangle$, composed of two T-boxes $\mathcal{T}_i$ (the source) and $\mathcal{T}_j$ (the target) and a set of bridge rules $\mathfrak{B}_{ij}$ from i to j, and we prove that DDL extended with reification bridge rules still fulfils the requirement of directionality:

Proposition 1. $\langle \mathcal{T}_i, \mathcal{T}_j, \mathfrak{B}_{ij} \rangle \models i : X \sqsubseteq Y$ *if and only if* $\mathcal{T}_i \models X \sqsubseteq Y$

The proof can be found in [4]. According to Proposition 1, bridge rules from i to j affect only the logical consequences in j, and leave the consequences in i unchanged. In the following we characterise the knowledge propagated from i (the source) to j (the target) using a set of *propagation rules* of the form:

$$\frac{\begin{array}{l}\text{bridge rules from } i \text{ to } j\\ \text{axioms in } i\end{array}}{\text{axiom in } j}$$

which must be read as: if $\mathcal{T}_i$ entails all the axioms in i, and $\mathfrak{B}_{ij}$ contains the bridge rules from i to j, then $\langle \mathcal{T}_i, \mathcal{T}_j, \mathfrak{B}_{ij}\rangle$ satisfies axioms in j.

An extensive description of the effects of homogeneous and heterogeneous bridge rules can be found in [3]. In this section we investigate the additional effects induced by the reification bridge rules.

Simple propagation rules which describe the effects of the reification bridge rules in a scenario in which concepts are mapped to roles are:

$$\frac{\begin{array}{l} i : C \xrightarrow{\exists}\!\!\!\!\rightarrow j : R \\ i : P \rightarrow j : R^{\#1} \\ i : A \xrightarrow{\sqsubseteq} j : B \\ i : C \sqsubseteq \exists P.A \end{array}}{j : \exists R.\top \sqsubseteq B} \quad (25)$$

$$\frac{\begin{array}{l} i : C \xrightarrow{\exists}\!\!\!\!\rightarrow j : R \\ i : Q \rightarrow j : R^{\#2} \\ i : A \xrightarrow{\sqsubseteq} j : B \\ i : C \sqsubseteq \exists Q.A \end{array}}{j : \top \sqsubseteq \forall R.B} \quad (26)$$

$$\frac{\begin{array}{l} i : C \xrightarrow{\exists}\!\!\!\!\rightarrow j : R \\ i : P_1 \rightarrow j : R^{\#1} \\ i : P_2 \rightarrow j : R^{\#2} \\ i : A_1 \xrightarrow{\sqsubseteq} j : B_1 \\ i : A_2 \xrightarrow{\sqsubseteq} j : B_2 \\ i : C \sqsubseteq \exists P_1.A_1 \sqcup \exists P_2.A_2 \end{array}}{j : \exists R.\neg B_2 \sqsubseteq B_1} \quad (27)$$

These effects propagate knowledge from i to j and allow to infer information over the domain and range of the role $i : R$. Let us go back to the purchase order example of Section 4. Rules (25) and (26) are the rules that allow to infer that $2 : \texttt{Taxable}$ and $2 : \texttt{Person}$ are the domain and range of `BoughtBy`, respectively. Note that, we still obtain these inferences if we relax the "reification" axiom to weaker axioms which state only the existence of the roles `Buyer` and `Good` and their ranges, and omit functionality. Effect (27) considers the scenario in which we can guarantee, for each `PurchaseOrder` d, the existence of at least one role among `Buyer` and `Good`, written as $\texttt{PurchaseOrder} \sqsubseteq \exists \texttt{Buyer.Customer} \sqcup \exists \texttt{Good.Hardware}$. In this case we obtain that for each pair (d_1, d_2) of `BoughtBy`, we can classify at least one among d_1 and d_2 and say that d_1 is a `Taxable` or d_2 is a `Person`.

Simple effects, which can be considered the "counterpart" of (25)–(27) in a scenario in which roles are mapped to concepts, are:

$$\frac{\begin{array}{l} i : R \xrightarrow{\exists}\!\!\!\!\rightarrow j : C \\ i : R^{\#1} \rightarrow j : P \\ i : A \xrightarrow{\sqsubseteq} j : B \\ i : \exists R.\top \sqsubseteq A \end{array}}{j : C \sqsubseteq \forall P.B} \quad (28)$$

$$\frac{\begin{array}{l} i : R \xrightarrow{\exists}\!\!\!\!\rightarrow j : C \\ i : R^{\#2} \rightarrow j : P \\ i : A \xrightarrow{\sqsubseteq} j : B \\ i : \top \sqsubseteq \forall R.A \end{array}}{j : C \sqsubseteq \forall P.B} \quad (29)$$

$$\frac{\begin{array}{l} i : R \xrightarrow{\exists}\!\!\!\!\rightarrow j : C \\ i : R^{\#1} \rightarrow j : P_1 \\ i : R^{\#2} \rightarrow j : P_2 \\ i : A_1 \xrightarrow{\sqsubseteq} j : B_1 \\ i : A_2 \xrightarrow{\sqsubseteq} j : B_2 \\ j : \exists R.\neg A_2 \sqsubseteq A_1 \end{array}}{i : C \sqsubseteq \forall P_1.B_1 \sqcup \forall P_2.B_2} \quad (30)$$

Bridge rule (28) allow to propagate information about the domain of the relation R, to all the P-successors of C, when P corresponds to the first component of the pairs in R. Analogously (29) propagates the information about the range of R, if P corresponds to the second component of the pairs in R. Finally, (30) is the analogous of the effect described in (27). In fact if only know how to classify (at least) one of the two components of the relation R, then we can infer knowledge about at least one of the P_i-successors of C (if any).

In addition to the effects shown above, imposing reification bridge rules from roles to concepts also allows to infer the following effect

$$\begin{array}{l} i : R \xrightarrow{\sqsupseteq}\!\!\!\!\rightarrow C \\ i : R^{\#1} \twoheadrightarrow P_1 \\ i : R^{\#2} \twoheadrightarrow P_2 \\ i : R \xrightarrow{\sqsubseteq} j : Q \\ \hline j : P_1^{-} \circ P_2 \restriction C \sqsubseteq Q \end{array} \tag{31}$$

which correspond to the encoding of axioms of the form $R \circ S \sqsubseteq T$ discussed in Section 4. $P_2 \restriction C$ expresses the domain restriction of P_2 over C (see Section 4 for the encoding).

The general form of the propagation rules is given in [4] and is omitted here for lack of space. Given two T-boxes $\mathcal{T}_i$ and $\mathcal{T}_j$, and a set of bridge rules $\mathfrak{B}_{ij}$ from $\mathcal{DL}_i$ to $\mathcal{DL}_j$, these general rules provide the basis for the definition of an operator $\mathfrak{B}_{ij}(\cdot)$ which takes as input $\mathcal{T}_i$ and produces a T-box $\mathcal{T}_j$, augmented with the conclusions of the rules. The theorem below states that the effects produced by the general rules characterise all, and only, the knowledge transferred to the target ontology via the current set of bridge rules.

Theorem 1 (Soundness and Completeness of $\mathfrak{B}_{ij}(\cdot)$). *Let $\mathfrak{T}_{ij} = \langle \mathcal{T}_i, \mathcal{T}_j, \mathfrak{B}_{ij} \rangle$ be a distributed T-box, where $\mathcal{T}_i$ and $\mathcal{T}_j$ are expressed in the $\mathcal{ALCQI}_b$ description logic. Then $\mathfrak{T}_{ij} \models j : X \sqsubseteq Y \Longleftrightarrow \mathcal{T}_j \cup \mathfrak{B}_{ij}(\mathcal{T}_i) \models X \sqsubseteq Y$.*

The proof can be found in [4]. The generalisation for an arbitrary network of ontologies can be obtained following the technique used in [6].

6 Discussion

In this paper we have extended the framework presented in [3] in order to properly capture the reification modelling paradigm. In [3] it was possible to assert bridge rules involving heterogeneous ontology terms, but is was not possible to link attributes to domain or range of roles. This limitation seriously hamper the possibility of mapping relations with their reified counterpart.

The need for mappings enabling the reconciliation modelling differences has been underlined in several works (e.g. [7, 6, 2]). However, to the best of our knowledge, our is the first framework to capture heterogeneous mappings. The closest approach to ours is in [2], where it is presented a framework that allows any kind of arbitrary heterogeneous mapping, which are evaluated in a common "reference" interpretation. Appropriate functions take care of relating local domains with the reference one (*equalising functions*). However, these functions maps just elements of local domains and not e.g. pairs. In this way can be seen as our "plain" bridge relations; so, hardly providing the necessary semantic expressiveness. A more detailed discussion on related works is contained in [3].

The work presented here leaves a few open problems which need to be addressed in order to provide an effective semantic framework to reconciliate schematic differences between ontologies. First of all we need to devise an intuitive syntax restriction which guarantees decidability. Roughly speaking, once a role in i has been "reified" into a concept in j, then it cannot be also mapped in a role in j. The rationale behind this is the fact that having in the same ontology both a reified role and the plain role itself doesn't seem a good modelling practice. Obviously, adding a restriction interacts with the role hierarchy, which depends on the (heterogeneous) mapping themselves; therefore the proper syntactic restriction is not obvious. Note that once the decidability has been proven, the propagation rules provide a naive fixpoint-based algorithm for checking satisfiability. However, investigating a direct distributed tableaux-based algorithm in the style of [6] would provide additional insight into the framework and room for direct optimisation. Finally, up until now only the terminological reasoning has been investigated. We are currently looking into the problem of establishing mapping among individuals and considering the problem of instance checking (i.e. query answering w.r.t. unary acyclic conjunctive queries).

References

1. C. Batini, M. Lenzerini, and S. B. Navathe. A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv.*, 18(4):323–364, 1986.
2. J. Euzenat, F. Scharffe, and A. Zimmermann. Expressive alignment language and implementation. deliverable D2.2.10, Knowledge Web NoE, 2007.
3. C. Ghidini, L. Serafini, and S. Tessaris. On relating heterogeneous elements from different ontologies. In *Proc. of the 6-th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT'07)*, volume 4635 of *LNAI*, pages 234–247. Springer, 2007.
4. C. Ghidini, L. Serafini, and S. Tessaris. Bridging heterogeneous representations of binary relations. Technical report, FBK-irst, February 2008. http://sra.itc.it/people/ghidini/publications/reification.pdf.
5. Natasha Noy, Alan Rector, Pat Hayes, and Chris Welty. Defining n-ary relations on the semantic web. Technical report, W3C, 2006.
6. L. Serafini, A. Borgida, and A. Tamilin. Aspects of distributed and modular ontology reasoning. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 570–575, August 2005.
7. H. Stuckenschmidt and M. Uschold. Representation of semantic mappings. In *Semantic Interoperability and Integration*, number 04391 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, 2005.
8. S. Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation.* PhD thesis, RWTH Aachen, Germany, 2001.
9. M Wessel. Obstacles on the way to qualitative spatial reasoning with description logics: Some undecidability results. In *Proceedings of the International Workshop on Description Logics (DL-2001)*, 2001.