PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 06/10

Revising the Constraints of the Mediated Schema

Marco Antonio Casanova, Tanara Lauschner,
Luiz André Portes Paes Leme,
Karin Koogan Breitman, Antonio Luz Furtado,
Vânia Maria Ponte Vidal

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO

Monografias em Ciência da Computação, No. 06/10 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena May, 2010

Revising the Constraints of the Mediated Schema*

Marco Antonio Casanova, Tanata Lauschner,
Luiz André Portes Paes Leme[1],
Karin Koogan Breitman, Antonio Luz Furtado,
Vânia Maria Ponte Vidal[2]

[1]Instituto de Computação, Universidade Federal Fluminense
[2]Departamento de Computação, Universidade Federal do Ceará

casanova@inf.puc-rio.br, tlauschner@inf.puc-rio.br,
lapaesleme@ic.uff.br,
karin@inf.puc-rio.br, furtado@inf.puc-rio.br,
vvidal@lia.ufc.br

Abstract. This paper addresses the problem of changing the constraints of a mediated schema to accommodate the set of constraints of a new export schema. The relevance of this problem lies in that the constraints of a mediated schema capture the common semantics of the data sources and, as such, they must be maintained and made available to the users of the mediation environment. The paper first argues that such problem can be solved by computing the greatest lower bound of two sets of constraints. Then, for an expressive family of conceptual schemas, it shows how to efficiently decide logical implication and how to compute the greatest lower bound of two sets of constraints.

Keywords: constraint revision, mediated schema, Description Logics.

Resumo. Este trabalho endereça o problema de modificar as restrições de um esquema mediado para acomodar as restrições de um novo esquema exportado. A relevância deste problema reside no fato de que as restrições do esquema mediado capturam a semântica compartilhada pelas fontes de dados e, portanto, devem ser atualizadas e tornadas disponíveis para os usuários do ambiente de mediação. O trabalho inicialmente argumenta que tal problema pode ser resolvido computando-se o maior conjunto de restrições que é simultaneamente conseqüência lógica de dois conjuntos de restrições. Em seguida, para uma família de esquemas conceituais, o trabalho mostra como decidir implicação lógica eficientemente e como computar o maior conjunto de restrições que é simultaneamente conseqüência lógica de dois conjuntos de restrições.

Palavras-chave: revisão de restrições, esquema mediado, Lógica de Descrição.

* This work was partly supported by CNPq under grants 301497/2006-0, 473110/2008-3, 557128/2009-9, FAPERJ E-26/170028/2008, and CAPES/PROCAD NF 21/2009.

In charge of publications:

Rosane Teles Lins Castilho

Assessoria de Biblioteca, Documentação e Informação

PUC-Rio Departamento de Informática

Rua Marquês de São Vicente, 225 - Gávea

22451-900 Rio de Janeiro RJ Brasil

Tel. +55 21 3527-1516 Fax: +55 21 3527-1530

E-mail: bib-di@inf.puc-rio.br

Web site: http://bib-di.inf.puc-rio.br/techreports/

1 Introduction

A *mediation environment* contains a *mediated schema* M and several *export schemas* $E_1,...,E_n$ that describe data sources. For each export schema E_i, the environment also features an *import schema* I_i and a *local mapping* γ_i that defines the concepts of I_i in terms of the concepts of E_i. The environment also has a *mediated mapping* γ that defines the concepts of M in terms of those of $I_1,...,I_n$. Figure 1 depicts these concepts.

The constraints of the mediated schema are relevant for a correct understanding of what the semantics of the external schemas have in common. For example, consider a virtual store mediating access to online booksellers. The class hierarchy of the mediated schema indicates what the booksellers' book classifications have in common; if the mediated schema enforces that all books must have ISBNs, then it means that all booksellers must abide by the same requirement; if it allows books with no (known) authors, then at least one bookseller must so allow; and so on.

We may break into three steps the process of adding to the mediation environment a new export schema E_0, with import schema I_0 and local mapping γ_0. The *concept revision step* adjusts the vocabulary of M to perhaps include classes and properties originally defined in I_0. The *mapping revision step* may modify the mediated mapping. Finally, the *constraint revision step* applies a minimum set of changes to the set of constraints of M to account for the set of constraints of I_0.

One may have to iterate through these three steps since, in particular, revising the constraints of the mediated schema may interfere with the definition of the schema mappings. For example, the local mapping γ_0 may have to be adjusted to preserve the class hierarchy of the mediated schema, or the class hierarchy of the mediated schema may have to be changed to reflect the class hierarchy of I_0.

In this paper, we are primarily concerned with the constraint revision step, with a bias to mediation environments in the context of the Web. Maintaining mediation environments in such context becomes a challenge because the number of data sources may be very large and, moreover, the mediator does not have much control over the data sources, which may join or leave the mediation environment at will.

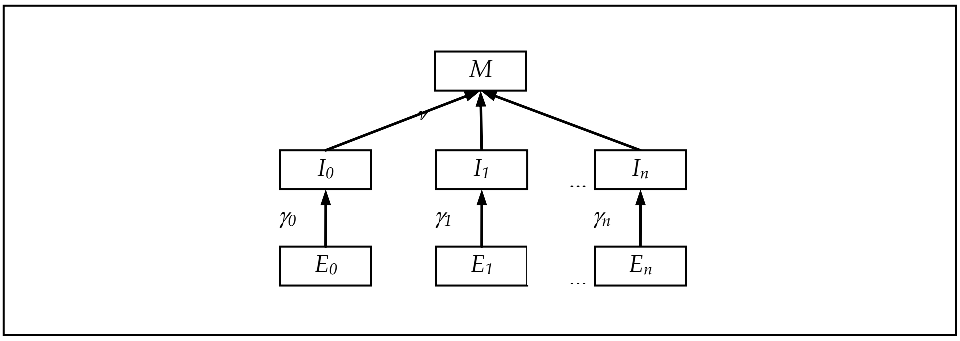

Fig. 1. Components of a mediation environment.

We break the constraint revision step in two sub-steps. The *constraint translation step* translates the set EC_0 of constraints of E_0 to I_0, creating a set of constraints IC_0 in such a way that γ_0 maps states of E_0 that satisfy EC_0 into states of I_0 that satisfy IC_0. Intuitively, as a result of this step, we express the semantics of E_0 in terms of I_0.

The *least constraint change step* applies a minimum set of changes to the constraints of M to accommodate IC_0 in such a way that all schema mappings remain correct. This step intuitively means to harmonize the semantics of E_0 with the semantics of all export schemas previously added to the mediation environment, captured in the constraints of M. The key questions here are how to precisely define what it means to apply a minimum set of changes to a set of constraints, and how to guarantee that the mappings remain correct.

The contributions of this paper are twofold. First, we formulate the problem of changing the constraints of the mediated schema as the problem of computing the greatest lower bound of two sets of constraints, defined as the intersection of their theories. Second, for an expressive family of conceptual schemas, we show how to efficiently decide logical implication and how to compute the greatest lower bound of two sets of constraints.

In more detail, we work with schemas that partly correspond to OWL Lite [7] and support the equivalent of named classes, datatype and object properties, minCardinalities and maxCardinalities, InverseFunctionalProperties, subset constraints, and disjointness constraints. The schemas we work with are also sufficiently expressive to encode commonly used UML constructs, such as classes, attributes, binary associations without association classes, multiplicity of attributes and binary associations, ISA hierarchies and disjointness.

The decision procedure described in Section 4.2, and detailed in the appendix, is based on the satisfiability algorithm for Boolean formulas in conjunctive normal form with at most two literals per clause, described in [2]. The intuition is that the constraints we consider can be treated much in the same way as Boolean implications. However, cardinality constraints pose considerable technical problems to the proof of the theorems. The decision procedure essentially explores the structure of a set of constraints, captured as a graph. The procedure to compute the greatest lower bound of two sets of constraints is a direct consequence of the decision procedure. These results are new, and cover an expressive and useful family of constraints, defined in Section 3.2.

This paper is organized as follows. Section 2 surveys related work. Section 3 reviews concepts of Description Logics and introduces the notion of mediation environment. Section 4 shows how to generate the revised set of constraints of the mediated schema. Section 5 contains the conclusions. Finally, the appendix presents the proofs for the main results.

2 Related Work

Research on the construction of mediated schemas concentrates on vocabulary matching techniques, on the definition of schema mappings, and on query processing, mostly ignoring the question of constraint revision.

Matching techniques are useful for the process of revising the vocabulary of the mediated schema, a topic we do not directly address, but mention on Section 4.1. Euzenat and Shvaiko [17] present a comprehensive survey of ontology matching. Rahm and Bernstein [39] survey schema matching, and Bernstein and Melnik [5] list the requirements for model management systems that support the matching process. Köpcke and Rahma [25] comparatively analyze eleven frameworks for entity matching.

Schema matching techniques may be classified as syntactic, semantic, or hybrid [14]. For example, Melnik et al. [35] and Madhavan et al. [34] describe syntactic techniques based on modeling the schemas as graphs. Bilke and Naumann [8] propose a semantic technique based on an analysis of duplicated instances. Brauner et al. [9] adopt this strategy to align thesauri. Wang et al. [42] describe a semantic technique based on probing the databases.

Departing from this classification, Qi and Linga [38] present algorithms to resolve schematic discrepancies by transforming metadata into the attribute values of entity types, keeping the information and constraints of original schemas. Zhaoa and Ramb [43] propose an iterative procedure for detecting both schema-level and instance-level matchings from heterogeneous data sources.

Schema and ontology reuse, as proposed in Lonsdale et. al. [32] and in Simperla [41], is a fruitful strategy to overcome interoperability issues. The use of templates to help exchange schemas, as proposed in Papott and Torlone [37], is s similar strategy that may also be used to circumvent interoperability problems.

As for the mappings between the external schema and the mediated schema, two basic approaches have been used [29]. The first approach, called *global-as-view (GAV)*, requires that the mediated schema be expressed in terms of the data sources. More precisely, a view over the data sources is associated with each element of the global schema, so that the meaning of the element is specified in terms of the data stored at the data sources. This means that adding a new data source may impact the previously defined mappings, which may need to be updated. Several projects, such as TSIMMIS [19], IBIS [12] and INFOMIX [30] adopt the GAV approach.

The second approach, called *local-as-view (LAV)*, requires that the mediated schema be specified independently from the data sources. The data sources are in turn defined as views over the mediated schema [22]. This means that adding a new data source only requires adding a new assertion to the mediated mapping. This approach improves maintainability and extensibility of the systems [6]. Agora [32], StyX [1] and Picsel [21] are examples of LAV systems.

Mappings may also be classified according to their accuracy into sound, complete and exact [11, 29]. Let V be a view associated with an element E of the mediated schema. In the GAV approach, V is *sound* when all data provided by V satisfies E, but there may be additional data satisfying E that V does not provide. View V is *complete* when not all data provided by V satisfies E, but all data satisfying E is provided by V. Finally, V is *exact*, when all data provided by V satisfies E, and all data satisfying E is provided by V [11].

Rull et al. [40] present an approach for validating schema mappings that allows the mapping designer to ask whether they have certain desirable properties.

The approach we take in Section 3.2 to define the mediation environment is akin to the idea of sound views. Yet, we consider that constraints should be included in the mediated schema to capture the common semantics of the data sources, unlike most proposals based on the concept of exact views, which assume that the mediated schema has no constraints, as observed in [29].

Calì et al. [11] argue that the constraints of a mediated schema should be taken into account during query processing and that the schema definition language should incorporate flexible and powerful representation mechanisms for integrity constraints. The authors also argue that, when the mediated schema contains constraints, the semantics of the data integration system is best described in terms of a set of databases, and that query processing should be based on the notion of querying incomplete databases.

Calvanese et al. [13] introduce a Description Logics framework, similar to that in Section 3.1, to address schema integration and query answering. Atzeni et al. [3] cover the problem of rewriting a schema from one model to another, but they do not touch on the more complex problem of generating a new set of constraints that generalizes a pair of sets of constraints from different schemas, which we address in Section 4. Hick and Hainauta [24] show how requirements changes are propagated to database schemas, to data and to programs through a general strategy.

Hartmanna et al. [23] apply techniques from Propositional Logic to offer decision support for specifying Boolean and multivalued dependencies.

Turning to a different aspect, the subsumption problem in Description Logics (DL) refers to the question of deciding if a concept description always denotes a subset of the set denoted by another concept description. The subsumption problem is decidable for expressive dialects of DL, but typically belongs to hard complexity classes [4], especially in the presence of axioms (or constraints) [16]. For certain dialects of DL, there are polynomial decision procedures for the subsumption problem that explore the structure of the concept descriptions and that are, for this reason, called *structural subsumption procedures* [18, 31]. However, such procedures do not take axioms into account. Furthermore, the reductions suggested to encode the axioms lead us back to dialects for which the subsumption problem is hard [16].

From the point of view of deciding logical implication and computing the greatest lower bound of two sets of constraints, we depart from the tradition of Description Logics deduction services, which are mostly based on tableaux techniques [4]. As mentioned in the introduction, the decision procedure described in Section 4.2, and detailed in the appendix, is based on the satisfiability algorithm for Boolean formulas in conjunctive normal form with at most two literals per clause, described in [2]. The procedure to compute the greatest lower bound of two sets of constraints is a direct consequence of the decision procedure.

3 Mediation Environment

3.1 A Brief Review of Concepts from Description Logics

We adopt a family of *attributive languages* [36] defined as follows. A *language* $\mathcal{L}$ in the family is characterized by an *alphabet* $\mathcal{A}$, consisting of a set of *atomic concepts*, a set of *atomic roles*, the *universal concept* and the *bottom concept*, denoted by $\top$ and $\bot$, respectively, the *universal role* and the *bottom role*, also denoted by $\top$ and $\bot$, respectively, and a set of *constants*.

The set of *role descriptions* of $\mathcal{L}$ is inductively defined as

- An atomic role, and the universal and bottom roles are role descriptions
- If p and q are role descriptions, then the following expressions are role descriptions

p^-	(the *inverse* of p)
$p \circ q$	(the *composition* of p and q)
$p \sqcup q$	(the *union* of p and q)

The set of *concept descriptions* of $\mathcal{L}$ is inductively defined as

- An atomic concept, and the universal and bottom concepts are concept descriptions
- If $a_1,...,a_n$ are constants, then $\{a_1,...,a_n\}$ is a concept description
- If e and f are concept descriptions and p is a role description, then the following expressions are concept descriptions

$\neg e$	(negation)
$e \sqcap f$	(intersection)
$e \sqcup f$	(union)
$\exists p$	(existential quantification)
$\exists p.e$	(full existential quantification)
$\forall p.e$	(value restriction)
$(\leq n\ p)$	(at-most restriction)
$(\geq n\ p)$	(at-least restriction)

An *interpretation* $\boldsymbol{s}$ for $\mathcal{A}$ consists of a nonempty set Δ^s, the *domain* of $\boldsymbol{s}$, whose elements are called *individuals*, and an *interpretation function*, also denoted $\boldsymbol{s}$, where:

- $\boldsymbol{s}(\bot) = \emptyset$, when $\bot$ denotes the bottom concept or the bottom role
- $\boldsymbol{s}(\top) = \Delta^s$, when $\top$ denotes the universal concept
- $\boldsymbol{s}(\top) = \Delta^s \times \Delta^s$, when $\top$ denotes the universal role

- $\boldsymbol{s}(A) \subseteq \Delta^s$, for each atomic concept $\mathcal{A}$ of $\mathcal{L}$
- $\boldsymbol{s}(P) \subseteq \Delta^s \times \Delta^s$, for each atomic role P of $\mathcal{L}$
- $\boldsymbol{s}(a) \in \Delta^s$, for each constant a of $\mathcal{L}$, such that distinct constants denote distinct individuals (the *uniqueness assumption*)

The function $\boldsymbol{s}$ is extended to role and concept descriptions of $\mathcal{L}$ as follows:

- $\boldsymbol{s}(p^-) = \boldsymbol{s}(p)^-$ (the inverse of $\boldsymbol{s}(p)$)
- $\boldsymbol{s}(p \circ q) = \boldsymbol{s}(p) \circ \boldsymbol{s}(q)$ (the composition of $\boldsymbol{s}(p)$ with $\boldsymbol{s}(q)$)
- $\boldsymbol{s}(p \sqcup q) = \boldsymbol{s}(p) \cup \boldsymbol{s}(q)$ (the union of $\boldsymbol{s}(p)$ with $\boldsymbol{s}(q)$)
- $\boldsymbol{s}(\{a_1,...,a_n\}) = \{\boldsymbol{s}(a_1),..., \boldsymbol{s}(a_n)\}$ (the set consisting of the individuals $\boldsymbol{s}(a_1),..., \boldsymbol{s}(a_n)$)
- $\boldsymbol{s}(\neg e) = \Delta^s - \boldsymbol{s}(e)$ (the complement of $\boldsymbol{s}(e)$ w.r.t. Δ^s)
- $\boldsymbol{s}(e \sqcap f) = \boldsymbol{s}(e) \cap \boldsymbol{s}(f)$ (the intersection of $\boldsymbol{s}(e)$ and $\boldsymbol{s}(f)$)
- $s(e \sqcup f) = s(e) \cup s(f)$ (the union of $s(e)$ and $s(f)$)
- $\boldsymbol{s}(\exists p) = \{I \in \Delta^s \ / \ (\exists J \in \Delta^s)((I,J) \in \boldsymbol{s}(p)\}$

  (the set of individuals that $\boldsymbol{s}(p)$ relates to some individual)
- $\boldsymbol{s}(\exists p.e) = \{I \in \Delta^s \ / \ (\exists J \in \Delta^s)((I,J) \in \boldsymbol{s}(p) \wedge J \in \boldsymbol{s}(e)\}$

  (the set of individuals that $\boldsymbol{s}(p)$ relates to some individual in $\boldsymbol{s}(e)$)
- $\boldsymbol{s}(\forall p.e) = \{I \in \Delta^s \ / \ (\forall J \in \Delta^s)((I,J) \in \boldsymbol{s}(p) \Rightarrow J \in \boldsymbol{s}(e)\}$

  (the set of individuals I such that, if $\boldsymbol{s}(p)$ relates I to an individual J, then J is in $\boldsymbol{s}(e)$)
- $\boldsymbol{s}(\geq n\ p)=\{I \in \Delta^s \ / \ |\{J \in \Delta^s \ / \ (I,J) \in \boldsymbol{s}(p)\}| \geq n\}$

  (the set of individuals that $\boldsymbol{s}(p)$ relates to at least n distinct individuals)
- $\boldsymbol{s}(\leq n\ p)=\{I \in \Delta^s \ / \ |\{J \in \Delta^s \ / \ (I,J) \in \boldsymbol{s}(p)\}| \leq n\}$

  (the set of individuals that $\boldsymbol{s}(p)$ relates to at most n distinct individuals)

A *formula* of $\mathcal{L}$ is an expression of the form $u \sqsubseteq v$, called an *inclusion*, or of the form $u \mid v$, called a *disjunction*, or of the form $u \equiv v$, called an *equivalence*, where u and v are both concept descriptions or they are both role descriptions of $\mathcal{L}$. A *definition* is an equivalence of the form $T \equiv u$, where T is an atomic concept and u is a concept description, or T is an atomic role and u is a role description.

An interpretation $\boldsymbol{s}$ for $\mathcal{L}$ *satisfies* $u \sqsubseteq v$ iff $\boldsymbol{s}(u) \subseteq \boldsymbol{s}(v)$, $\boldsymbol{s}$ *satisfies* $u \mid v$ iff $\boldsymbol{s}(u) \cap \boldsymbol{s}(v)=\varnothing$, and $\boldsymbol{s}$ *satisfies* $u \equiv v$ iff $\boldsymbol{s}(u) = \boldsymbol{s}(v)$. We adopt the following familiar notation, where σ is a formula and Σ and Γ are sets of formulas:

- $\boldsymbol{s} \models \sigma$ indicates that $\boldsymbol{s}$ satisfies σ

- $s \models \Sigma$ indicates that s satisfies all formulas in Σ; in this case, we say that s is a *model* of Σ
- Σ is *satisfiable* iff there is a model of Σ
- $\Sigma \models \sigma$ indicates that any model of Σ satisfies σ; in this case, we say that Σ *logically implies* σ
- $\Sigma \models \Gamma$ indicates that any model of Σ is also a model of Γ; in this case, we say that Σ *logically implies* Γ
- *Th(Σ)* denotes the *theory induced* by Σ, which is the smallest set of formulas that contains Σ and is closed under logical implication

Also, in Sections 3 and 4, we will use concept and role descriptions over an alphabet $\mathcal{A}$ which is the union of disjoint alphabets $\mathcal{A}_1,...,\mathcal{A}_n$. The syntax of concept and role descriptions remains the same. An interpretation s for $\mathcal{A}$ is constructed from interpretations $s_1,...,s_n$ for $\mathcal{A}_1,...,\mathcal{A}_n$ in the obvious way, except that we assume that

- *(Domain Disjointness Assumption)* Any pair of interpretations for $\mathcal{A}_i$ and $\mathcal{A}_j$ have disjoint domains, for each $i,j \in [1,n]$, with $i \neq j$

3.2 Extralite Schemas

We will work with *extralite schemas* that partly correspond to OWL Lite [7]. Extralite schemas support the equivalent of named classes, datatype and object properties, minCardinalities and maxCardinalities, InverseFunctionalProperties, which capture simple keys, subset constraints, and disjointness constraints. Extralite schemas are also sufficiently expressive to encode commonly used UML constructs, such as classes, attributes, binary associations without association classes, multiplicity of attributes and binary associations, ISA hierarchies and disjointness, but not complete generalizations.

Formally, an *extralite schema* is a pair $S=(\mathcal{A},C)$ such that

- $\mathcal{A}$ is an alphabet, called the *vocabulary* of S, whose atomic concepts and atomic roles are called the *classes* and *properties* of S, respectively
- C is a set of formulas, called the *constraints* of S, which must be of one the forms
- *Domain Constraint:* $\exists P \sqsubseteq D$ (property P has class D as domain)
- *Range Constraint:* $\exists P^- \sqsubseteq R$ (property P has class R as range)
- *minCardinality constraint:* $C \sqsubseteq (\geq k\ P)$ or $C \sqsubseteq (\geq k\ P^-)$
  (property P or its inverse P^- maps each individual in class C to at least k distinct individuals)
- *maxCardinality constraint:* $C \sqsubseteq (\leq k\ P)$ or $C \sqsubseteq (\leq k\ P^-)$
  (property P or its inverse P^- maps each individual in class C to at most k distinct individuals)

- *Subset Constraint:* $E \sqsubseteq F$ (class E is a subclass of class F)
- *Disjointness Constraint:* $E \mid F$ (classes E and F are disjoint)

We also admit constraints of one of the forms:

- $C \sqsubseteq \bot$ (class C is always empty)
- $\exists P \sqsubseteq \bot$ or $\exists P^{-} \sqsubseteq \bot$ (property P is always empty, i.e., P has an empty domain or an empty range)

We will use the terms *class*, *property*, *vocabulary* and *state* interchangeably with *atomic concept*, *atomic role*, *alphabet* and *interpretation*, respectively.

Example 1: Figure 2 contains schemas for fragments of the Amazon and the eBay databases, using the namespace prefixes "a:" and "e:" to refer to their vocabularies, respectively. Figures 2(a) and 2(c) show the schemas using an informal notation. Figures 2(b) and 2(d) formalize the constraints: the first column shows the domain and range constraints; the second column depicts the cardinality constraint; and the third column contains the subset and disjointness constraints.

For example, the first column of Figure 2(b) indicates that:

- `a:title` is a property with domain `a:Product` and range `string` (the set of XML Schema strings)
- `a:pub` is a property with domain `a:Book` and range `a:Publ`

a:Product
  a:title range string
  a:price range decimal
  a:currency range string
a:Book
  a:isbn range string
  a:author range string
  a:pub range a:Publ
a:Publ
  a:name range string
  a:city range string
	a:Book is-a a:Product
	a:Music is-a a:Product
	
	a:Book disjoint-from a:Music

Fig. 2(a). Informal definition of the Amazon schema.

$\exists$ a:title $\sqsubseteq$ a:Product $\exists$ a:title^{-} $\sqsubseteq$ string ... $\exists$ a:pub $\sqsubseteq$ a:Book $\exists$ a:pub^{-} $\sqsubseteq$ a:Publ ... $\exists$ a:name $\sqsubseteq$ a:Publ $\exists$ a:name^{-} $\sqsubseteq$ string ...	a:Product $\sqsubseteq$ ($\leq$ 1 a:title) a:Product $\sqsubseteq$ ($\leq$ 1 a:price) a:Product $\sqsubseteq$ ($\leq$ 1 a:currency) a:Book $\sqsubseteq$ ($\leq$ 1 a:isbn) a:Book $\sqsubseteq$ ($\geq$ 2 a:pub) a:Publ $\sqsubseteq$ ($\leq$ 1 a:name) a:Publ $\sqsubseteq$ ($\geq$ 3 a:city)	a:Book $\sqsubseteq$ a:Product a:Music $\sqsubseteq$ a:Product a:Book | a:Music

Fig. 2(b). Formal definition of (some of) the constraints of the Amazon schema.

e:Seller
  e:name range string
e:Offer
  e:qty range integer
  e:price range double
  e:currency range string
  e:seller range e:Seller
  e:product range e:Product
	e:Product
	  e:type range string
	  e:ean range integer
	  e:title range string
	  e:author range string
	  e:edition range integer
	  e:year range integer
	  e:place range string

Fig. 2(c). Informal definition of the eBay schema.

$\exists$ e:name $\sqsubseteq$ e:Seller $\exists$ e:name$^-$ $\sqsubseteq$ string ... $\exists$ e:seller $\sqsubseteq$ e:Offer $\exists$ e:seller$^-$ $\sqsubseteq$ e:Seller $\exists$ e:product $\sqsubseteq$ e:Offer $\exists$ e:product$^-$ $\sqsubseteq$ e:Product ...	e:Seller $\sqsubseteq$ ($\leq$ 1 e:name) e:Offer $\sqsubseteq$ ($\leq$ 1 e:qty) e:Offer $\sqsubseteq$ ($\leq$ 1 e:price) ... e:Product $\sqsubseteq$ ($\leq$ 1 e:type) e:Product $\sqsubseteq$ ($\leq$ 1 e:ean) e:Product $\sqsubseteq$ ($\leq$ 1 e:title) ... e:Product $\sqsubseteq$ ($\geq$ 1 e:place)	(no subset or disjointness constraints)

Fig. 2(d). Formal definition of (some of) the constraints of the eBay schema.

The second column of Figure 2(b) shows the cardinalities of the Amazon schema:

- all properties have maxCardinality equal to 1, except `a:author`, `a:pub` and `a:city`
- `a:author` has unbounded maxCardinality, consistently with the fact that a book may have multiple authors
- `a:pub` has minCardinality equal to 2
- `a:city` has minCardinality equal to 3

The third column of Figure 2(b) indicates that `a:Book` and `a:Music` are subclasses of `a:Product`, and that `a:Book` and `a:Music` are disjoint classes.

Figure 2(d) likewise describes the constraints of the eBay schema. In particular, the second column indicates that all properties have maxCardinality equal to 1, except `e:place`. □

3.3 Components of a Mediation Environment

A *mediation environment* contains a *mediated schema M*, a *mediated mapping* γ and, for each $k=1,\ldots,n$, an *export schema* E_k, an *import schema* I_k and a *local mapping* γ_k.

Import schemas are a notational convenience to divide the definition of the mappings into two stages: the definition of the local mappings and the definition of the mediated mapping. We restrict the import schemas as follows:

(1) for $k=1,\ldots,n$, the vocabulary of I_k is equal to the vocabulary of M, in the sense that the two vocabularies have the same classes and properties, but different namespaces

Assume that the classes and properties in M are $C_1,\ldots,C_u$ and $P_1,\ldots,P_v$. We adopt namespace prefixes, as in the examples, to distinguish the occurrence of a symbol in the vocabulary of M from the occurrence of the same symbol in the vocabulary of I_k. However, in the formal development, we follow a more abstract notation. For each class C_i (or property P_j) in the vocabulary of M, we denote the occurrence of C_i (or P_j) in the vocabulary of I_k by C_i^k (or P_j^k), and say that C_i^k (or P_j^k) *matches* C_i (or P_j).

For each $k=1,...,n$, the *local mapping* γ_k defines the classes and properties of I_k in the terms of the vocabulary of the export schema E_k. We restrict γ_k as follows:

- for each class C_i^k of I_k, the local mapping γ_k contains a definition of the form

(2) $C_i^k \equiv \rho_i^k$, where ρ_i^k is a concept description over the vocabulary of E_k

- for each property P_j^k of I_k, the local mapping γ_k contains a definition of the form

(3) $P_j^k \equiv \pi_j^k$, where π_j^k is a role description over the vocabulary of E_k

Note that ρ_i^k may be the bottom concept $\bot$ to indicate that E_k does not contribute with any individual to class C_i^k. In other words, the interpretation of C_i^k is always an empty set. Combined with the requirement that the vocabulary of I_k be equal to the vocabulary of M, this might seem an unnecessary complication. However, these technical details simplify the computation of the revised set of constraints of a mediated schema. Likewise, π_j^k may be the bottom role $\bot$, when E_k does not contribute with any individual to property P_j^k.

We introduce $\bar{\gamma}_k$ as the *function induced by* γ_k, defined as the function from states of E_k into states of I_k such that, for each state $\boldsymbol{s}$ of E_k, $\bar{\gamma}_k(\boldsymbol{s}) = \boldsymbol{r}$ iff

- $\boldsymbol{r}(C_i^k) = \boldsymbol{s}(\rho_i^k)$, if $C_i^k \equiv \rho_i^k$ is the definition for class C_i^k in γ_k
- $\boldsymbol{r}(P_j^k) = \boldsymbol{s}(\pi_j^k)$, if $P_j^k \equiv \pi_j^k$ is the definition for property P_j^k in γ_k

For each $k=1,...,n$, let EC_k be the set of constraints of E_k. The set IC_k of constraints of the import schema I_k should be defined so that $\bar{\gamma}_k$ maps consistent states of E_k into consistent states of I_k. We refer the reader to Lauschner et al. [26] for efficient strategies to generate IC_k, when EC_k is the family of schema constraints considered in Section 3.2 and the local mapping γ_k uses an expressive family of concept and role expressions.

We illustrate the concepts just introduced with the help of an example.

Example 2: Consider the Sales mediated schema with the vocabulary shown in Figure 4(a), distinguished by the namespace prefix "`s:`".

Figure 3(a) defines the vocabulary of the Amazon import schema, which is equal to that of the Sales mediated schema, but is identified by the namespace prefix "`ai:`". Figure 3(b) contains the translation of the constraints of the Amazon export schema, shown in Figure 2(b), to the Amazon import schema. Figure 3(c) contains the local mapping that

Classes:	Properties:
ai:Product ai:Book ai:Music	ai:title ai:city

Fig. 3(a). Vocabulary of the Amazon import schema.

∃ ai:title ⊑ ai:Product ∃ ai:title⁻ ⊑ string ∃ ai:city ⊑ ai:Book ∃ ai:city⁻ ⊑ string	ai:Product ⊑ (≤ 1 ai:title) ai:Book ⊑ (≥ 3 ai:city)	ai:Book ⊑ ai:Product ai:Music ⊑ ai:Product ai:Book | ai:Music

Fig. 3(b). Constraints of the Amazon import schema.

ai:Product ≡ a:Product ai:Music ≡ a:Music ai:Book ≡ a:Book	ai:title ≡ a:title ai:city ≡ a:pub∘a:city

Fig. 3(c). Local mapping from the Amazon export schema to Amazon import schema.

defines the concepts of the vocabulary of the Amazon import schema in terms of the concepts of the vocabulary of the Amazon export schema of Figure 2(a).

For example, the definitions `ai:city` ≡ `a:pub` ∘ `a:city` and `ai:Book` ≡ `a:Book` have several consequences. First, the domain and range of `ai:city` are `ai:Book` and `string`. Second, `ai:city` has minCardinality 3 with respect to `ai:Book` since, observing Figure 2(b), `a:pub` has minCardinality 2 with respect to `a:Book`, `a:city` has minCardinality 3 with respect to `a:Publ`, and `a:Publ` is both the range of `a:pub` and the domain of `a:city`.

Intuitively, in the Amazon schema, each book is associated with at least 2 publishers and each publisher is located in at least 3 cities, which are not necessarily distinct from the cities associated with other publishers. Hence, in the Amazon import schema, all we can assert is that each book is associated with at least 3 publishers' cities. As a concrete example, suppose that: (1) the book "Semantic Web" is associated with two publishers, "Springer Verlag" and "Ed. Campus"; (2) "Springer Verlag" is located in three cities "London", "Berlin" and "Sidney"; "Ed. Campus" is also located in "London", "Berlin" and "New York". Note that these individuals do not violate the cardinality constraints of the Amazon export schema. Then, the book "Semantic Web" is associated with three cities, "London", "Berlin" and "Sidney".

The other constraints of the Amazon import schema follow directly from those of the Amazon export schema, since each of the other classes and properties of the import schema is defined in terms of a single class or property of the export schema.

Figure 3(d) defines the vocabulary of the eBay import schema, which is again equal to that of the Sales mediated schema, but is identified by the namespace prefix "`ei:`". Figure 3(e) contains the translation of the constraints of the eBay export schema, shown in Figure 2(c), to the eBay import schema. Figure 3(f) contains the local mapping for the eBay export schema of Figure 2(c). In particular, observe that, in Figure 3(f), `ei:Music` and `ei:Book` are defined as restrictions of `e:Product` (given an atomic concept *A*, a *restriction of A* is an intersection of the form *A* ⊓ *e*). As a consequence, we have the two subset constraints and the disjointness constraint shown on the third column of Figure 3(e), albeit the original eBay schema has no such constraints (see Figure 2(d)). Note that the disjointness constraint requires assuming that distinct constants denote distinct individuals. □

Classes:	Properties:
ei:Product ei:Book ei:Music	ei:title ei:city

Fig. 3(d). Vocabulary of the eBay import schema.

∃ ei:title ⊑ ei:Product ∃ ei:title⁻ ⊑ string ∃ ei:city ⊑ ei:Product ∃ ei:city⁻ ⊑ string	ei:Product ⊑ (≤ 1 ei:title) ei:Product ⊑ (≥ 1 ei:city)	ei:Book ⊑ ei:Product ei:Music ⊑ ei:Product ei:Book | ei:Music

Fig. 3(e). Constraints of the eBay import schema.

ei:Product ≡ e:Product ei:Music ≡ e:Product ⊓ ∃e:type.{'music'} ei:Book ≡ e:Product ⊓ ∃e:type.{'book'}	ei:title ≡ e:title ei:city ≡ e:place

Fig. 3(f). Local mapping from the eBay export schema to the eBay import schema.

We now complete the description of a mediation environment with the definition of the mediated mapping. We restrict a mediated mapping γ as follows:

- for each $i=1,\ldots,u$, the mapping γ contains a definition of the form

(4) $C_i \equiv C_i^1 \sqcup \ldots \sqcup C_i^n$

  where C_i^k is the class of I_k that matches C_i (which always exists by (1)), for each $k=1,\ldots,n$

- for each $j=1,\ldots,v$, the mapping γ contains a definition of the form

(5) $P_j \equiv P_j^1 \sqcup \ldots \sqcup P_j^n$

  where P_j^k is the property of I_k that matches P_j (which always exists by (1)), for each $k=1,\ldots,n$

We introduce $\bar{\gamma}$ as the *function induced by* the mediated mapping γ and the local mapping $\gamma_1,\ldots,\gamma_n$ as the mapping from states of $E_1,\ldots,E_n$ into states of M such that, for states $\mathbf{s}_1,\ldots,\mathbf{s}_n$ of $E_1,\ldots,E_n$, $\bar{\gamma}(\mathbf{s}_1,\ldots,\mathbf{s}_n) = \mathbf{r}$ iff, for $i=1,\ldots,u$ and $j=1,\ldots,v$

- $\mathbf{r}(C_i) = \mathbf{s}_1(C_i^1) \cup \ldots \cup \mathbf{s}_n(C_i^n)$, if $C_i \equiv C_i^1 \sqcup \ldots \sqcup C_i^n$ is the definition of C_i in γ
- $\mathbf{r}(P_j) = \mathbf{s}_1(P_j^1) \cup \ldots \cup \mathbf{s}_n(P_j^n)$, if $P_j \equiv P_j^1 \sqcup \ldots \sqcup P_j^n$ is the definition of P_j in γ

Example 3: A complete description of a mediation environment would be as follows:

- for the mediated schema Sales
  - the vocabulary listed in Figure 4(a)
  - the constraints shown in Figure 4(b), whose construction is discussed in Example 4 in Section 4.1
  - the mediated mapping shown in Figure 4(c)

- for the Amazon database fragment:
  - the export schema shown in Figures 2(a) and 2(b)
  - the import schema with the vocabulary listed in Figure 3(a) and the constraints shown in Figure 3(b)
  - the local mapping shown in Figure 3(c)
- for the eBay database fragment:
  - the export schema shown in Figures 2(c) and 2(d)
  - the import schema with the vocabulary listed in Figure 3(d) and the constraints shown in Figure 3(e)
  - the local mapping shown in Figure 3(f). □

Classes:	Properties:
s:Product s:Book s:Music	s:title s:city

Fig. 4(a). Vocabulary of the Sales mediated schema.

∃ s:title ⊑ s:Product ∃ s:title⁻ ⊑ string ∃ s:city ⊑ s:Product ∃ s:city⁻ ⊑ string	s:Product ⊑ (≤ 1 s:title) s:Book ⊑ (≥ 1 s:city)	s:Book ⊑ s:Product s:Music ⊑ s:Product s:Book | s:Music

Fig. 4(b). Constraints of the Sales mediated schema.

s:Product ≡ ai:Product ⊔ ei:Product s:Music ≡ ai:Music ⊔ ei:Music s:Book ≡ ai:Book ⊔ ei:Book	s:title ≡ ai:title ⊔ ei:title s:city ≡ ai:city ⊔ ei:city

Fig. 4(c). Mediated mapping.

4 Construction of the Mediated Schema Constraints

4.1 Basic Steps of the Constraint Revision Process

Consider a mediation environment with mediated schema *M* and mediated mapping γ. Assume that *MV* is the vocabulary and *MC* is the set of constraints of *M*. Let E_0 be a new export schema, with vocabulary EV_0 and set of constraints EC_0.

To create a revised mediation environment that includes E_0, we treat *M* much in the same way as a data source, as follows:

1. *(Concept revision step)*

   1.1. Define the vocabulary MV_r of the revised mediated schema M_r with the same classes and properties as *MV* and perhaps new classes and properties to reflect those in EV_0.

1.2. Define a new vocabulary MV^+ by adding to MV these new classes and properties.

1.3. Define the vocabulary IV_0 of the import schema I_0 for E_0 with the same classes and properties as MV_r.

2. *(Mapping revision step)*

2.1. Define the local mapping γ_0 between I_0 and E_0.

2.2. Define a new mediated mapping γ^+ by adding to γ definitions for the new classes and properties in MV^+.

2.3. Define the mediated mapping γ_r as in Equations (4) and (5).

3. *(Constraint revision step)*

3.1. Define the set IC_0 of constraints of I_0 by inspecting EC_0 and γ_0.

3.2. Define a new set of constraints MC^+ by adding to MC constraints for the new classes and properties in MV^+.

3.3. Define the set of constraints MC_r of M_r by applying a minimum set of changes to MC^+ to account for IC_0.

Step 3.3 is the main trust of this paper and is discussed in detail in this and the next sections. Steps 1.1, 1.2, 1.3 and 2.1 may be carried out by the automated matching process we discussed in [10, 27, 28]. Step 3.1 was discussed in [26]. Steps 2.2, 2.3 and 3.2 are quite simple, but raise a few points that we address in what follows.

As in Section 3.3, assume that the classes and properties in MV are C_i and P_j, for $i=1,\dots,u$ and $j=1,\dots,v$. Suppose that the classes and properties in MV_r are C_i^r and P_j^r, for $i=1,\dots,u+p$ and $j=1,\dots,v+q$. Then, for $i=u+1,\dots,u+p$ and $j=v+1,\dots,v+q$

- the new classes and properties in MV^+ are C_i and P_j, which *match* C_i^r and P_j^r
- the new definitions in γ^+ are $C_i \equiv \bot$ and $P_j \equiv \bot$
- the new constraints in MC^+ are $C_i \sqsubseteq \bot$, $\exists P_j \sqsubseteq \bot$ and $\exists(P_j)^- \sqsubseteq \bot$

Observe that the new constraints in MC^+ are a trivial consequence of the fact that, for $i=u+1,\dots,u+p$ and $j=v+1,\dots,v+q$, the new definitions in γ^+ force C_i and P_j to always have empty interpretations. In particular, the constraints for P_j capture that P_j is an empty property by saying that the domain and range of P_j are always empty. This strategy is necessary since the constraints we consider do not allow expressions of the form $P_j \sqsubseteq \bot$. Furthermore, note that it is redundant (but not wrong) to add constraints saying that both the domain and the range of P_j are always empty.

Also observe that IC_0 will likewise have a constraint of the form $C_i^0 \sqsubseteq \bot$, whenever γ_0 contains a definition of the form $C_i^0 \equiv \bot$, and constraints of the forms $\exists P_j^0 \sqsubseteq \bot$ and $\exists(P_j^0)^- \sqsubseteq \bot$, whenever γ_0 contains a definition of the form $P_j^0 \equiv \bot$.

The revised mapping can then be written as follows:

- for each $i=1,...,u+p$, the revised mediated mapping γ_r contains a definition of the form

(6) $C_i^r \equiv C_i^0 \sqcup C_i$, where C_i^0 is the class of I_0 that *matches* C_i^r and C_i is the class of M that *matches* C_i^r

- for each $j=1,...,v+q$, the revised mediated mapping γ_r contains a definition of the form

(7) $P_j^r \equiv P_j^0 \sqcup P_j$, where P_j^0 is the property of I_0 that *matches* P_j^r and P_j is the property of M that *matches* P_j^r

We focus on how to create the revised set of constraints MC_r. The reader should bear in mind the notation just introduced, which will be used in what follows.

There are two questions here: (1) what it means to apply a minimum set of changes to a set of constraints; (2) how to maintain the correctness of the schema mappings. To address the first question, we introduce a lattice of sets of constraints.

Recall from Section 3.1 that $Th(\Phi)$ denotes the theory induced by a set of formulas Φ. Let $\mathcal{T}$ be the set of all sets of constraints. Then, $(\mathcal{T}, \models)$ is a lattice where, given any two sets of constraints, Φ_1 and Φ_2, their least upper bound (l.u.b.) is $\Phi_1 \triangledown \Phi_2 = Th(\Phi_1) \cup Th(\Phi_2)$ and their greatest lower bound (g.l.b.) is $\Phi_1 \vartriangle \Phi_2 = Th(\Phi_1) \cap Th(\Phi_2)$. Note that $\Phi_i \models \Phi_1 \vartriangle \Phi_2$ and $\Phi_1 \triangledown \Phi_2 \models \Phi_i$, for $i=1,2$.

We argue that MC_r can be taken as the g.l.b. of the translation of MC^+ to MV_r and the translation of IC_0 to MV_r. Note that a translation step is necessary since, technically, no two constraints respectively in $Th(MC^+)$ and in $Th(IC_0)$ would be equal since they are written in different vocabularies. Intuitively, the translation would be just a matter of changing namespaces.

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two languages with alphabets $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively.

- An injective mapping λ:$A_1 \rightarrow L_2$ is called a *substitution function* from A_1 into L_2 iff
  - $\lambda(\bot)=\bot$ and $\lambda(\top)=\top$
  - if s is an atomic concept of A_1 and $\lambda(s)=e$ then e is a concept expression of L_2
  - if s is an atomic role of A_1 and $\lambda(s)=e$ then e is a role expression of L_2
- The *translation* of a formula β of L_1 to L_2 via λ is the formula of L_2, denoted by $\beta[\lambda]$, obtained by replacing in β each symbol A of A_1 by $\lambda(A)$.
- The *translation* of a set of formulas B of L_1 to L_2 via λ is the set of formulas of L_2, denoted $B[\lambda]$, obtained by translating each formula in B to L_2 via λ.

In particular, the mediated mapping γ_r induces three *canonical substitution functions:*

- $\hat{\gamma}_r^0$ from IV_0 into MV_r such that $\hat{\gamma}_r^0(A)=B$ iff A is an atomic concept or an atomic role of IV_0 that occurs in the body of the definition for B in γ_r
- $\hat{\gamma}_r^+$ from MV^+ into MV_r such that $\hat{\gamma}_r^+(A)=B$ iff A is an atomic concept or an atomic role of MV^+ that occurs in the body of the definition for B in γ_r
- $\hat{\gamma}_r$ from MV_r into $IV_0 \cup MV^+$ such that $\hat{\gamma}_r(B)=e$ iff the definition of B in γ_r is $B \equiv e$

To improve the notation, we write the translation of a constraint β of IC_0 from IV_0 to MV_r using $\hat{\gamma}_r^0$ as $\beta[IV_0 \rightarrow MV_r]$, the translation of a constraint β of MV^+ from MV^+ to MV_r using $\hat{\gamma}_r^+$ as $\beta[MV^+ \rightarrow MV_r]$, and the translation of a constraint β of M_r from MV_r to $IV_0 \cup MV^+$ using $\hat{\gamma}_r$ as $\beta[MV_r \rightarrow IV_0 \cup MV^+]$.

Therefore, the translation of IC_0 to MV_r is the set of constraints $IC_0[IV_0 \rightarrow MV_r]$ and the translation of MC^+ to MV_r is the set of constraints $MC^+[MV^+ \rightarrow MV_r]$.

We are now ready to state that MC_r can be taken as the g.l.b. of $IC_0[IV_0 \rightarrow MV_r]$ and $MC^+[MV^+ \rightarrow MV_r]$ without impairing consistency preservation.

Theorem 1: Let $MC_r = IC_0[IV_0 \rightarrow MV_r] \bigtriangleup MC^+[MV^+ \rightarrow MV_r]$. Suppose that:

(i) *(Domain Disjointness Assumption)* Any pair of interpretations for EV_i and EV_j have disjoint domains.

(ii) The mediated mapping γ and the local mapping $\gamma_1,...,\gamma_n$ induce a mapping from consistent states of $E_1,...,E_n$ into consistent states of M.

(iii) The local mapping γ_0 induces a mapping from consistent states of E_0 into consistent states of I_0.

Then, the revised mediated mapping γ_r and the local mappings $\gamma_0,\gamma_1,...,\gamma_n$ induce a mapping from consistent states of EC_0, $EC_1,...,$ EC_n into states of the revised mediated schema that satisfy MC_r. □

The appendix contains a proof of Theorem 1. We just anticipate here that it depends on the definition of the mediated mapping with the help of union expressions, as in Equations (6) and (7), and on the Domain Disjointness Assumption, introduced at the end of Section 3.1.

We also stress that, since MC_r is defined as the g.l.b. of $IC_0[IV_0 \rightarrow MV_r]$ and $MC^+[MV^+ \rightarrow MV_r]$ with respect to $(\mathcal{T}, \models)$, we consider that MC_r is the least way to revise MC – in the sense that MC_r is the smallest such theory – and yet retain correctness of the mappings, in view of Theorem 1.

We now give a simple example that illustrates how the constraints of a mediated schema can be defined.

Example 4: We illustrate how the constraints of the Sales mediated schema, listed in Figure 4(b), can be gradually constructed from the constraints of the Amazon and the eBay import schemas, shown in Figures 3(b) and 3(e). Then, we discuss how to include a third import schema.

(A) Assume that the Sales mediation environment contains just the definition of the vocabulary listed in Figure 4(a). Suppose that one wishes to add to the mediation environment the Amazon fragment described in Figures 2(a) and (b), with the import schema defined in Figures 3(a) and (b), and the local mapping introduced in Figure 3(c).

Then, after this initial step, the Amazon import schema is treated as the mediated schema, and the mediated mapping is simply empty. Furthermore, the initial vocabulary of the mediated schema is in fact that of the Amazon import schema, identified by the namespace prefix "`ai:`", with classes `ai:Book`, `ai:Music` and `ai:Product`, and properties `ai:title` and `ai:city`.

(B) Consider adding to the mediation environment the eBay fragment described in Figures 2(c) and (d), with the import schema defined in Figures 3(d) and (e), and the local mapping introduced in Figure 3(f).

We perform three steps:

(Concept revision step) Assume for the sake of argument that no new classes or properties are added. Thus, the Sales vocabulary, now identified by the namespace prefix "`s:`", has classes `s:Book`, `s:Music` and `s:Product`, and properties `s:title` and `s:city`.

(Mapping revision step) Figure 5(a) shows the revised mediated mapping of the Sales mediation environment.

s:Product ≡ ai:Product ⊔ bi:Product	s:title ≡ ai:title ⊔ bi:title
s:Music ≡ ai:Music ⊔ bi:Music	s:city ≡ ai:city ⊔ bi:city
s:Book ≡ ai:Book ⊔ bi:Book	

Fig. 5(a). Revised mediated mapping of the Sales mediation environment.

(Constraint revision step) Consider the following sets of constraints:

- Ψ_A, Ψ_E – the sets of constraints of the Amazon and eBay import schemas, shown in Figures 3(b) and (e).
- Φ_A, Φ_E – the sets of constraints obtained by translating, respectively, Ψ_A and Ψ_E to the vocabulary of the mediated schema. The translation is simply a process that replaces `ai:Product` by `s:Product`, etc.

∃ s:title ⊑ s:Product ∃ s:title⁻ ⊑ string ∃ s:city ⊑ s:Product ∃ s:city⁻ ⊑ string	s:Product ⊑ (≤ 1 s:title) s:Book ⊑ (≥ 1 s:city)	s:Book ⊑ s:Product s:Music ⊑ s:Product s:Book | s:Music

Fig. 5(b). Constraints of the Sales mediated schema.

We stress that it does not make sense to compute the g.l.b. of Ψ_A and Ψ_E, since these constraints are written in different vocabularies. Therefore, we compute the g.l.b. of Φ_A and Φ_E, which are constraints in the same vocabulary (that of the mediated schema). Since

$$\Phi_A \triangle \Phi_E = Th(\Phi_A) \cap Th(\Phi_E)$$

we have to find the constraints that are simultaneously derivable from Φ_A and from Φ_E. For ease of reference, Figure 5(b) repeats the constraints of the Sales mediated schema.

We first analyze in detail what minCardinality constraints for property `s:city` are in $\Phi_A \triangle \Phi_E$. From Figures 3(b) and (e), we have the following minCardinality constraints for `city` in Ψ_A and Ψ_E:

(1) `ai:Book` ⊑ (≥ 3 `ai:city`) (in Ψ_A)

(2) `ei:Product` ⊑ (≥ 1 `ei:city`) (in Ψ_E)

We also have the following subset constraint in Ψ_E:

(3) `ei:Book` ⊑ `ei:Product` (in Ψ_E)

When translated to the vocabulary of the mediated schema, identified by the prefix "`s:`", the constraints in (1) to (3) become:

(4) `s:Book` ⊑ (≥ 3 `s:city`) (in Φ_A)

(5) `s:Product` ⊑ (≥ 1 `s:city`) (in Φ_E)

(6) `s:Book` ⊑ `s:Product` (in Φ_E)

Hence, the only minCardinality constraint for property `s:city` that is simultaneously derivable from Φ_A and Φ_E is

(7) `s:Book` ⊑ (≥ 1 `s:city`) (in $\Phi_A \triangle \Phi_E$)

Indeed, we have that:

- (4) implies (7), if we observe that a minCardinality of n implies a minCardinality of m, if $m \leq n$
- (5) and (6) imply (7)

By a simpler argument, we also have:

(8) `s:Product` ⊑ (≤ 1 `s:title`) (in $\Phi_A \triangle \Phi_E$)

The subset and disjointness constraints in $\Phi_A \triangle \Phi_E$ are those shown in the third column of Figure 5(b); in fact, they are in the intersection of Φ_A and Φ_E.

The domain and range constraints in $\Phi_A \triangle \Phi_E$ are those shown in the first column of Figure 5(b); in fact, they are in the intersection of Φ_A and Φ_E, except for the domain constraint ∃`s:city` ⊑ `s:Product`, which is derived as follows.

From Figures 3(b) and (e), we have the following domain constraints in Ψ_A and Ψ_E :

(9) ∃ `ai:city` ⊑ `ai:Book` (in Ψ_A)

(10) ∃ `ei:city` ⊑ `ei:Product` (in Ψ_E)

We also have the following subset constraints in Ψ_A:

(11) `ai:Book` ⊑ `ai:Product` (in Ψ_A)

When translated to the vocabulary of the mediated schema, once again, identified by the prefix "`s:`", the constraints in (9) to (11) become:

(12) ∃ `s:city` ⊑ `s:Book` (in Φ_A)

(13) ∃ `s:city` ⊑ `s:Product` (in Φ_E)

(14) `s:Book` ⊑ `s:Product` (in Φ_A)

Hence, the domain constraint for property `s:city` that is simultaneously derivable from Φ_A and Φ_E is

(15) ∃ `s:city` ⊑ `s:Product` (in $\Phi_A \triangle \Phi_E$)

This illustrates the computation of the constraints of a mediated schema as the g.l.b. of the sets of constraints of the import schemas, after proper translation.

(C) Let BN be a new export schema (say, a fragment of the Barnes&Noble database), shown in Figures 5(c) and 5(d).

Classes:	Properties:
`b:Product` `b:CultProd` `b:Music` `b:Book`	`b:title`

Fig. 5(c) Vocabulary of the BN export schema.

∃`b:title` ⊑ `b:Product`
∃`b:title`⁻ ⊑ `string`
	(no cardinality constraints)	`b:CultProd` ⊑ `b:Product`
		`b:Book` ⊑ `b:CultProd`
		`b:Music` ⊑ `b:CultProd`

Fig. 5(d). Constraints of the BN export schema.

Classes:	Properties:
sr:Product sr:Music sr:Book	sr:title sr:city

Fig. 5(e) Vocabulary of the Sales/BN mediated schema.

Classes:	Properties:
s:Product s:Music s:Book	s:title s:city

Fig. 5(f) Vocabulary of the Sales mediated schema.

Classes:	Properties:
bi:Product bi:Music bi:Book	bi:title bi:city

Fig. 5(g) Vocabulary of BN import schema.

bi:Product ≡ b:Product bi:Music ≡ b:Music bi:Book ≡ b:Book	bi:title ≡ b:title bi:city ≡ ⊥

Fig. 5(h). Local mapping from the BN export schema to the BN import schema.

sr:Product ≡ bi:Product ⊔ s:Product sr:Music ≡ bi:Music ⊔ s:Music sr:Book ≡ bi:Book ⊔ s:Book	sr:title ≡ bi:title ⊔ s:title sr:city ≡ bi:city ⊔ s:city

Fig. 5(i). Mediated mapping of the Sales/BN mediation environment.

$\exists$bi:title ⊑ bi:Product $\exists \text{bi:title}^{-} \sqsubseteq$ string $\exists$bi:city ⊑ ⊥ $\exists \text{bi:city}^{-} \sqsubseteq \bot$		bi:Book ⊑ bi:Product bi:Music ⊑ bi:Product

Fig. 5(j). Constraints of the BN import schema.

To include BN in the Sales mediation environment, creating the Sales/BN mediation environment, we again perform three steps:

(Concept revision step) Assume for the sake of argument that the vocabulary of the Sales/BN mediated schema, with namespace "sr:", as in Figure 5(e), is equal to that of the Sales mediated schema. The vocabulary of the Sales mediated schema is still identified with namespace prefix "s:", as in Figure 5(f). The BN import schema has the vocabulary shown in Figure 5(g).

(Mapping revision step) Figure 5(h) shows the local mapping from the BN export schema to the BN import schema. Note that the definition bi:city ≡ ⊥ indicates that property bi:city will always be empty in the BN import schema.

Figure 5(i) depicts the mediated mapping of the Sales/BN mediation environment.

(Constraint revision step) Figure 5(j) contains the constraints of the BN import schema.

Note that the constraints $\exists$ bi:city ⊑ ⊥ and $\exists \text{ bi:city}^{-} \sqsubseteq \bot$ in Figure 5(j) follow from the definition bi:city ≡ ⊥ in Figure 5(h). Indeed, these constraints capture that bi:city is an empty property by saying that its domain and range are always empty. This strategy is necessary since the constraints we consider do not allow expressions of the form bi:city ⊑ ⊥. Furthermore, note that it is redundant (but not wrong) to add a constraint saying that the domain of bi:city is always empty, as well as a constraint saying that the range of bi:city is always empty.

Since the BN external schema has no explicit cardinality constraints, the BN import schema has no non-trivial cardinality constraints. However, ∃ bi:city ⊑ ⊥ logically implies that ⊤ ⊑ (≤k bi:city), where k is any positive integer. Hence, ∃ bi:city ⊑ ⊥ trivially implies maxCardinality constraints of the form e ⊑ (≤k bi:city), where e is any concept expression and k is any positive integer. Likewise, ∃ bi:city ⊑ ⊥ trivially implies disjointness constraints of the form ∃ sr:city | c, where c is any expression. Any of these constraints need not be made explicit since they will be in the theory of the constraints of the BN import schema. Similar observations apply to ∃ bi:city⁻ ⊑ ⊥.

We translate the set of constraints of the BN import schema to the vocabulary of the Sales/BN mediated schema simply by replacing bi:Book by sr:Book, etc. This results in the set of constraints Φ_B, where

(16) ∃ sr:title ⊑ sr:Product (in Φ_B)

(17) ∃ sr:title⁻ ⊑ string (in Φ_B)

(18) ∃ sr:city ⊑ ⊥ (in Φ_B)

(19) ∃ sr:city⁻ ⊑ ⊥ (in Φ_B)

(20) sr:Book ⊑ sr:Product (in Φ_B)

(21) sr:Music ⊑ sr:Product (in Φ_B)

Now, recalling that sr:city is an empty property in the BN import schema, *Th*(Φ_B) also contains

(22) ∃ sr:city ⊑ sr:Book (in *Th*(Φ_B))

(23) ∃ sr:city⁻ ⊑ string (in *Th*(Φ_B))

(24) ∃ sr:city ⊑ (≤ 1 sr:title) (in *Th*(Φ_B))

We also translate the set of constraints of the old Sales mediate schema, shown in Figure 5(b), to the vocabulary of the Sales/BN mediated schema, obtaining the set of constraints Φ_S, where

(25) ∃ sr:title ⊑ sr:Product (in Φ_S)

(26) ∃ sr:title⁻ ⊑ string (in Φ_S)

(27) ∃ sr:city ⊑ sr:Product (in Φ_S)

(28) ∃ sr:city⁻ ⊑ string (in Φ_S)

(29) sr:Product ⊑ (≤ 1 sr:title) (in Φ_S)

(30) `sr:Book` $\sqsubseteq$ ($\geq$ 1 `sr:city`) (in Φ_S)

(31) `sr:Book` $\sqsubseteq$ `sr:Product` (in Φ_S)

(32) `sr:Music` $\sqsubseteq$ `sr:Product` (in Φ_S)

(33) `sr:Book | sr:Music` (in Φ_S)

Observe that, by (27) and (29), $Th(\Phi_S)$ contains the following constraint:

(34) $\exists$ `sr:city` $\sqsubseteq$ ($\leq$ 1 `sr:title`)

The constraints of the (revised) Sales/BN mediated schema are then computed as:

$$SC_r = \Phi_B \triangle \Phi_S = Th(\Phi_B) \cap Th(\Phi_S)$$

Figure 5(k) lists the constraints in SC_r. By inspection, observe that $SC_r = Th(\Phi_B) \cap Th(\Phi_S)$ contains:

- the domain and range constraints for `sr:title`, by (16), (17), (25) and (26)
- the domain and range constraints for `sr:city`, by (22), (23), (27) and (28)
- the subset constraints for `sr:Product`, by (20), (21), (31) and (32)
- a single cardinality constraint, of a rather unanticipated nature, by (24) and (34)
- no disjointness constraints since $Th(\Phi_S)$ does not contain any of the trivial disjointness constraints in $Th(\Phi_B)$ of the form $\exists$ `sr:city | C` or of the form $\exists$ `sr:city`$^{-}$ `| C`, where `C` is any expression. □

$\exists$ `sr:title` $\sqsubseteq$ `sr:Product` $\exists$ `sr:title`$^{-}$ $\sqsubseteq$ `string` $\exists$ `sr:city` $\sqsubseteq$ `sr:Product` $\exists$ `sr:city`$^{-}$ $\sqsubseteq$ `string`	$\exists$ `sr:city` $\sqsubseteq$ ($\leq$ 1 `s:title`)	`sr:Book` $\sqsubseteq$ `sr:Product` `sr:Music` $\sqsubseteq$ `sr:Product`

Fig.5(k). Constraints of the revised Sales/BN mediated schema.

4.2 Computing the Greatest Lower Bound of Two Sets of Constraints

The solution to the least constraint revision problem, outlined up to this point, gives no indication on how to generate the revised set of constraints of the mediated schema. In this section, we then show how to compute the g.l.b. of two sets of constraints, with the help of examples. In the Appendix, we prove the results that justify the techniques introduced.

Recall from Section 3.2 that the constraints of a schema are of one of the following forms:

- $\exists P \sqsubseteq D$ (property P has domain D)
- $\exists P^{-} \sqsubseteq R$ (property P has range R)

- $C \sqsubseteq (\geq k\ P)$ or $C \sqsubseteq (\geq k\ P^{-})$
  (P or P^{-} maps each individual in C to at least k distinct individuals)
- $C \sqsubseteq (\leq k\ P)$ or $C \sqsubseteq (\leq k\ P^{-})$
  (P or P^{-} maps each individual in C to at most k distinct individuals)
- $C \sqsubseteq D$ (class C is a subclass of class D)
- $C \mid D$ (classes C and D are disjoint)

We also admit constraints of one of the forms:

- $C \sqsubseteq \bot$ (class C is always empty)
- $\exists P \sqsubseteq \bot$ or $\exists P^{-} \sqsubseteq \bot$ (property P is always empty, i.e., P has an empty domain or an empty range)

We normalize a set of constraints by rewriting:

- $\exists P \sqsubseteq D$ as $(\geq 1\ P) \sqsubseteq D$
- $\exists P^{-} \sqsubseteq R$ as $(\geq 1\ P^{-}) \sqsubseteq R$
- $C \sqsubseteq (\leq k\ P)$ as $C \sqsubseteq \neg(\geq k{+}1\ P)$
- $C \sqsubseteq (\leq k\ P^{-})$ as $C \sqsubseteq \neg(\geq k{+}1\ P^{-})$
- $C \mid D$ as $C \sqsubseteq \neg D$ (or, equivalently, $D \sqsubseteq \neg C$)
- $\exists P \sqsubseteq \bot$ as $(\geq 1\ P) \sqsubseteq \bot$
- $\exists P^{-} \sqsubseteq \bot$ as $(\geq 1\ P^{-}) \sqsubseteq \bot$

We observe that, after normalization, negated expressions (including the bottom concept $\bot$) occur only on the right-hand side of the constraints.

The question of computing the greatest lower bound of two sets of constraints is not straightforward since constraints may interact in unanticipated ways, as the following simple example illustrates.

Example 5: Suppose that Σ={ $A \sqsubseteq B$, $A \sqsubseteq C$, $B \mid C$ }. Since B and C are disjoint and A is a subset of both B and C, the set of constraints Σ implies that A will always be empty, that is, $\Sigma \models A \sqsubseteq \bot$.

As a second example, assume that Σ={ $A \sqsubseteq (\leq m\ P)$, $A \sqsubseteq (\geq n\ P)$ }. Suppose that $m<n$. Then, since $(\leq m\ P)$ and $(\geq n\ P)$ denote disjoint sets, and A is a subset of both constraints, we again have that $\Sigma \models A \sqsubseteq \bot$.

Finally, note that $A \sqsubseteq \bot$ logically implies $A \sqsubseteq e$, for any expression e, which affects how we compute $Th(\Sigma)$ and, consequently, how we compute $\Sigma \vartriangle \Gamma$, where Γ is a second set of constraints. □

The first sequence of definitions indicates how to construct a graph that captures the structure of a set of constraints.

We say that the *complement* of a non-negated expression e is $\neg e$, and vice-versa; furthermore, the *complement* of $\bot$ is $\top$, and vice-versa. We denote the complement of an expression c by $\bar{c}$. A *constraint expression* is an expression that may occur on the right- or left-hand sides of a normalized constraint.

Let Σ be a set of normalized constraints and Ω be a set of constraint expressions (we leave the alphabet understood from the context). The labeled graph $g(\Sigma,\Omega)=(\gamma,\delta,\kappa)$ that *captures* Σ and Ω, where κ labels each node with an expression, is defined as follows:

(i) For each concept expression e that occurs on the right- or left-hand side of an inclusion in Σ, or that occurs in Ω, there is exactly one node in γ labeled with e.

(ii) For each atomic role P, there is exactly one node in γ labeled with P (this is just a theoretical convenience, explored in the appendix).

(iii) If there is a node in γ labeled with a concept expression e, then there must be exactly one node in γ labeled with $\bar{e}$.

(iv) For each constraint $e \sqsubseteq f$ in Σ, there is an arc (M,N) in δ, where M and N are the nodes labeled with e and f, respectively.

(v) If there are nodes M and N in γ labeled with $(\geq m\ p)$ and $(\geq n\ p)$, where p is either P or P^- and $m<n$, then there is an arc (N,M) in δ.

(vi) If there is an arc (M,N) in δ, where M and N are the nodes labeled with e and f respectively, then there is an arc (K,L) in δ, where K and L are the nodes labeled with $\bar{f}$ and $\bar{e}$, respectively.

(vii) These are the only nodes and arcs of $g(\Sigma)$.

The labeled graph $G(\Sigma,\Omega)=(\eta,\varepsilon,\lambda)$ that *represents* Σ and Ω, where λ labels each node with a set of expressions, is defined from $g(\Sigma,\Omega)$ by collapsing each clique of $g(\Sigma,\Omega)$ into a single node labeled with the expressions that previously labeled the nodes in the clique. When Ω is the empty set, we simply write $G(\Sigma)$ and say that the graph *represents* Σ.

If a node K of $G(\Sigma,\Omega)$ is labeled with an expression e, then $\overline{K}$ denotes the node labeled with $\bar{e}$ (which may be K itself). We say that K and $\overline{K}$ are *dual nodes* of $G(\Sigma,\Omega)$.

A node K of $G(\Sigma,\Omega)$ is a *$\bot$-node with level n*, for a non-negative integer n, iff one of the following conditions hold:

(i) K is is a *$\bot$-node with level 0* iff

a. K is labeled with $\bot$, or

b. There are nodes M and N, not necessarily distinct from K, and a non-negated concept expression h such that M and N are respectively labeled with h and $\neg h$, and there are paths in $G(\Sigma,\Omega)$ from K to M and from K to N.

(ii) K is is a $\bot$-*node with level* $n+1$ iff

a. There is a $\bot$-node M of level n, distinct from K, such that there is a path in $G(\Sigma,\Omega)$ from K to M, and M is the $\bot$-node with the smallest level such that there is a path in $G(\Sigma,\Omega)$ from K to M, or

b. K is labeled with a minCardinality constraint of the form ($\geq 1\ P$) (or of the form ($\geq 1\ P^-$)) and there is a $\bot$-node M of level n, distinct from K, such that M is labeled with ($\geq 1\ P^-$) (or with ($\geq 1\ P$)), and M is the $\bot$-node with the smallest level labeled with ($\geq 1\ P^-$) or ($\geq 1\ P$).

A node K is a $\bot$-*node* iff K is a $\bot$-node with level n, for some non-negative integer n, and K is a $\top$-*node* iff $\bar{K}$ is a $\bot$-node.

Proposition 5 in the appendix lists the properties of $G(\Sigma,\Omega)$ that matter to our formal development. An informal account of some properties that help understand the construction of $G(\Sigma,\Omega)$ is:

- There is a path in $G(\Sigma,\Omega)$ from a node labeled with e to a node labeled with f iff there is a path in $G(\Sigma,\Omega)$ from a node labeled with $\bar{f}$ to a node labeled with $\bar{e}$.
- If two concept expressions, e and f, label the same node of $G(\Sigma,\Omega)$, then $\Sigma \models e \equiv f$, that is, Σ forces e and f to denote the same set of individuals.
- If a concept expression e labels a $\bot$-node with level 0 of $G(\Sigma,\Omega)$, then $\Sigma \models e \sqsubseteq \bot$, that is, Σ forces e to denote an empty set of individuals.
- If there is a path in $G(\Sigma,\Omega)$ from a node labeled with e to a node labeled with f, then $\Sigma \models e \sqsubseteq f$.

Example 6: Consider the constraints of the Sales mediated schema, listed in Figure 5(b). Abbreviate the names of the classes and properties by just their first letter, ignoring the namespace prefix. Let Σ be the set obtained by normalizing such constraints:

(1)	$\exists\ t \sqsubseteq P$	normalized as: $(\geq 1\ t) \sqsubseteq P$
(2)	$\exists\ t^- \sqsubseteq S$	normalized as: $(\geq 1\ t^-) \sqsubseteq S$
(3)	$\exists\ c \sqsubseteq P$	normalized as: $(\geq 1\ c) \sqsubseteq P$
(4)	$\exists\ c^- \sqsubseteq S$	normalized as: $(\geq 1\ c^-) \sqsubseteq S$
(5)	$P \sqsubseteq (\leq 1\ t)$	normalized as: $P \sqsubseteq \neg(\geq 2\ t)$
(6)	$B \sqsubseteq (\geq 1\ c)$	
(7)	$B \sqsubseteq P$	
(8)	$M \sqsubseteq P$	
(9)	$B \mid M$	normalized as: $B \sqsubseteq \neg M$

Figure 6 depicts $g(\Sigma)$, the graph capturing Σ, using the normalized form of the constraints. In this case, $g(\Sigma)$ is equal to $G(\Sigma)$, the graph representing Σ. By inspecting $G(\Sigma)$, note that:

- There is a path from the node labeled with ($\geq$1 c) to the node labeled with $\neg$($\geq$2 t), which implies that

(10) $\Sigma \models (\geq 1\ c) \sqsubseteq \neg(\geq 2\ t)$

- There are paths from the node K labeled with ($\geq$2 t) to the node labeled with $\neg P$ and the node labeled with P. Hence, K is a $\perp$-node with level 0, which implies that

(11) $\Sigma \models (\geq 2\ t) \sqsubseteq \perp$

Intuitively, t never maps an individual to two or more individuals, in the presence of the constraints in Σ. □

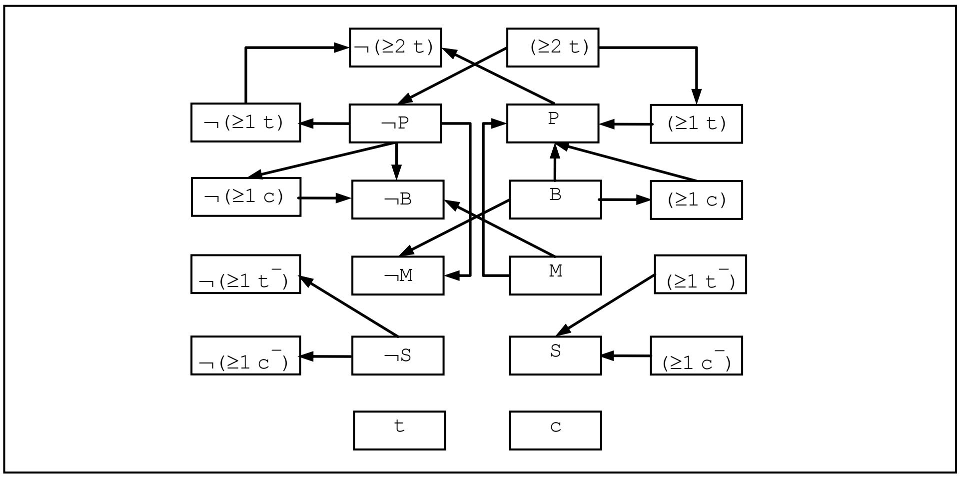

Fig. 6. The graph $G(\Sigma)$ that represents Σ.

Example 7: Consider the constraints of the BN import schema, listed in Figure 5(j), and again abbreviate the names of the classes and properties by just their first letter, ignoring the namespace prefix for the moment. Let Φ be the set obtained by normalizing such constraints:

(12)	$\exists\ t \sqsubseteq P$	normalized as: $(\geq 1\ t) \sqsubseteq P$
(13)	$\exists\ t^{-} \sqsubseteq S$	normalized as: $(\geq 1\ t^{-}) \sqsubseteq S$
(14)	$\exists\ c \sqsubseteq \perp$	normalized as: $(\geq 1\ c) \sqsubseteq \perp$
(15)	$\exists\ c^{-} \sqsubseteq \perp$	normalized as: $(\geq 1\ c^{-}) \sqsubseteq \perp$
(16)	$B \sqsubseteq P$	
(17)	$M \sqsubseteq P$	

Figure 7 depicts the graph $G(\Phi)$ representing Φ (using the normalized form of constraints). □

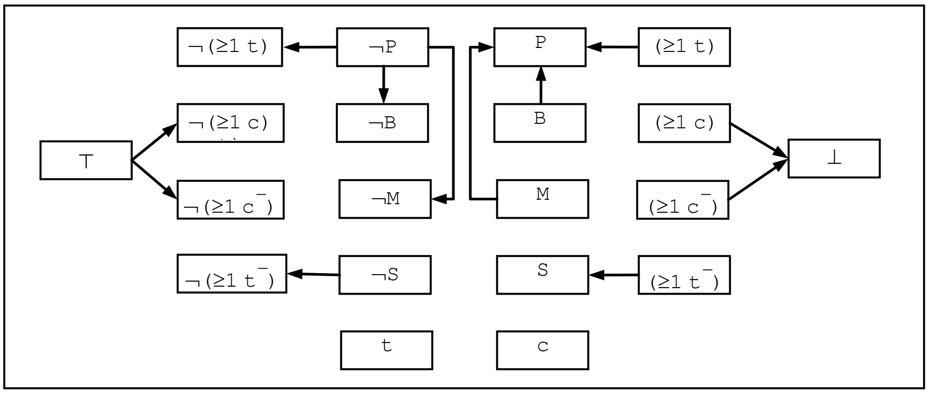

Fig. 7. The graph $G(\Phi)$ that represents Φ.

Let Σ be a set of normalized constraints. The computation of $Th(\Sigma)$ is based on the following result, which Example 6 already partly illustrates.

Theorem 2: Let Σ be a set of normalized constraints. Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω. Then, $\Sigma \models e \sqsubseteq f$ iff one of the following conditions holds:

(i) The node of $G(\Sigma,\Omega)$ labeled with e is a $\bot$-node; or

(ii) The node of $G(\Sigma,\Omega)$ labeled with f is a $\top$-node; or

(iii) There is a path in $G(\Sigma,\Omega)$ from the node labeled with e to the node labeled with f.□

Corollary 1: Let Σ be a set of normalized constraints. Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω, and $G(\Sigma)$ be the graph that represents Σ. Suppose that $\Sigma \models e \sqsubseteq f$. Then:

(a) Either e labels a node of $G(\Sigma)$ or e is of the form $(\geq k\ P)$ and there is a node of $G(\Sigma)$ labeled with $(\geq j\ P)$, where $j<k$.

(b) Either f labels a node of $G(\Sigma)$ or f is of the form $\neg(\geq n\ P)$ and there is a node of $G(\Sigma)$ labeled with $\neg(\geq m\ P)$, where $m<n$. □

Let Σ_1 and Σ_2 be two sets of normalized constraints. Let $G(\Sigma_1)$ and $G(\Sigma_2)$ be the graph that represent Σ_1 and Σ_2. Denote their transitive closure by $G^*(\Sigma_1)$ and $G^*(\Sigma_2)$. Based on Theorem 2 and Corollary 1, we then construct a set of constraints Γ that *generates the g.l.b. of Σ_1 and Σ_2* as follows:

- A constraint $e \sqsubseteq f$ is in Γ iff there are $i, j \in \{1,2\}$, with $i \neq j$, such that one of the following conditions holds

  (a) There is a $\bot$-node M of $G(\Sigma_i)$ and a $\bot$-node P of $G(\Sigma_j)$ and

  - e is a non-negated constraint expression that labels both M and P
  - f is the bottom concept $\bot$

  (b) There is a $\bot$-node M of $G(\Sigma_i)$ and an arc (P,Q) of $G^*(\Sigma_j)$ such that P is not a $\bot$-node of $G(\Sigma_j)$ and

  - e is a non-negated constraint expression that labels both M and P
  - f is a constraint expression that labels Q

  (c) There is a $\top$-node N of $G(\Sigma_i)$ and an arc (P,Q) of $G^*(\Sigma_j)$ such that Q is not a $\top$-node of $G(\Sigma_j)$ and

  - e is a non-negated constraint expression that labels P
  - f is a constraint expression that labels both N and Q

  (d) There is an arc (M,N) of $G^*(\Sigma_i)$ and an arc (P,Q) of $G^*(\Sigma_j)$ such that M, N, P or Q is not a $\bot$-node or a $\top$-node and

  - e is a non-negated constraint expression that labels both M and P
  - f is a constraint expression that labels both N and Q

Note that Γ is a (normalized) set of constraints since, by construction, e is always a non-negated constraint expression and f is a constraint expression. Furthermore, note that Γ can be constructed in $O(n^2)$, where $n=max(n_1,n_2)$ and n_i is the number of nodes of $G(\Sigma_i)$. However, we do not claim that Γ is the best set of constraints that generates $\Sigma \bigtriangleup \Phi$, in the sense of having the smallest number of constraints. Corollary 2 indicates that Γ is correctly constructed, in the sense that $Th(\Gamma) = \Sigma \bigtriangleup \Phi = Th(\Sigma) \cap Th(\Phi)$, and follows from Theorem 2, Corollary 1 and the definition of Γ.

Corollary 2: Let Σ_1 and Σ_2 be two sets of normalized constraints. Let Γ be the set of constraints that generates the g.l.b. of Σ_1 and Σ_2. Then, we have

(i) $Th(\Gamma) = \Sigma_1 \bigtriangleup \Sigma_2$.

(ii) Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Gamma,\Omega)$ be the graph that represents Γ and Ω. Then, $e \sqsubseteq f$ is in $\Sigma_1 \bigtriangleup \Sigma_2$ iff one of the conditions holds:

  (a) The node of $G(\Gamma,\Omega)$ labeled with e is a $\bot$-node; or

  (b) The node of $G(\Gamma,\Omega)$ labeled with f is a $\top$-node; or

(c) There is a path in $G(\Gamma,\Omega)$ from the node labeled with e to the node labeled with f. □

We close the section with a final example that illustrates how to systematically obtain the set of constraints of the (revised) Sales/BN mediated schema informally derived in Step (C) of Example 4.

Example 8: Let Σ be the set of normalized constraints of Example 6, and Φ be the set of normalized constraints of Example 7. Figure 5(k) shows the set of (unnormalized) constraints Γ that generates the g.l.b. of Σ and Φ. Again abbreviating the names of the classes and properties by their first letter, and ignoring the namespace prefix, the constraints in Γ are:

(1)	$\exists\, t \sqsubseteq P$	normalized as: $(\geq 1\ t) \sqsubseteq P$
(2)	$\exists\, t^{-} \sqsubseteq S$	normalized as: $(\geq 1\ t^{-}) \sqsubseteq S$
(3)	$\exists\, c \sqsubseteq P$	normalized as: $(\geq 1\ c) \sqsubseteq P$
(4)	$\exists\, c^{-} \sqsubseteq S$	normalized as: $(\geq 1\ c^{-}) \sqsubseteq S$
(5)	$\exists\, c \sqsubseteq (\leq 1\ t)$	normalized as: $(\geq 1\ c) \sqsubseteq \neg(\geq 2\ t)$
(6)	$B \sqsubseteq P$	
(7)	$M \sqsubseteq P$	

Consider the graph $G(\Sigma)$ of Figure 6 and the graph $G(\Phi)$ of Figure 7. We systematically construct Γ as follows. Tables 1(a) and 1(b) show the arcs of $G^*(\Sigma)$ and $G^*(\Phi)$. Note that a tabular presentation of the arcs, as opposed to a graphical representation, is much more convenient since we are working with transitive closures. For example, line 3 of Table 1(a) indicates that $G^*(\Sigma)$ has arcs from the node labeled with B to the three nodes respectively labeled with $(\geq 1\ c)$, P and $\neg M$.

In this specific example, Table 1(c) induces Γ as follows:

(8) Lines 1, 7, 9 and 12 are discarded since they correspond to arcs in just one of the graphs $G^*(\Sigma)$ or $G^*(\Phi)$.

(9) Lines 2 and 5 are discarded since they have a negated expression on the left-hand side cell.

(10) Lines 8 and 11 correspond to Case (b) of the definition of Γ.

(11) Lines 3, 4, 6 and 10 correspond to Case (d) of the definition of Γ. The case corresponding to lines 2 and 5 deserves an additional comment. Consider line 5, for example. Note that the pair $(\neg S, \neg(\geq 1\ t^{-}))$ occurs in line 5 in Tables 1(a) and 1(b). However, we need note add $\neg S \sqsubseteq \neg(\geq 1\ t^{-})$ to Γ since line 10 forces the addition of the equiva-

lent constraint $(\geq 1\ t^{-}) \sqsubseteq S$. Finally, we warn the reader that the example does not illustrate all cases of the definition of Γ. □

Table 1: Construction of the set of constraints Γ that generates $\Sigma \triangle \Phi$.

	(a) $G^*(\Sigma)$		(b) $G^*(\Phi)$		(c) Γ	
1	P	$\neg(\geq 2\ t)$				
2	$\neg P$	$\neg(\geq 1\ t)$ $\neg(\geq 2\ t)$ $\neg(\geq 1\ c)$ $\neg B$ $\neg M$	$\neg P$	$\neg(\geq 1\ t)$ $\neg B$ $\neg M$		
3	B	$(\geq 1\ c)$ P $\neg M$	B	P	B	P
4	M	$\neg B$ P $\neg(\geq 2\ t)$	M	P	M	P
5	$\neg S$	$\neg(\geq 1\ t^{-})$ $\neg(\geq 1\ c^{-})$	$\neg S$	$\neg(\geq 1\ t^{-})$		
6	$(\geq 1\ t)$	P $\neg(\geq 2\ t)$	$(\geq 1\ t)$	P	$(\geq 1\ t)$	P
7	$\neg(\geq 1\ t)$	$\neg(\geq 2\ t)$				
8	$(\geq 1\ c)$	P $\neg(\geq 2\ t)$	$(\geq 1\ c)$	$\bot$	$(\geq 1\ c)$	P $\neg(\geq 2\ t)$
9	$\neg(\geq 1\ c)$	$\neg B$				
10	$(\geq 1\ t^{-})$	S	$(\geq 1\ t^{-})$	S	$(\geq 1\ t^{-})$	S
11	$(\geq 1\ c^{-})$	S	$(\geq 1\ c^{-})$	$\bot$	$(\geq 1\ c^{-})$	S
12			$\top$	$(\geq 1\ c)$ $\neg(\geq 1\ c)$		

5 Conclusions

In this paper, we addressed the problem of changing the constraints of a mediated schema to accommodate the set of constraints of a new export schema. We argued that such problem can be solved by computing the greatest lower bound of two sets of constraints. The approach we took to define the mediation environment is akin to the idea of exact views. Yet, we considered that constraints should be included in the mediated schema to capture the common semantics of the data sources.

For the family of extralite schema, we described efficient procedures to decide logical implication and to compute the greatest lower bound of two sets of constraints. The procedures essentially explore the structure of a set of constraints, captured as a graph. However, cardinality constraints posed considerable technical problems to the proof of the theorems, which we overcame with the help of the notion of canonical Herbrand interpretation, introduced in the appendix. These developments are new, and cover an expressive and useful family of constraints, which justifies the proofs included in the appendix.

As for future work, we plan to extend the schema matching framework described in [20] to a full-fledged tool that helps create mediation environments, by including the strategy described in this paper. Additional work should also be devoted to minimize the set of constraints that generates $\Sigma \triangle \Phi$, which will require a careful analysis of the graphs that represent Σ and Φ.

Acknowledgements. This work was partly supported by CNPq under grants 301497/2006-0, 473110/2008-3, 557128/2009-9, FAPERJ E-26/170028/2008, and CAPES/PROCAD NF 21/2009.

References

1. Amann, B.; Beeri, C.; Fundulaki, I; Scholl, M. (2002) "Querying xml sources using an ontology-based mediator". In: *CoopIS/DOA/ODBASE*, pp. 429–448.
2. Aspvall, B.; Plass, M.F; Tarjan, R.E. (1979) "A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas". *Information Processing Letters*, Vol. 8, No. 3 (March 1979), pp. 121–123.
3. Atzeni, P.; Cappellari, P.; Torlone, R.; Bernstein, P.A.; Gianforme, G. (2008) "Model-independent schema translation". *The VLDB Journal*, Vol. 17, No. 6, pp. 1347–1370.
4. Baader, F.; Nutt, W. (2003) "Basic Description Logics". In: Baader, F.; Calvanese, D.; McGuiness, D.L.; Nardi, D.; Patel-Schneider, P.F. (Eds) *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, UK.
5. Bernstein, P.; Melnik, S. (2007) "Model management 2.0: manipulating richer mappings". In: Proc. 27th ACM SIGMOD Int'l. Conf. Management of Data, Beijing, China, pp. 1–12.
6. Bilke A. (2007): Duplicate-based Schema Matching. PhD Thesis. Berlin University.
7. Breitman, K.; Casanova, M.; Truszkowski, W. (2007) *Semantic web: concepts, technologies, and applications*. Springer, London.

8. Bilke, A.; Naumann, F. (2005) “Schema matching using duplicates”. In: Proc. 21st Int'l. Conf. on Data Engineering (Apr 2005), pp. 69–80.

9. Brauner, D.F.; Casanova, M.A.; Milidiu, R. (2007) “Towards Gazetteer Integration Through an Instance-based Thesauri Mapping Approach”. In: Clodoveu A. Davis Jr; Antonio M.V.M. Monteiro. (eds.). *Advances in Geoinformatics*. Heidelberg: Springer, 2007, pp. 235–245.

10. Brauner, D.F.; Gazola, A.; Casanova, M.A.; Breitman, K.K. (2008) “Adaptative Matching of Database Web Services Export Schemas”. In: Proc. of ICEIS 2008 – Tenth International Conference on Enterprise Information Systems. Lisboa: INSTICC – Institute for Systems and Technologies of Information, Control and Communication, 2008. pp.49 – 56.

11. Calì, A., Calvanese, D., De Giacomo, G., Lenzerini, M. (2002) “Data Integration under Integrity Constraints”. In: Proc. 14th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2002). Volume 2348 of Lecture Notes in Computer Science.

12. Calì, A.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Naggar, P.; Vernacotola, F. (2003) “IBIS: Semantic data integration at work”. In: Proc. of the 15th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2003), pp. 79–94.

13. Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rosati, R.; Ruzzi, M. (2008). “Data Integration through DL-Lite-A Ontologies”. In: Proc. 3rd Int’l. Workshop on Semantics in Data and Knowledge Bases, pp. 26–47.

14. Casanova, M.A.; Breitman, K.K.; Brauner, D.F.; Marins, A. (2007) “Database Conceptual Schema Matching”. *Computer (Long Beach)*, v. 40, pp. 102–104.

15. Chawathe, S.S.; Garcia-Molina, H.; Hammer, J.; Ireland, K.; Papakonstantinou, Y.; Ullman, J.D.; Widom, J. (1994) “The TSIMMIS project: Integration of heterogeneous information sources”. In: Proc. 10th Meeting of the Information Processing Society of Japan (IPSJ’94), pp. 7–18.

16. Donini, F.M. (2003) “Complexity of Reasoning”. In: Baader, F.; Calvanese, D.; McGuiness, D.L.; Nardi, D.; Patel-Schneider, P.F. (Eds) *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, UK.

17. Euzenat, J.; Shvaiko, P. (2007) *Ontology matching*. Springer-Verlag.

18. Franconi, E. (2002) "Structural Description Logics: FL-". In: *Description Logics Course*. Available at: http://www.inf.unibz.it/~franconi/dl/course/slides/struct-DL/flminus.pdf

19. Garcia-Molina, H.; Papakonstantinou, Y.; Quass, D.; Rajaraman, A.; Sagiv, Y.; Ullman, J.D.; Vassalos, V.; Widom, J. (1997) “The TSIMMIS approach to mediation: Data models and languages”. *J. of Intelligent Information Systems*, Vol. 8, No. 2, pp. 117–132.

20. Gomes, R.V.; Leme, L.A.P.P; Casanova, M.A. (2010) “MatchMaking – A Tool to Match OWL Schemas”, *Journal of Theoretical and Applied Informatics (ER2009 Posters and Demos session)*, 17.

21. Goasdoué, F.; Lattes, V.; Rousset, M-C. (2000) “The use of CARIN language and algorithms for information integration: The Picsel system”. *Int. J. of Cooperative Information Systems*, Vol. 9, No. 4, pp. 383–401.

22. Halevy A. (2001) “Answering queries using views: A survey”. *VLDB Journal*, Vol. 10, No. 4, pp. 270–294.

23. Hartmanna, S.; Linkb, S.; Trinha, T. (2009) “Constraint acquisition for Entity-Relationship models” , *Data & Knowledge Engineering*, Volume 68, Issue 10, October 2009, pp. 1128-1155.

24. Hick, J-M.; Hainauta, J-H. (2006) “Database application evolution: A transformational approach”, *Data & Knowledge Engineering*, Volume 59, Issue 3, December 2006, pp. 534-558.

25. Köpcke, H.; Rahma, E. (2010) "Frameworks for entity matching: A comparison", *Data & Knowledge Engineering,* Volume 69, Issue 2, February 2010, pp. 197-210.

26. Lauschner, T.; Casanova, M.A.; Vidal, V.M.P.; Macedo, J.A. (2009) "Efficient Decision Procedures for Query Containment and Related Problems". In: Proc. XXIV Brazilian Symposium on Databases, Fortaleza. Oct. 2009

27. Leme, L. A. P.; Brauner, D. F.; Breitman, K. K.; Casanova, M. A., and Gazola, A. "Matching object catalogues". *J. Innovations in Systems and Software Engineering* 4(4), Springer, pp. 315–328.

28. Leme, L.A.P.; Casanova, M.A.; Breitman, K.K; Furtado, A.L. (2009) "Instance-based OWL Schema Matching". Proc. 11th Int'l. Conf. on Enterprise Inf. Systems, Milan, Italy.

29. Lenzerini, M. (2002) "Data Integration: A Theoretical Perspective". In: Proc. ACM Symposium on Principles of Database Systems.

30. Leone, N.; Eiter, T.; Faber, W.; Fink, M.; Gottlob, G.; Greco, G.; Kalka, E.; Ianni, G.; Lembo, D.; Lenzerini, M.; Lio, V.; Nowicki, B.; Rosati, R.; Ruzzi, M.; Staniszkis, W.; Terracina, G. (2005) "The INFOMIX system for advanced integration of incomplete and inconsistent data". In: Proc. of the ACM SIGMOD Int. Conf. on Management of Data, pp. 915–917.

31. Levesque, H.J.; Brachman, R.J. (1987) "Expressiveness and tractability in knowledge representation and reasoning". *Computational Intelligence* 3, p. 78-93.

32. Lonsdale, D.; Embley, D.W.; Dinga, Y.; Xub, L.; Heppc,M. (2010) "Reusing ontologies and language components for ontology generation", *Data & Knowledge Engineering,* Vol. 69, Issue 4, April 2010, pp. 318-330.

33. Manolescu, I.; Florescu, D.; Kossmann, D.; Xhumari, F.; Olteanu, D. (2000) "Agora: Living with XML and Relational". In: Proc. 26th Int. Conf. on Very Large Data Bases, pp. 623-626.

34. Madhavan J.; Bernstein, P.A.; Rahn, E. (2001) "Generic schema matching with Cupid". In: Proc. 27th Int'l. Conf. on Very Large Data Bases, pp. 49–58.

35. Melnik, S.; Garcia-Molina, H.; Rahm, E. (2002) "Similarity flooding: a versatile graph matching algorithm and its application to schema matching". In: Proc. 18th Int'l. Conf. on Data Engineering, pp. 117–128.

36. Nardi, D.; Brachman, R.J. (2003) "An Introduction to Description Logics". In: Baader, F.; Calvanese, D.; McGuiness, D.L.; Nardi, D.; Patel-Schneider, P.F. (Eds) The *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, Cambridge, UK.

37. Papotti, P.; Torlone, R. (2009) "Schema exchange: Generic mappings for transforming data and metadata", *Data & Knowledge Engineering*, Volume 68, Issue 7, July 2009, pp. 665-682.

38. Qi He, O.; Linga, T.W. (2006) "An ontology based approach to the integration of entity–relationship schemas", *Data & Knowledge Engineering,* Vol. 58, Issue 3, Sept. 2006, pp. 299-326.

39. Rahn, E.; Bernstein, P. (2001) "A survey of approaches to automatic schema matching". *The VLDB Journal* 10, 4 (2001), pp. 334–350.

40. Rull, G.; Farré, C.; Teniente, E.; Urpí, T. (2008) "Validation of mappings between schemas", *Data & Knowledge Engineering,* Volume 66, Issue 3, September 2008, pp. 414-437.

41. Simperla, E. (2009) "Reusing ontologies on the Semantic Web: A feasibility study", *Data & Knowledge Engineering,* Volume 68, Issue 10, October 2009, pp. 905-925.

42. Wang, J.; Wen, J.; Lochovsky, F.; Ma, W. (2004) "Instance-based schema matching for web databases by domain-specific query probing". In: Proc. 13th Int'l. Conf. on Very Large Data Bases (Aug 2004), pp. 408–419.

43. Zhaoa, H.; Ramb, S. (2007) "Combining schema and instance information for integrating heterogeneous data sources", *Data & Knowledge Engineering* , Vol. 61, Issue 2, May 2007, pp 281-303.

Appendix - Proofs of the Main Results

A.1 Proof of Theorem 1

In this section, we prove Theorem 1 that states that the revised set of constraints of the mediated schema can be taken as the g.l.b. of the current set of constraints of the mediated schema and the set of constraints of the new import schema, without impairing consistency preservation. We follow the same notation as in Section 4.1, not repeated here for brevity.

Theorem 1: Let $MC_r = IC_0[IV_0 \rightarrow MV_r] \triangle MC^+[MV^+ \rightarrow MV_r]$. Suppose that:

(i) *(Domain Disjointness Assumption)* Any pair of interpretations for V and V_0 have disjoint domains.

(ii) The mediated mapping γ and the local mapping $\gamma_1,...,\gamma_n$ induce a mapping from consistent states of $E_1,...,E_n$ into consistent states of M.

(iii) The local mapping γ_0 induces a mapping from consistent states of E_0 into consistent states of I_0.

Then, the revised mediated mapping γ_r and the local mappings $\gamma_0,\gamma_1,...,\gamma_n$ induce a mapping from consistent states of EC_0, EC_1,..., EC_n into states of the revised mediated schema that satisfy MC_r.

Proof

Let $\sigma \in Th(MC_r)$. Then, by definition of g.l.b., we have:

- $\sigma \in Th(IC_0[IV_0 \rightarrow MV_r])$
- $\sigma \in Th(MC^+[MV^+ \rightarrow MV_r])$

But, by definition of the canonical translation functions, we have:

- $\sigma \in Th(IC_0[IV_0 \rightarrow MV_r])$ iff $\sigma[MV_r \rightarrow IV_0] \in Th(IC_0)$
- $\sigma \in Th(MC^+[MV^+ \rightarrow MV_r])$ iff $\sigma[MV_r \rightarrow MV^+] \in Th(MC^+)$

Let $k=0,\ldots,n$. Let $\boldsymbol{s}_k$ be a consistent state of E_k. Since $\bar{\gamma}_k$ preserves consistency, $\bar{\gamma}_k(\boldsymbol{s_k}) = \boldsymbol{t}_k$ is a consistent state of I_k. Furthermore, since $\bar{\gamma}$ preserves consistency, $\bar{\gamma}(\boldsymbol{s_1},...\boldsymbol{s_n}) = \boldsymbol{s}$ is a consistent state of M. Note that, by definition of MC^+, $\boldsymbol{s}$ is also consistent with respect to MC^+.

Therefore, we have

- $\boldsymbol{s}_0 \models \sigma[MV_r \rightarrow IV_0]$
- $\boldsymbol{s} \models \sigma[MV_r \rightarrow MV^+]$

Recall that $\sigma[MV_r \rightarrow MV^+ \cup IV_0]$ denotes the constraint obtained from σ by replacing each class C_i^r of MV_r by the union expression $C_i^0 \sqcup C_i$, where C_i^0 and C_i respectively are the classes of I_0 and M that match C_i^r, and likewise for the properties of MV_r. Note that $\sigma[MV_r \rightarrow MV^+ \cup IV_0]$ is a constraint written in $MV^+ \cup IV_0$, the union of the vocabularies MV^+ and IV_0.

Let $\boldsymbol{s} \cup \boldsymbol{t_0}$ denote the interpretation for $MV^+ \cup IV_0$ induced by $\boldsymbol{s}$ and $\boldsymbol{t_0}$ in the obvious way. Then, using the domain disjointness assumption, we can prove that:

- $\boldsymbol{s} \cup \boldsymbol{t_0} \models \sigma[MV_r \rightarrow MV^+ \sqcup IV_0]$

Now, by definition by definition of $\bar{\gamma}_r$, from (7), we finally have:

- $\bar{\gamma}_r(\boldsymbol{s}, \boldsymbol{t_0}) \models \sigma$

Therefore, recalling that $\bar{\gamma}(\boldsymbol{s_1},...\boldsymbol{s_n}) = \boldsymbol{s}$, we finally have that $\bar{\gamma}_r(\bar{\gamma}(\boldsymbol{s_1},...\boldsymbol{s_n}), \boldsymbol{t_0})$ is a consistent state of M_r, as desired. □

A.2 Proof of Theorem 2

In this section, we prove Theorem 2, which leads to efficient ways to construct the theory of a set of constraints, and to construct the greatest lower bound of two sets of constraints. To facilitate reading the section, we start by repeating a few definitions already introduced in Section 4.2.

We say that the *complement* of a non-negated expression e is $\neg e$, and vice-versa; furthermore, the *complement* of $\perp$ is $\top$, and vice-versa. If c is an expression, we denote its complement by $\bar{c}$.

A *constraint expression* is an expression that may occur on the right- or left-hand sides of a normalized constraint.

Let Σ be a set of normalized constraints and Ω be a set of constraint expressions (we leave the alphabet understood from the context).

Definition 1: The labeled graph $g(\Sigma,\Omega)=(\gamma,\delta,\kappa)$ that *captures* Σ and Ω, where κ labels each node with an expression, is defined as follows:

(i) For each concept expression e that occurs on the right- or left-hand side of an inclusion in Σ, or that occurs in Ω, there is exactly one node in γ labeled with e.

(ii) For each atomic role P, there is exactly one node in γ labeled with P (this is just a theoretical convenience, explored in Definitions 6, 7 and 8).

(iii) If there is a node in γ labeled with a concept expression e, then there must be exactly one node in γ labeled with $\bar{e}$.

(iv) For each inclusion $e \sqsubseteq f$ in Σ, there is an arc (M,N) in δ, where M and N are the nodes labeled with e and f, respectively.

(v) If there are nodes M and N in γ labeled with ($\geq m\ p$) and ($\geq n\ p$), where p is either P or P^{-} and $m<n$, then there is an arc (N,M) in δ.

(vi) If there is an arc (M,N) in δ, where M and N are the nodes labeled with e and f respectively, then there is an arc (K,L) in δ, where K and L are the nodes labeled with $\bar{f}$ and $\bar{e}$, respectively.

(vii) These are the only nodes and arcs of $g(\Sigma)$. □

Definition 2: The labeled graph $G(\Sigma,\Omega)=(\eta,\varepsilon,\lambda)$ that *represents* Σ and Ω, where λ labels each node with a set of expressions, is defined from $g(\Sigma,\Omega)$ by collapsing each clique of $g(\Sigma,\Omega)$ into a single node labeled with the expressions that previously labeled the nodes in the clique. When Ω is the empty set, we simply write $G(\Sigma)$ and say that the graph *represents* Σ. □

If a node K of $G(\Sigma,\Omega)$ is labeled with an expression e, then $\overline{K}$ denotes the node labeled with $\bar{e}$ (which may be K itself). We say that K and $\overline{K}$ are *dual nodes* of $G(\Sigma,\Omega)$. We use $K\rightarrow M$ to indicate that there is a path in $G(\Sigma,\Omega)$ (or in $g(\Sigma,\Omega)$) from K to M, and $K\nrightarrow M$ to indicate that no such path exists. Also, to simplify the notation, we use $e\rightarrow f$ to denote that there is a path in $G(\Sigma,\Omega)$ (or in $g(\Sigma,\Omega)$) from the node labeled with e to the node labeled with f, and $e\nrightarrow f$ to indicate that no such path exists.

Definition 3: Let $G(\Sigma,\Omega)=(\eta,\varepsilon,\lambda)$ be the labeled graph that represents Σ and Ω. We say that a node K of $G(\Sigma,\Omega)$ is a *⊥-node with level n*, for a non-negative integer n, iff one of the following conditions hold:

(i) K is is a *⊥-node with level 0* iff

a. K is labeled with ⊥, or

b. There are nodes M and N, not necessarily distinct from K, and a non-negated concept expression h such that M and N are respectively labeled with h and $\neg h$, and $K\rightarrow M$ and $K\rightarrow N$.

(ii) K is is a *⊥-node with level n+1* iff

a. There is a ⊥-node M of level n, distinct from K, such that $K\rightarrow M$, and M is the ⊥-node with the smallest level such that $K\rightarrow M$, or

b. K is labeled with a minCardinality constraint of the form ($\geq 1\ P$) (or of the form ($\geq 1\ P^{-}$)) and there is a ⊥-node M of level n, distinct from K, such that M is labeled with ($\geq 1\ P^{-}$) (or with ($\geq 1\ P$)), and M is the ⊥-node with the smallest level labeled with ($\geq 1\ P^{-}$) or ($\geq 1\ P$). □

In view of Case (ii-b), the notion of level is necessary to avoid a circular definition. In Case (i-b), note that, if $K=M=N$, then K is labeled with both h and $\neg h$; other special cases occur when $K=M$, and when $K=N$. Also note that K may be labeled with both $(\geq 1\ P)$ and $(\geq 1\ P^-)$, and yet be a $\bot$-node by virtue of Cases (i) and (ii-a), but not because of Case (ii-b).

Definition 4: Let $G(\Sigma,\Omega)=(\eta,\varepsilon,\lambda)$ be the labeled graph that represents Σ and Ω. Let K be a node of $G(\Sigma,\Omega)$. We say that

(i) K is a *$\bot$-node* iff K is a $\bot$-node with level n, for some non-negative integer n.

(ii) K is a *role $\bot$-node* iff K is labeled with an atomic role P and the node labeled with $(\geq 1\ P)$ is a $\bot$-node.

(iii) K is a *$\top$-node* iff $\overline{K}$ is a $\bot$-node.

(iv) K *satisfies the consistency check* iff K is not a $\bot$-node.

(v) K *satisfies the dual of the consistency check* iff K is not a $\top$-node.

(vi) $G(\Sigma,\Omega)$ *satisfies the consistency check* iff all nodes labeled with an atomic concept or with a minCardinality of the form $(\geq 1\ P)$ satisfy the consistency check. □

We are now ready to prove the major results of the paper. To avoid repetitions, in what follows, let Σ be a set of normalized constraints and Ω be a set of constraint expressions. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω.

Proposition 5:

(i) $G(\Sigma,\Omega)$ is acyclic.

(ii) For any pair of nodes M and N, we have that $M \rightarrow N$ iff $\overline{N} \rightarrow \overline{M}$.

(iii) For any node K of $G(\Sigma,\Omega)$, for any expression e, we have that e labels K iff $\bar{e}$ labels $\overline{K}$.

(iv) For any node K of $G(\Sigma,\Omega)$,

  (a) K is labeled only with $\bot$, or

  (b) K is labeled only with $\top$, or

  (c) K is labeled only with a single atomic role, or

  (d) K is labeled only with non-negated concept expressions, which must be atomic concepts or minCardinality constraints of the form $(\geq m\ p)$, where p is either P or P^- and $m \geq 1$, or

(e) K is labeled only with negated concept expressions, which must be negated atomic concepts or minCardinality constraints of the form $\neg(\geq m\ p)$, where p is either P or P^{-} and $m \geq 1$.

(v) For any pair of nodes M and N of $G(\Sigma,\Omega)$, for any pair of expressions e and f that label M and N, respectively, if $M \rightarrow N$ then $\Sigma \models e \sqsubseteq f$.

(vi) For any node K of $G(\Sigma,\Omega)$, for any pair of expressions e and f that label K, $\Sigma \models e \equiv f$.

(vii) For any node K of $G(\Sigma,\Omega)$, for any expression e that labels K, if K is a $\bot$-node, then $\Sigma \models e \sqsubseteq \bot$.

(viii) For any node K of $G(\Sigma,\Omega)$, for any expression e that labels K, if K is a $\top$-node, then $\Sigma \models \top \sqsubseteq e$.

(ix) For any node K of $G(\Sigma,\Omega)$ labeled with an atomic role P, if K is a role $\bot$-node, then any model $\boldsymbol{s}$ of Σ is such that $\boldsymbol{s}(P) = \varnothing$.

(x) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is a $\bot$-node and K is not labeled with $\bot$. Then, K is labeled only with atomic concepts or minCardinality constraints of the form $(\geq m\ p)$, where p is either P or P^{-} and $m \geq 1$.

(xi) Let L be a node of $G(\Sigma,\Omega)$. Assume that L is a $\top$-node and L is not labeled with $\top$. Then, L is labeled only with negated atomic concepts or negated minCardinality constraints of the form $\neg(\geq m\ p)$, where p is either P or P^{-} and $m \geq 1$.

Proof

(i), (ii), (iii) All three properties follow directly from the definition of $G(\Sigma,\Omega)$.

(iv-a) Let K be a node of $g(\Sigma,\Omega)$. Assume that K is labeled with $\bot$. Since $\bot$ occurs only in constraints of the form $e \sqsubseteq \bot$, $g(\Sigma,\Omega)$ has no arcs leaving K. Therefore, K is the single node of a clique of $g(\Sigma,\Omega)$. Therefore, K is a node of $G(\Sigma,\Omega)$ and K is labeled only with $\bot$**.**

(iv-b) Let K be a node of $g(\Sigma,\Omega)$. Assume that K is labeled with $\top$. By (iii), $\overline{K}$ is labeled with $\bot$. Hence, by (iv-a), $\overline{K}$ is only labeled with $\bot$. Consequently, K is only labeled with $\top$.

(iv-c) Let K be a node of $g(\Sigma,\Omega)$. Assume that K is labeled with an atomic role P. Then, there are no arcs touching K. Therefore, K is the single node of a clique of $g(\Sigma,\Omega)$. Therefore, K is a node of $G(\Sigma,\Omega)$ and K is labeled only with P**.**

(iv-d) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is labeled with a non-negated concept expression e.

Suppose that K is labeled with a negated expression f. By (iv-a), f cannot be $\bot$. Then, f is of the form $\neg D$ or $\neg(\geq m\ p)$. Furthermore, since e and f both label node K of $G(\Sigma,\Omega)$, there must be path $e \rightarrow f$ and $f \rightarrow e$ in $g(\Sigma,\Omega)$.

By construction of $G(\Sigma,\Omega)$, and since $f \rightarrow e$, there is a sequence of expressions $h_0, h_1,\ldots,h_m$ such that $h_0 = f$, $h_m = e$ and (h_{i-1}, h_i) is an arc of $g(\Sigma,\Omega)$.

But, if (a,b) is an arc of $g(\Sigma,\Omega)$ such that a is a negated expression, then b is also a negated expression. Indeed, let (a,b) be an arc of $g(\Sigma,\Omega)$ and assume that a is a negated expression. Then, since (a,b) is an arc of $g(\Sigma,\Omega)$, either: (1) $a \sqsubseteq b$ is in Σ; or (2) $\bar{b} \sqsubseteq \bar{a}$ is in Σ; or (3) a is of the form $\neg(\geq m\ p)$ and b is of the form $\neg(\geq n\ p)$, with $m<n$. But no constraint in Σ has a negated expression (including the bottom concept $\bot$) on the left-hand side. Hence, hypothesis (1) is ruled out, since a is negated by assumption. Likewise, in hypothesis (2), $\bar{b}$ must be positive, that is, b must be a negated expression. In case (3), b is already a negated expression of the form $\neg(\geq n\ p)$. Therefore, b is always a negated expression.

Therefore, by induction, since f is a negated expression by assumption, we may conclude that h_m is a negated expression, that is, e is a negated expression, which contradicts the assumption that e is a non-negated concept expression.

Therefore, we may conclude that f cannot be of the form $\neg D$ or $\neg(\geq m\ p)$. Thus, if K is labeled with a non-negated concept expression, then K is labeled only with non-negated concept expressions. Since such non-negated expressions occur in the constraints of Σ, they must be atomic concepts or minCardinality constraints of the form $(\geq m\ p)$, where p is either P or P^- and $m \geq 1$.

(iv-e) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is labeled with a negated concept expression e.

By (iii), $\bar{K}$ is labeled with $\bar{e}$, which is non-negated. Therefore, by (iv-d), $\bar{K}$ is labeled only with non-negated concept expressions. Hence, by (iii) again, K is labeled only with negated concept expressions, which must be negated atomic concepts or minCardinality constraints of the form $\neg(\geq m\ p)$, where p is either P or P^- and $m \geq 1$.

(v), (vi) First observe that, if there is an arc (K,L) of $g(\Sigma,\Omega)$, with K and L labeled with c and d, then $\Sigma \models c \sqsubseteq d$. Hence, for any pair of nodes M and N of $g(\Sigma,\Omega)$, for any pair of expressions e and f that label M and N, respectively, if there is a path from M to N in $g(\Sigma,\Omega)$ then $\Sigma \models e \sqsubseteq f$, by the transitivity of $\sqsubseteq$. Then, properties (v) and (vi) follow by the construction of $G(\Sigma,\Omega)$.

(vii) Let K be a node of $G(\Sigma,\Omega)$ and e be an expression that labels K. Assume that K is a $\bot$-node. The proof follows by induction on the $\bot$-level of K.

Basis: K has $\bot$-level 0.

Case B.1: K is labeled with $\bot$.

Then, by (vi), $\Sigma \models e \equiv \bot$, which trivially implies that $\Sigma \models e \sqsubseteq \bot$.

Case B.2: There are nodes M and N and a non-negated concept expression h such that M and N are respectively labeled with h and $\neg h$, and $K \rightarrow M$ and $K \rightarrow N$.

Then, by (v), $\Sigma \models e \sqsubseteq h$ and $\Sigma \models e \sqsubseteq \neg h$, which implies that $\Sigma \models e \sqsubseteq \bot$.

Induction hypothesis: Assume that the property holds when K has $\bot$-level n.

Induction step: Assume that K has $\bot$-level $n+1$.

Case I.1: There is a $\bot$-*node* M such that $K \rightarrow M$

Then, by the induction hypothesis and by (v), $\Sigma \models e \sqsubseteq \bot$.

Case I.2: K is labeled with a minCardinality constraint of the form ($\geq 1\ P$) or ($\geq 1\ P^-$) and the node M labeled with ($\geq 1\ P^-$) or ($\geq 1\ P$) is a $\bot$-node with level n.

Assume that K is labeled with ($\geq 1\ P$) and M is labeled with ($\geq 1\ P^-$) (the other case is identical). Then, by the induction hypothesis, $\Sigma \models (\geq 1\ P^-) \sqsubseteq \bot$. But this implies that $\Sigma \models (\geq 1\ P) \sqsubseteq \bot$. Since K is labeled with e and ($\geq 1\ P$), by (vi), $\Sigma \models e \equiv (\geq 1\ P)$. Hence, $\Sigma \models e \sqsubseteq \bot$.

(viii) Let K be a node of $G(\Sigma,\Omega)$ and e be an expression that labels K. Assume that K is a $\top$-node. Then, $\overline{K}$ is a $\bot$-node and $\bar{e}$ labels $\overline{K}$, by (iii). Hence, by (vii), $\Sigma \models \bar{e} \sqsubseteq \bot$. Therefore, we have that $\Sigma \models \top \sqsubseteq e$.

(ix) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is labeled with an atomic role P and that K is a role $\bot$-node. Then, the node L labeled with ($\geq 1\ P$) is a $\bot$-node. By (vii), $\Sigma \models (\geq 1\ P) \sqsubseteq \bot$. Hence, any model $\mathbf{s}$ of Σ is such that $\mathbf{s}(P) = \varnothing$.

(x) Let K be a node $G(\Sigma,\Omega)$. Assume that K is a $\bot$-node and K is not labeled with $\bot$.

Note that the assumptions on K rule out Cases (i-a) of definition of $\bot$-node. Therefore, there are three cases to consider.

Case 1: K is labeled with a minCardinality constraint of the form ($\geq 1\ P$) or ($\geq 1\ P^-$), and the node respectively labeled with ($\geq 1\ P^-$) or ($\geq 1\ P$) is not a $\bot$-node.

Then, by (iv-d), K is labeled only with non-negative expressions, which must be atomic concepts or minCardinality constraints of the form ($\geq m\ p$), where p is either P or P^- and $m \geq 1$.

Case 2: There is a node M labeled with $\bot$ such that $K \rightarrow M$.

Since K is a $\bot$-node and there is a node M labeled with $\bot$ such that $K \rightarrow M$, we have that $e \rightarrow \bot$ is a path in $g(\Sigma,\Omega)$. By construction of $g(\Sigma,\Omega)$, and since $e \rightarrow \bot$, there is a sequence of expressions $h_0, h_1, \ldots, h_m$ such that $h_0 = e$, $h_m = \bot$ and (h_{i-1}, h_i) is an arc of $G(\Sigma,\Omega)$. But, if (a,b) is an arc of $G(\Sigma,\Omega)$ such that a is a negated expression, then b is also a negated expression, as already proved in (iv-d). Therefore, by induction, since e is a negated expression by assumption, we may conclude that h_{m-1} is a negated expression. Then, since (h_{m-1}, h_m), that is, $(h_{m-1},\bot)$ is an arc of $g(\Sigma,\Omega)$, either: (1) $h_{m-1} \sqsubseteq \bot$ is in Σ; or (2) $\top \sqsubseteq \overline{h}_{m-1}$ is in Σ. But, this is impossible, again because no constraint in Σ has a negated expression on the left-hand side, and ⊤ cannot occur on a constraint. Therefore, we may conclude that *e* cannot be of the form ¬*D* or ¬(≥*m p*).

Case 3: There are nodes *M* and *N* and a non-negated expression *h* such that *M* and *N* are respectively labeled with *h* and ¬*h*, and *K*→*M* and *K*→*N*.

Since *K*→*M*, by construction of *G(Σ,Ω)*, we have that *e*→*h* is a path in *g(Σ,Ω)*. By construction of *g(Σ,Ω)*, and since *e*→*h*, there is a sequence of expressions h_0, h_1,…,h_m such that h_0 = *e* , h_m = *h* and (h_{i-1}, h_i) is an arc of *g(Σ,Ω)*. Again, if *(a,b)* is an arc of *G(Σ,Ω)* such that *a* is a negated expression, then *b* is also a negated expression. Therefore, by induction, we may conclude that h_m, and hence *h*, is a negated expression, which contradicts the assumption that *h* is non-negated. Hence, *e* cannot be of the form ¬*D* or ¬(≥*m p*).

Thus, we may conclude that, in both cases, *e* must a non-negated concept expression, that is, *e* must be an atomic concept *C* or a minCardinality of the form (≥*m p*), where *p* is either *P* or P^- and *m* ≥ 1.

(xi) Let *L* be a node of *G(Σ,Ω)*. Assume that *L* is a ⊤-node and that *L* is not labeled with ⊤.

Then, by definition of ⊤-node, $\bar{L}$ is a ⊥-node. Furthermore, by (iii), $\bar{L}$ is not labeled with ⊥, recalling that the complement of ⊥ is ⊤, and vice-versa. Thus, by (x), $\bar{L}$ is labeled only with non-negated atomic concepts or minCardinality constraints of the form (≥*m p*), where *p* is either *P* or P^- and *m* ≥ 1. Therefore, by (iii), *L* is labeled only with negated atomic concepts or negated minCardinality constraints of the form ¬(≥*m p*), where *p* is either *P* or P^- and *m* ≥ 1. □

Definition 5: Let *Φ* be a set of distinct *Skolem function symbols for G(Σ,Ω)* as follows:

(i) For any node *N* of *G(Σ,Ω)* labeled with (≥*n P*), associate *n* distinct unary *Skolem function symbols* $f_1[N,P]$,…, $f_n[N,P]$

(ii) For any node *N* of *G(Σ,Ω)* labeled with (≥*n* P^-), associate *n* distinct unary *Skolem function symbols* $g_1[N,P]$,…, $g_n[N,P]$.

(iii) For any node *N* of *G(Σ,Ω)* labeled with an atomic concept or with (≥1 *P*), associate a distinct *Skolem constant c[N]* (a constant is a 0-ary function symbol).

The *Herbrand Universe Δ[Φ]* for *Φ* is the set of first-order terms constructed using the function symbols in *Φ*. The terms in *Δ[Φ]* are called *individuals*. □

Again, to avoid repetitions, let *Φ* be a set of distinct Skolem function symbols for *G(Σ,Ω)* and *Δ[Φ]* be the Herbrand Universe for *Φ*.

Definition 6:

(i) An *instance labeling function for* $G(\Sigma,\Omega)$ and $\Delta[\Phi]$ is a function s' that associates a set of individuals in $\Delta[\Phi]$ to each node of $G(\Sigma,\Omega)$ not labeled with an atomic role, and a set of pairs of individuals in $\Delta[\Phi]$ to each node of $G(\Sigma,\Omega)$ labeled with an atomic role.

(ii) Let N be a node of $G(\Sigma,\Omega)$ labeled with an atomic concept or with (≥ 1 P). Assume that N is not a $\bot$-node. Then, the Skolem constant $c[N]$ is a *seed term* of N, and N is the *seed node* of $c[N]$.

(iii) Let N_P be the node of labeled with the atomic role P. Assume that N_P is not a role $\bot$-node. For each term a, for each node M labeled with ($\geq m$ P), if $a \in s'(M)$ and there is no node K labeled with ($\geq k$ P) such that $m \leq k$ and $a \in s'(K)$, then the pair $(a, f_r[M,P](a))$ is called a *seed pair* of N_P *triggered by* $a \in s'(M)$. We also say that the term $f_r[M,P](a)$ is a *seed term* of the node L labeled with (≥ 1 P^-), and L is called the *seed node* of $f_r[M,P](a)$, for $r \in [2,m]$, if a is of the form $g_i[J,P](b)$, for some node J and some term b, and for $r \in [1,m]$, otherwise.

(iv) Let N_P be the node of labeled with the atomic role P. Assume that N_P is not a role $\bot$-node. For each term b, for each node N labeled with ($\geq n$ P^-), if $b \in s'(N)$ and there is no node K labeled with ($\geq k$ P^-) such that $n \leq k$ and $b \in s'(K)$, then the pair $(g_r[N,P](b),b)$ is called a *seed pair* of N_P *triggered by* $b \in s'(N)$. We also say that the term $g_r[N,P](b)$ is a *seed term* of the node L labeled with (≥ 1 P), and L is called the *seed node* of $g_r[N,P](b)$, for $r \in [2,n]$, if b is of the form $f_i[J,P](a)$, for some node J and some term a, and for $r \in [1,n]$, otherwise.□

Definition 7: A *canonical instance labeling function for* $G(\Sigma,\Omega)$ *and* $\Delta[\Phi]$ is an instance labeling function that satisfies the following restrictions, for each node K of $G(\Sigma,\Omega)$:

(i) Assume that K is not labeled with an atomic role, and that K is neither a $\bot$-node nor a $\top$-node. Then, $t \in s'(K)$ iff t is a seed term of a node J and there is a path from J to K (nodes J and K may be equal, in which case the path is trivial).

(ii) Assume that K is labeled with an atomic role P, and that K is not a role $\bot$-node. Then, $(t,u) \in s'(K)$ iff (t,u) is a seed pair of L triggered by $a \in s'(M)$, where M is a node of $G(\Sigma,\Omega)$ labeled with ($\geq m$ P), or triggered by $b \in s'(N)$, where N is a node of $G(\Sigma,\Omega)$ labeled with ($\geq n$ P^-).

(iii) Assume that K is not labeled with an atomic role, and that K is a $\bot$-node. Then, $s'(K)=\varnothing$.

(iv) Assume that K is not labeled with an atomic role, and that K is a $\top$-node. Then, $s'(K)=\Delta[\Phi]$.

(v) Assume that K is labeled with an atomic role P, and that K is a role $\bot$-node. Then, $s'(K)=\varnothing$. □

Proposition 6: Let s' be canonical instance labeling function for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$. Then

(i) For any pair of nodes M and N of $G(\Sigma,\Omega)$ that are not labeled with an atomic role, if $M \rightarrow N$ then $s'(M) \subseteq s'(N)$.

(ii) For any pair of nodes M and N of $G(\Sigma,\Omega)$ that are not labeled with an atomic role, and that are not a $\bot$-node, $s'(M) \cap s'(N) \neq \varnothing$ iff

a. either M or N is a $\top$-node, or

b. both M and N are not a $\top$-node, and there is a seed node K such that $K \rightarrow M$ and $K \rightarrow N$ (nodes K and M, and K and N may be equal, in which case the respective path is trivial).

(iii) For any node N_P of $G(\Sigma,\Omega)$ labeled with an atomic role P, for any node M of $G(\Sigma,\Omega)$ labeled with ($\geq m\ P$), for any term $t \in s'(M)$, either $s'(N_P)$ contains all seed pairs triggered by $t \in s'(M)$, or there are no seed pairs triggered by $t \in s'(M)$.

(iv) For any node N_P of $G(\Sigma,\Omega)$ labeled with an atomic role P, for any node N of $G(\Sigma,\Omega)$ labeled with ($\geq n\ P^{-}$), for any term $t \in s'(N)$, either $s'(N_P)$ contains all seed pairs triggered by $t \in s'(N)$, or there are no seed pairs triggered by $t \in s'(N)$.

Proof

(i) Let M and N be a pair of nodes of $G(\Sigma,\Omega)$. Assume that M and N are not labeled with an atomic role. Suppose that $M \rightarrow N$. There are 3 cases to consider.

Case 1: M is a $\bot$-node.

Then, by Def. 7(iii), $s'(M)=\varnothing$, which trivially implies $s'(M) \subseteq s'(N)$.

Case 2: N is a $\top$-node.

Then, by Def. 7(iv), $s'(N)=\Delta[\Phi]$, which trivially implies $s'(M) \subseteq s'(N)$.

Case 2: M is a $\top$-node.

By definition of $\top$-node, $\overline{M}$ is a $\bot$-node. By Prop. 5(ii), $\overline{N} \rightarrow \overline{M}$. Then, by definition of $\bot$-node, $\overline{N}$ is also a $\bot$-node. Hence, N is also a $\top$-node. Hence, $s'(M)=\Delta[\Phi]=s'(N)$.

Case 3: M is neither a $\bot$-node nor a $\top$-node, and N is not a $\top$-node.

Since M is not a $\bot$-node and $M \rightarrow N$, by definition of $\bot$-node, N is also not a $\bot$-node. We then have that M and N are not labeled with an atomic role and are not a $\bot$-node or a $\top$-node. Hence, the conditions of Def. 7(i) apply to both M and N.

Let $t \in s'(M)$. By Def. 7(i), t is a seed term of a node J and $J \rightarrow M$. Since $M \rightarrow N$, we then have $J \rightarrow N$. Hence, by Def. 7(i), $t \in s'(N)$. Hence, we may conclude that $s'(M) \subseteq s'(N)$.

(ii) Let M and N be a pair of nodes of $G(\Sigma,\Omega)$. Assume that M and N are not a $\perp$-node. Then, both $s'(M) \neq \emptyset$ and $s'(N) \neq \emptyset$.

Case 1: Either M or N are a $\top$-node.

Then, either $s'(M)=\Delta[\Phi]$ or $s'(N)=\Delta[\Phi]$. Hence, since both $s'(M) \neq \emptyset$ and $s'(N) \neq \emptyset$, we trivially have that $s'(M) \cap s'(N) \neq \emptyset$.

Case 2: Neither M nor N are a $\top$-node.

By assumptions, M and N are not labeled with an atomic role and are neither a $\perp$-node nor a $\top$-node. Hence, the conditions of Def. 7(i) apply to both M and N. Then, $t \in s'(M) \cap s'(N)$ iff t is a seed term of a node J and $J \rightarrow M$ and $J \rightarrow N$.

(iii) This property follows directly from Def. 7(ii) and (v), by observing that there may not be any seed pair triggered by $t \in s'(M)$, where M is labeled with ($\geq m$ P) such that $t \in s'(M)$, if there is a node K labeled with ($\geq k$ P) such that $t \in s'(K)$ and $m < k$.

(iv) Follows as for (iii). □

Definition 8: Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$. The interpretation s for *Σ induced by s'* is defined as follows:

(i) $\Delta[\Phi]$ is the domain of s.

(ii) $s(C)=s'(M)$, for each atomic concept C, where M is the node of $G(\Sigma,\Omega)$ labeled with C (there is just one such node).

(iii) $s(P)=s'(N)$, for each atomic role P, where N is the node of $G(\Sigma,\Omega)$ labeled with P (again, there is just one such node). □

Lemma 1: Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$. Let s be the interpretation induced by s'. Then, we have:

(i) For each node N of $G(\Sigma,\Omega)$, for each non-negated concept expression e that labels N, $s'(N)=s(e)$.

(ii) For each node N of $G(\Sigma,\Omega)$, for each negated concept expression $\neg e$ that labels N, $s'(N) \subseteq s(\neg e)$.

Proof

Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$. Let s be the interpretation induced by s'.

(i) Let N be a node of $G(\Sigma,\Omega)$.

Let e be a non-negated concept expression that labels N. We have to prove that $s'(N)= s(e)$.

Case 1: N is not a $\bot$-node or a $\top$-node.

By the restrictions on constraints and constraint expressions – and this is important – there are 3 cases to consider:

Case 1.1: e is an atomic concept C.

By Def. 8(ii), $s'(N)= s(C)$.

Case 1.2: e is of the form ($\geq n$ P).

Let N_P be the node labeled with P. Then, N_P is not a role $\bot$-node. Indeed, assume otherwise. Then, the node L labeled with (≥ 1 P) would be a $\bot$-node, by definition of role $\bot$-node. But, by construction of $G(\Sigma,\Omega)$, there would be an arc from N (the node labeled with ($\geq n$ P)) to L. Hence, N would be a $\bot$-node, contradicting the assumption of Case 1.

Then, since N_P is not a role $\bot$-node, Def. 7(ii) applies to $s'(N_P)$.

Recall that N is the node labeled with ($\geq n$ P) and that N is not a $\bot$-node or a $\top$-node. We first prove that

(1) $a \in s'(N)$ implies that $a \in s((\geq n\ P))$

Let $a \in s'(N)$. Let K be the node labeled with ($\geq k$ P) such that $a \in s'(K)$ and k is the largest possible. Since $a \in s'(K)$ and k is the largest possible, there are k pairs in $s'(N_P)$ whose first element is a, by Prop. 6(iii). By Def. 8(iii), $s(P)=s'(N_P)$. Hence, by definition of minCardinality, $a \in s((\geq k\ P))$. But again by definition of minCardinality, $s((\geq k\ P)) \subseteq s((\geq n\ P))$, since $n \leq k$, by the choice of k. Therefore, $a \in s((\geq n\ P))$.

We now prove that

(2) $a \in s((\geq n\ P))$ implies that $a \in s'(N)$

Let $a \in s((\geq n\ P))$. By definition of minCardinality, there must be n distinct pairs $(a,b_1),\dots,(a,b_n)$ in $s(P)$ and, consequently, in $s'(N_P)$, since $s(P)=s'(N_P)$, by Def. 8(iii).

Recall that N_P is not a role $\bot$-node. Then, by Def. 7(ii) and Def. 6(iii), possibly by reordering $b_1,\dots,b_n$, we then have that there are nodes $L_0,L_1,\dots L_v$ such that

(3) (a,b_1) is a seed pair of N_P of the form $(g_{i0}[L_0,P](u),u)$, triggered by $u \in s'(L_0)$, where L_0 is labeled with ($\geq l_0$ P^-), for some $i_0 \in [1,l_0]$

or

(4) (a,b_1) is a seed pair of N_P of the form $(a, f_1[L_1,P](a))$, triggered by $a \in s'(L_1)$, where L_1 is labeled with ($\geq l_1$ P) and

(5) (a,b_j) is a seed pair of N_P of the form $(a, f_{wj}[L_i,P](a))$, triggered by $a \in s'(L_i)$, where L_i is labeled with ($\geq l_i$ P), $j \in [(\sum_{r=1}^{i-1} l_r)+1, \sum_{r=1}^{i} l_r]$, with $w_j \in [1,l_i]$ and $i \in [2,v]$

Furthermore, $l_i \neq l_j$, for $i,j \in [2,v]$, with $i \neq j$, since only one node is labeled with ($\geq l_i$ P). We may therefore assume without loss of generality that $l_1 > l_2 > \ldots > l_v$. But note that we then have that $a \in s'(L_i)$ and $a \in s'(L_j)$ and $l_i > l_j$, for each $i,j \in [1,v]$, with $i<j$. But this contradicts the fact that $(a, f_{wj}[L_j,P](a))$ is a seed pair of N_P triggered by $a \in s'(L_j)$ since, by Def. 6(iii), there could be no node L_i labeled with ($\geq l_i$ P) with $l_i > l_j$ and $a \in s'(L_i)$. This means that in fact there is just one node, L_1, that satisfies (5).

We are now ready to show that $a \in s'(N)$.

Case 1.2.1: $n=1$.

Case 1.2.1.1: a is of the form $g_{i0}[L_0,P](u)$.

Recall that N_P is not a role $\bot$-node. Then, by Def. 6(iv), $g_{i0}[L_0,P](u)$ is a seed term of the node labeled with ($\geq$1 P), which must be N, since $n=1$ and there is just one node labeled with ($\geq$1 P). Therefore, since N is not a $\bot$-node or a $\top$-node, by Def. 7(i), $a \in s'(N)$.

Case 1.2.1.2: a is not of the form $g_{i0}[L_0,P](u)$.

Then, by (4) and assumptions of the case, $a \in s'(L_1)$. Since, L_1 is labeled with ($\geq l_1$ P) and N with ($\geq$1 P), either $n=l_1=1$ and $N=L_1$, or $l_1>n=1$ and (L_1,N) is an arc of $G(\Sigma,\Omega)$, by definition of $G(\Sigma,\Omega)$. Then, $s'(L_1) \subseteq s'(N)$, using Prop. 6(i), for the second alternative. Therefore, $a \in s'(N)$ as desired, since $a \in s'(L_1)$.

Case 1.2.2: $n>1$.

We first show that $n \leq l_1$. First observe that, by (5) and $n>1$, $s'(N_P)$ contains a seed pair $(a, f_{wj}[L_1,P](a))$ triggered by $a \in s'(L_1)$. Then, by Prop. 6(iii), $s'(N_P)$ contains all seed pairs triggered by $a \in s'(L_1)$. In other words, we have that $a \in s((\geq n\ P))$ and $(a,b_1),\ldots,(a,b_n) \in s'(N_P)$ and $(a,b_1),\ldots,(a,b_n)$ are triggered by $a \in s'(L_1)$. Therefore, either $(a,b_1),\ldots,(a,b_n)$ are all pairs triggered by $a \in s'(L_1)$, in which case $n=l_1$, or $(a,b_1),\ldots,(a,b_n), (a,b_{n+1}),\ldots,(a, b_{l_1})$, in which case $n<l_1$. Hence, we have that $n \leq l_1$.

Since L_1 is labeled with ($\geq l_1$ P) and N with ($\geq n$ P), with $n \leq l_1$, either $n=l_1$ and $N=L_1$, or $l_1>n$ and (L_1,N) is an arc of $G(\Sigma,\Omega)$, by definition of $G(\Sigma,\Omega)$. Then, $s'(L_1) \subseteq s'(N)$, using Prop. 6(i), for the second alternative. Therefore, $a \in s'(N)$ as desired, since $a \in s'(L_1)$.

Therefore, we established that (2) holds. Hence, from (1) and (2), $s'(N)=s((\geq n\ P))$, as desired.

Case 1.3: e is of the form ($\geq n$ P^-).

The proof of this case is entirely similar to that of Case 1.2.

Case 2: N is a $\perp$-node.

By Def. 7(iii), we then have $s'(N)=\emptyset$. Let e be a non-negated concept expression that labels N. We show that $s'(N)=s(e)=\emptyset$.

We begin by observing that, by Prop. 5(x), either N is labeled with $\perp$, or N is labeled only with non-negated atomic concepts or minCardinality constraints of the form ($\geq n\ p$), where p is either P or P^- and $1\leq n$.

Then, there are two cases to consider.

Case 2.1: e is a non-negated atomic concept C.

Then, we trivially have, by Def. 8 (ii), that $s(C)=\emptyset$.

Case 2.2: e is a minCardinality constraint of the form ($\geq n\ p$), where p is either P or P^- and $1\leq n$.

We prove that $s((\geq n\ p))=\emptyset$, using an argument similar to that in Case 1.2.

Let N_P be the node labeled with P.

Case 2.2.1: N_P is a role $\perp$-node

Then, by Def. 7(v) and Def. 8(iii), $s(P)=s'(N_P)=\emptyset$. Hence, $s((\geq n\ p))=\emptyset$.

Case 2.2.2: N_P is not a role $\perp$-node.

Then, Def. 7(ii) applies to $s'(N_P)$.

Assume that $s((\geq n\ p))\neq\emptyset$ and let $a\in s((\geq n\ p))$. By definition of minCardinality and since $s(P)=s'(N_P)$, there must be n distinct pairs $(a,b_1),\ldots,(a,b_n)$ in $s'(N_P)$. Using an argument similar to that in Case 1.2, there are nodes L_0 and L_1 such that

(6) (a,b_1) is a seed pair of N_P of the form $(g_{i0}[L_0,P](u),u)$, triggered by $u\in s'(L_0)$, where L_0 is labeled with ($\geq l_0\ P^-$), for some $i_0\in[1,l_0]$

or

(7) (a,b_1) is a seed pair of N_P of the form $(a, f_1[L_1,P](a))$, triggered by $a\in s'(L_1)$, where L_1 is labeled with ($\geq l_1\ P$)

and

(8) (a,b_j) is a seed pair of N_P of the form $(a, f_{wj}[L_1,P](a))$, triggered by $a\in s'(L_1)$, where L_1 is labeled with ($\geq l_1\ P$), with $j\in[2,l_1]$

We are now ready to show that no such $a\in s((\geq n\ p))$ exists. Recall that $n>1$. We first show that $n\leq l_1$. First observe that, by (8) and $n>1$, $s'(N_P)$ contains a seed pair $(a, f_{wj}[L_1,P](a))$ triggered by $a\in s'(L_1)$. Then, by Prop. 6(iii), $s'(N_P)$ contains all seed pairs triggered by $a\in s'(L_1)$. In other words, we have that $a\in s((\geq n\ P))$ and $(a,b_1),\ldots,(a,b_n)\in s'(N_P)$ and $(a,b_1),\ldots,(a,b_n)$ are triggered by $a\in s'(L_1)$. Therefore, either $(a,b_1),\ldots,(a,b_n)$ are all pairs triggered by $a\in s'(L_1)$, in which case $n=l_1$, or $(a,b_1),\ldots,(a,b_n)$, $(a,b_{n+1}),\ldots,(a,b_{l_1})$, in which case $n<l_1$.

Hence, we have that $n \leq l_1$. Since L_1 is labeled with $(\geq l_1\ P)$ and N with $(\geq n\ P)$, with $n \leq l_1$, either $n=l_1$ and $N=L_1$, or $l_1>n$ and (L_1,N) is an arc of $G(\Sigma,\Omega)$, by definition of $G(\Sigma,\Omega)$. Then, $s'(L_1)\subseteq s'(N)$, using Prop. 6(i), for the second alternative. Therefore, $a\in s'(N)$, since $a\in s'(L_1)$. But this is impossible, since $s'(N)=\emptyset$.

Hence, we conclude that $s((\geq n\ p))=\emptyset$.

Therefore, we have that, if N is a $\bot$-node, then $s'(N)=s(e)=\emptyset$, for any non-negated concept expression e that labels N.

Case 3: N is a $\top$-node.

By Def. 7(iv), we then have $s'(N)=\Delta[\Phi]$. Let e be a non-negated expression that labels N. We show that $s'(N)=s(e)=\Delta[\Phi]$.

By Prop 5(xi), N is either labeled only with $\top$, or labeled only with negated expressions. Therefore, e can only be the top concept $\top$. Therefore, trivially, $s(e)=\Delta[\Phi]$.

Therefore, we established in all three cases that Lemma 1(i) holds.

(ii) Let N be a node of $G(\Sigma,\Omega)$.

Let $\neg e$ be a negated expression that labels N. We have to prove that $s'(N) \subseteq s(\neg e)$.

Case 1: N is not a $\bot$-node or a $\top$-node.

Suppose, by contradiction, that there is a term t such that $t\in s'(N)$ and $t\notin s(\neg e)$.

Since $t\notin s(\neg e)$, we have that $t\in s(e)$, by definition. Let M be the node labeled with e. Hence, by Lemma 1(i), $t\in s'(M)$. That is, $t \in s'(M) \cap s'(N)$.

Note that M and N are in fact dual nodes. Therefore, since N is not a $\bot$-node or a $\top$-node, M is also not a $\top$-node or a $\bot$-node, by definition of $\top$-node. Hence, by Prop. 6(ii) and Def. 7(i), since both M and N are not a $\bot$-node or a $\top$-node, there is a seed node K such that $K\rightarrow M$ and $K\rightarrow N$ and $t\in s'(K)$. But this is impossible. Indeed, we would have that $K\rightarrow M$ and $K\rightarrow N$, M is labeled with e, and N is labeled with $\neg e$, which implies that K is a $\bot$-node. Hence, by Def. 7(iii), $s'(K)=\emptyset$, which implies that $t\notin s'(K)$.

Therefore, we established that, for all terms t, if $t\in s'(N)$ then $t\in s(\neg e)$. That is, $s'(N) \subseteq s(\neg e)$, as desired.

Case 2: N is a $\bot$-node.

By Def. 7(iii), we then have $s'(N)=\emptyset$, which trivially implies that $s'(N) \subseteq s(\neg e)$.

Case 3: N is a $\top$-node.

By Def. 7(iv), we then have $s'(N)=\Delta[\Phi]$. We show that $s(\neg e)=\Delta[\Phi]$. Let $\overline{N}$ be the dual node of N. Since N is a $\top$-node, we have that $\overline{N}$ is a $\bot$-node. Furthermore, since $\neg e$ labels N, e

labels $\overline{N}$. Since e is a positive expression, by Lemma 1(i), $s'(\overline{N})=s(e)=\emptyset$. Thus, $s(\neg e)=\Delta[\Phi]$, which trivially implies that $s'(N) \subseteq s(\neg e)$.

Therefore, we established that, in all three cases, Lemma 1(ii) holds. □

Lemma 2: Let s be the interpretation for Σ induced by a canonical instance labeling function for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$. Then, we have

(i) s is a model of Σ.

(ii) Let N be a node of $G(\Sigma,\Omega)$. Let e be an atomic concept or a minCardinality of the form ($\geq 1\ P$) that labels N. Assume that N is not a $\perp$-node. Then $s(e)\neq\emptyset$.

(iii) Let N be a node of $G(\Sigma,\Omega)$. Let P be an atomic role that labels N. Assume that N is not a role $\perp$-node. Then, $s(P)\neq\emptyset$.

Proof

Let Σ be a set of normalized constraints and Ω be a set of constraint expressions. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω. Let Φ be a set of distinct function symbols and $\Delta[\Phi]$ be the Herbrand Universe for Φ. Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$ and s be the interpretation for Σ induced by s'.

(i) We prove that s satisfies all constraints in Σ.

Let $e \sqsubseteq f$ be a constraint in Σ. By the restrictions on the constraints in Σ, e must be non-negated and f can be negated or not. Therefore, there are two cases to consider.

Case 1: e and f are both non-negated.

Then, by Lemma 1(i), $s'(M)=s(e)$ and $s'(N)=s(f)$, where M and N are the nodes labeled with e and f, respectively. If $M=N$, then we trivially have that $s'(M)=s'(N)$. So assume that $M\neq N$. Since $e \sqsubseteq f$ is in Σ and $M\neq N$, there must be an arc (M,N) of $G(\Sigma,\Omega)$. By Prop. 6(i), we then have $s'(M) \subseteq s'(N)$. Hence, $s(e) = s'(M) \subseteq s'(N) = s(f)$.

Case 2: e is non-negated and f is negated.

Then, by Lemma 1(i), $s'(M)=s(e)$ and, by Lemma 1(ii), $s'(N) \subseteq s(f)$, where M and N are the nodes labeled with e and f, respectively. Since negated expressions do not occur on the left-hand side of constraints in Σ, e and f cannot label nodes that belong to the same clique in the original graph. Therefore, we have that $M\neq N$. Since $e \sqsubseteq f$ is in Σ and $M\neq N$, there must be an arc (M,N) of $G(\Sigma,\Omega)$. By Prop. 6(i), we then have $s'(M) \subseteq s'(N)$. Hence, $s(e) = s'(M) \subseteq s'(N) \subseteq s(f)$.

Thus, in both cases, $s(e) \subseteq s(f)$. Therefore, for any constraint $e \sqsubseteq f$ in Σ, we have that $s \models e \sqsubseteq f$, which implies that s is a model of Σ.

(ii) Let N be a node of $G(\Sigma,\Omega)$. Let e be an atomic concept or a minCardinality of the form ($\geq 1\ P$) that labels N. Assume that N is not a $\bot$-node. Then, by Lemma 1(i), $s(e)=s'(N)$.

Case 1: N is a $\top$-node.

Then, we trivially have that $s(e)=s'(N)=\Delta[\Phi]\neq\emptyset$.

Case 2: N is not a $\top$-node.

Then, N is neither a $\bot$-node nor a $\top$-node. By Def. 6(ii) and Def. 7(i), the seed term $c[N]$ of N is such that $c[N]\in s(e)$. Hence, trivially, $s(e)\neq\emptyset$.

(iii) Let N be a node of $G(\Sigma,\Omega)$ and P be an atomic role that labels N. Assume that N is not a role $\bot$-node. Then, the node labeled with ($\geq 1\ P$) is not a $\bot$-node. Then, by (ii), $s((\geq 1\ P))\neq\emptyset$. Hence, $s(P)\neq\emptyset$. □

We are now ready to prove the second major result of the paper.

Theorem 2: Let Σ be a set of normalized constraints. Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω. Then, $\Sigma \models e \sqsubseteq f$ iff one of the following conditions holds:

(a) The node labeled with e is a $\bot$-node; or

(b) The node labeled with f is a $\top$-node; or

(c) There is a path in $G(\Sigma,\Omega)$ from the node labeled with e to the node labeled with f.

Proof

Let Σ be a set of normalized constraints. Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω. Observe that, by construction, $G(\Sigma,\Omega)$ has a node labeled with e and a node labeled with f. Let M and N be such nodes, respectively.

(⇐) We show that $\Sigma \models e \sqsubseteq f$. There are three cases to consider:

Case 1: M is a $\bot$-node.

Then, by Prop. 5 (vii), $\Sigma \models e \sqsubseteq \bot$, which trivially implies that $\Sigma \models e \sqsubseteq f$.

Case 2: N is a $\top$-node.

Then, by Prop. 5 (viii), $\Sigma \models \top \sqsubseteq f$, which trivially implies that $\Sigma \models e \sqsubseteq f$.

Case 3: There is a path in $G(\Sigma,\Omega)$ from M to N.

Then, by Prop. 5(v) and (vi), we have that $\Sigma \models e \sqsubseteq f$.

(⇒) We prove that, if the conditions of the theorem do not hold, then $\Sigma \nvDash e \sqsubseteq f$.

Since $e \sqsubseteq f$ is a constraint, we have:

(1) e is either an atomic concept C or minCardinalities of the form of the form $(\geq k\ p)$, where p is either P or P^-, and

(2) f is either the bottom concept $\bot$, an atomic concept C, a negated atomic concept $\neg D$, minCardinality constraints of the form $(\geq k\ p)$, or negated minCardinality constraints of the form $\neg(\geq k\ p)$, where p is either P or P^-

Assume that the conditions of the theorem do not hold, that is:

(3) The node M labeled with e is not a $\bot$-node; and

(4) The node N labeled with f is not a $\top$-node; and

(5) There is no path in $G(\Sigma,\Omega)$ from M to N .

To prove that $\Sigma \nvDash e \sqsubseteq f$, it suffices to exhibit a model r of Σ such that $r \nvDash e \sqsubseteq f$. Recall that $r \nvDash e \sqsubseteq f$ iff there is an individual t such that $t \in r(e)$ and $t \notin r(f)$ or, equivalently, $t \in r(\neg f)$.

Recall that, to simplify the notation, $e \rightarrow f$ denotes that there is a path in $G(\Sigma,\Omega)$ from the node labeled with e to the node labeled with f, and $e \nrightarrow f$ to indicate that no such path exists.

Since $e \sqsubseteq f$ is a constraint, e must be non-negated and f can be negated or not. Hence, there are 2 cases to consider.

Case 1: e and f are both non-negated.

Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and s be the model induced by s'. By Lemma 2, s is a model of Σ. We show that $s \nvDash e \sqsubseteq f$.

Case 1.1: N is a $\bot$-node.

Since N is a $\bot$-node, by Prop. 5(vii), we have that $\Sigma \vDash f \sqsubseteq \bot$, which implies that $s(f)=\emptyset$, since s is a model of Σ.

By (1), e is either an atomic concept C or minCardinalities of the form $(\geq k\ p)$, where p is either P or P^-. By (3), M is not a $\bot$-node. Hence, by Def. 7(i), if M is not a $\top$-node, or by Def. 7(iv), if M is a $\top$-node, and by Def. 8(ii), $s(e)\neq\emptyset$. Hence, we trivially have that $s \nvDash e \sqsubseteq f$.

Case 1.2: N is not a $\bot$-node.

Observe that M and N are neither a $\bot$-node nor a $\top$-node. Indeed, by assumption of the case and by (4), N is neither a $\bot$-node nor a $\top$-node. Now, by (3), M is not a $\bot$-node. Furthermore, since $e \sqsubseteq f$ is a constraint, either M and N are the same node or there is an arc (M,N) in $G(\Sigma,\Omega)$. Therefore, M cannot be a $\top$-node as otherwise N would be a $\top$-node, contradicting (4).

By Lemma 1(i), since e is non-negated by assumption, and Def. 6(ii) and Def. 7(i), since M is neither a $\bot$-node nor a $\top$-node, we have

(6) $s'(M)=s(e)$ and there is a seed term $c[M]\in s'(M)$

By definition of canonical instance labeling function, we have:

(7) For each node K of $G(\Sigma,\Omega)$ that is neither a $\bot$-node nor a $\top$-node or labeled with an atomic role, $c[M]\in s'(K)$ iff there is a path from M to K

By (5), we have $e\nrightarrow f$. Furthermore, N is neither a $\bot$-node nor a $\top$-node. Hence, by (7), we have:

(8) $c[M]\notin s'(N)$

Since f is positive, by Lemma 1(i), $s'(N)=s(f)$. Hence, we have

(9) $c[M]\notin s(f)$

Therefore, by (6) and (9), $s(e)\not\subseteq s(f)$, that is, $s\not\models e\sqsubseteq f$, as desired.

Case 2: e is non-negated and f is negated.

Assume that f is a negated expression of the form $\neg g$, where g is non-negated (if f is $\bot$ then g is $\top$).

Case 2.1: $e\rightarrow g$.

Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and s be the model induced by s'. By Lemma 2, s is a model of Σ. We show that $s\not\models e\sqsubseteq f$.

By Prop. 5(v) and (vi), and since s is a model of Σ, we have that $s\models e\equiv g$, if e and g label the same node, and $s\models e\sqsubseteq g$, otherwise. Hence, we have that $s\not\models e\sqsubseteq\neg g$. Now, since f is $\neg g$, we have $s\not\models e\sqsubseteq f$, as desired.

Case 2.2: $e\nrightarrow g$.

Construct Φ as follows:

(10) Φ is Σ with two new constraints, $H\sqsubseteq e$ and $H\sqsubseteq g$, where H is a new atomic concept

Let r' be a canonical instance labeling function for $G(\Phi,\Omega)$ and r be the model induced by r'. By Lemma 2, r is a model of Φ. We show that $r\not\models e\sqsubseteq f$.

We first observe that

(11) There is no expression h such that $e\rightarrow h$ and $g\rightarrow\neg h$ are paths in $G(\Sigma,\Omega)$

Indeed, by construction of $G(\Sigma,\Omega)$, $g\rightarrow\neg h$ iff $h\rightarrow\neg g$. But $e\rightarrow h$ and $h\rightarrow\neg g$ implies $e\rightarrow\neg g$, contradicting (5), since f is $\neg g$. Hence, (11) follows.

We now prove that

(12) There is no non-negated expression h such that $H \rightarrow h$ and $H \rightarrow \neg h$ are paths in $G(\Phi,\Omega)$

Assume otherwise. Let h be a non-negated expression such that $H \rightarrow h$ and $H \rightarrow \neg h$ are paths in $G(\Phi,\Omega)$.

Case 2.2.1: $H \rightarrow e \rightarrow h$ and $H \rightarrow g \rightarrow \neg h$ are paths in $G(\Phi,\Omega)$.

Then, $e \rightarrow h$ and $g \rightarrow \neg h$ must be paths in $G(\Sigma,\Omega)$, which contradicts (11).

Case 2.2.2: $H \rightarrow e \rightarrow \neg h$ and $H \rightarrow g \rightarrow h$ are paths in $G(\Phi,\Omega)$.

Then, $e \rightarrow \neg h$ and $g \rightarrow h$ must be paths in $G(\Sigma,\Omega)$. But, since $g \rightarrow h$ iff $\neg h \rightarrow \neg g$, we have $e \rightarrow \neg h \rightarrow \neg g$ is a path in $G(\Sigma,\Omega)$, which contradicts (5), recalling that f is $\neg g$.

Case 2.2.3: $H \rightarrow e \rightarrow h$ and $H \rightarrow e \rightarrow \neg h$ are paths in $G(\Phi,\Omega)$.

Then, $e \rightarrow h$ and $e \rightarrow \neg h$ must be paths in $G(\Sigma,\Omega)$, which contradicts (3), by definition of $\bot$-node.

Case 2.2.4: $H \rightarrow g \rightarrow h$ and $H \rightarrow g \rightarrow \neg h$ are paths in $G(\Phi,\Omega)$.

Then, $g \rightarrow h$ and $g \rightarrow \neg h$ must be paths in $G(\Sigma,\Omega)$. Now, observe that, since $\neg g$ is f, that is, f and g are complementary expressions, g labels $\overline{N}$, the dual node of N in $G(\Sigma,\Omega)$. Then, $g \rightarrow h$ and $g \rightarrow \neg h$ implies that $\overline{N}$ is a $\bot$-node of $G(\Sigma,\Omega)$, that is, N is a $\top$-node, which contradicts (4).

Hence, we established (12).

Let K be the node of $G(\Phi,\Omega)$ labeled with H. Note that, by construction of Φ, K is labeled only with H. Then, by (12), K is not a $\bot$-node.

By Lemma 2(i), r is a model of Φ. Furthermore, by Lemma 2(ii) and since K is not a $\bot$-node, we have

(13) $r(H) \neq \emptyset$

Since $H \sqsubseteq e$ and $H \sqsubseteq g$ are in Φ, and since r is a model of Φ, we also have:

(14) $r(H) \subseteq r(e)$ and $r(H) \subseteq r(g)$

Therefore, by (13) and (14) and since $f = \neg g$

(15) $r(e) \cap r(g) \neq \emptyset$ or, equivalently, $r(e) \not\subseteq r(\neg g)$ or, equivalently, $r(e) \not\subseteq r(f)$ or, equivalently, $r \not\models e \sqsubseteq f$

But since $\Sigma \subseteq \Phi$, r is also a model of Σ. Therefore, for Case 2.2, we also exhibited a model r of Σ such that $r \nvDash e \sqsubseteq f$, as desired.

Therefore, in all cases, we exhibited a model of Σ that does not satisfy $e \sqsubseteq f$. □

Corollary 1: Let Σ be a set of normalized constraints. Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω, and $G(\Sigma)$ be the graph that represents Σ. Suppose that $\Sigma \models e \sqsubseteq f$. Then:

(a) Either e labels a node of $G(\Sigma)$ or e is of the form ($\geq k$ P) and there is a node of $G(\Sigma)$ labeled with ($\geq j$ P), where $j<k$.

(b) Either f labels a node of $G(\Sigma)$ or f is of the form $\neg(\geq n$ $P)$ and there is a node of $G(\Sigma)$ labeled with $\neg(\geq m$ $P)$, where $m<n$.

Proof

Let Σ be a set of normalized constraints. Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω, and $G(\Sigma)$ be the graph that represents Σ. Suppose that $\Sigma \models e \sqsubseteq f$.

Then, by Theorem 2, one of the conditions must hold

(1) The node labeled with e is a $\perp$-node; or

(2) The node labeled with f is a $\top$-node; or

(3) There is a path in $G(\Sigma,\Omega)$ from the node labeled with e to the node labeled with f .

Since $e \sqsubseteq f$ is a constraint, e must be an atomic concept of an expression of the form ($\geq k$ P). Let M be the node of $G(\Sigma,\Omega)$ labeled with e, which always exists by construction of $G(\Sigma,\Omega)$. Assume that e does not label a node of $G(\Sigma)$. Then, by construction of $G(\Sigma,\Omega)$, if M is a $\perp$-node of $G(\Sigma,\Omega)$ or there is a path in $G(\Sigma,\Omega)$ starting on M, then there must be an arc (M,K) of $G(\Sigma,\Omega)$, but not of $G(\Sigma)$, since e does not label any node of $G(\Sigma)$). But this is possible only if e is a minCardinality of the form ($\geq k$ P) and there is a node of $G(\Sigma)$ labeled with ($\geq j$ P), where $j<k$.

Likewise, let N be the node of $G(\Sigma,\Omega)$ labeled with f, which always exists by construction of $G(\Sigma,\Omega)$. Assume that f does not label a node of $G(\Sigma)$.Then, by construction of $G(\Sigma,\Omega)$, if N is a $\top$-node of $G(\Sigma,\Omega)$ or there is a path in $G(\Sigma,\Omega)$ ending on N, then there must be an arc (L,N) of $G(\Sigma,\Omega)$, but not of $G(\Sigma)$, since f does not label any node of $G(\Sigma)$). But this is possible only if f is a negated minCardinality of the form $\neg(\geq n$ $P)$ and there is a node of $G(\Sigma)$ labeled with $\neg(\geq m$ $P)$, where $m<n$. □

Let Σ_1 and Σ_2 be two sets of normalized constraints. Let $G(\Sigma_1)$ and $G(\Sigma_2)$ be the graph that represent Σ_1 and Σ_2. Denote their transitive closure by $G^*(\Sigma_1)$ and $G^*(\Sigma_2)$.

Definition 9: The set Γ of constraints that *generates the g.l.b. of* Σ_1 *and* Σ_2 is defined as follows. A constraint $e \sqsubseteq f$ is in Γ iff there are $i, j \in \{1,2\}$, with $i \neq j$, such that one of the following conditions holds

(a) There is a $\perp$-node M of $G(\Sigma_i)$ and a $\perp$-node P of $G(\Sigma_j)$ and
- e is a non-negated constraint expression that labels both M and P
- f is the bottom concept $\perp$

(b) There is a $\perp$-node M of $G(\Sigma_i)$ and an arc (P,Q) of $G^*(\Sigma_i)$ such that P is not a $\perp$-node of $G(\Sigma_j)$ and
- e is a non-negated constraint expression that labels both M and P
- f is a constraint expression that labels Q

(c) There is a $\top$-node N of $G(\Sigma_i)$ and an arc (P,Q) of $G^*(\Sigma_i)$ such that Q is not a $\top$-node of $G(\Sigma_j)$ and
- e is a non-negated constraint expression that labels P
- f is a constraint expression that labels both N and Q

(d) There is a an arc (M,N) of $G^*(\Sigma_i)$ and an arc (P,Q) of $G^*(\Sigma_i)$ such that M, N, P or Q is not a $\perp$-node or a $\top$-node and
- e is a non-negated constraint expression that labels both M and P
- f is a constraint expression that labels both N and Q . □

Note that Γ is indeed a (normalized) set of constraints, in the sense of Sections 3.2 and 4.2. Indeed, by construction, e is always a non-negated constraint expression and f is a constraint expression.

Corollary 2 follows from Theorem 2, Corollary 1 and the definition of Γ, and indicates that Γ is indeed correctly constructed.

Corollary 2: Let Σ_1 and Σ_2 be two sets of normalized constraints. Let Γ be the set of constraints that generates the g.l.b. of Σ_1 and Σ_2. Let $G(\Gamma)$ be the graph that represents Γ. Then, we have

(i) $Th(\Gamma) = \Sigma_1 \vartriangle \Sigma_2$.

(ii) Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. Let $G(\Gamma,\Omega)$ be the graph that represents Γ and Ω. Then, $e \sqsubseteq f$ is in $\Sigma_1 \vartriangle \Sigma_2$ iff one of the conditions holds:

(a) The node of $G(\Gamma,\Omega)$ labeled with e is a $\perp$-node; or

(b) The node of $G(\Gamma,\Omega)$ labeled with f is a $\top$-node; or

(c) There is a path in $G(\Gamma,\Omega)$ from the node labeled with e to the node labeled with f.

Proof

Let Σ_1 and Σ_2 be two sets of normalized constraints. Let Γ be the set of constraints that generates the g.l.b. of Σ_1 and Σ_2.

(i) We first prove that $\Sigma_1 \triangle \Sigma_2 \subseteq Th(\Gamma)$.

Let $e \sqsubseteq f$ be a constraint. Assume that $e \sqsubseteq f$ is in $\Sigma_1 \triangle \Sigma_2$. Then, $e \sqsubseteq f$ is in $Th(\Sigma_1) \cap Th(\Sigma_2)$, that is, $\Sigma_i \models e \sqsubseteq f$, for $i=1,2$. Then, by Theorem 2, one of the conditions expressed in (10), (11) and (12) must hold for $G(\Sigma_i,\Omega)$, for $i=1,2$:

(1) The node of $G(\Sigma_i,\Omega)$ labeled with e is a $\bot$-node; or

(2) The node of $G(\Sigma_i,\Omega)$ labeled with f is a $\top$-node; or

(3) There is a path in $G(\Sigma_i,\Omega)$ from the node labeled with e to the node labeled with f.

Furthermore, by Corollary 1, we have

(4) Either e labels a node of $G(\Sigma_i)$ or e is of the form $(\geq k\ P)$ and there is a node of $G(\Sigma_i)$ labeled with $(\geq j\ P)$, where $j<k$.

(5) Either f labels a node of $G(\Sigma_i)$ or f is of the form $\neg(\geq n\ P)$ and there is a node of $G(\Sigma_i)$ labeled with $\neg(\geq m\ P)$, where $m<n$.

There are 9 cases to consider by combining (1) to (3) for $G(\Sigma_1,\Omega)$ with (1) to (3) for $G(\Sigma_2,\Omega)$. Using (4) and (5), they map to the four cases of Definition 9 as follows:

Case 1: (1) holds for $G(\Sigma_1,\Omega)$ and for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (a).

Case 2: (1) holds for $G(\Sigma_1,\Omega)$ and (2) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (b).

Case 3: (1) holds for $G(\Sigma_1,\Omega)$ and (3) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (b).

Case 4: (2) holds for $G(\Sigma_1,\Omega)$ and (1) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (b).

Case 5: (2) holds for $G(\Sigma_1,\Omega)$ and (2) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (a), using Prop. 5(ii) and (iii).

Case 6: (2) holds for $G(\Sigma_1,\Omega)$ and (3) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (c).

Case 7: (3) holds for $G(\Sigma_1,\Omega)$ and (1) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (b).

Case 8: (3) holds for $G(\Sigma_1,\Omega)$ and (2) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (c).

Case 9: (3) holds for $G(\Sigma_1,\Omega)$ and (3) for $G(\Sigma_2,\Omega)$. This case maps to Def. 9 (d).

In all 9 cases, we have that $e \sqsubseteq f$ is in $Th(\Gamma)$, as desired.

We now prove that $Th(\Gamma) \subseteq \Sigma_1 \triangle \Sigma_2$.

Let $e \sqsubseteq f$ be a constraint. Assume that $e \sqsubseteq f$ is in $Th(\Gamma)$, that is, $\Gamma \models e \sqsubseteq f$. Then, by Theorem 2, one of the conditions expressed in (6), (7) and (8) must hold for $G(\Gamma,\Omega)$:

(6) The node of $G(\Gamma,\Omega)$ labeled with e is a $\bot$-node; or

(7) The node of $G(\Gamma,\Omega)$ labeled with f is a $\top$-node; or

(8) There is a path in $G(\Gamma,\Omega)$ from the node labeled with e to the node labeled with f.

By Definition 9 and Theorem 2, we have that $e \sqsubseteq f$ is in $Th(\Sigma_1) \cap Th(\Sigma_2) = \Sigma_1 \triangle \Sigma_2$, as desired.

(ii) Let $e \sqsubseteq f$ be a constraint and $\Omega = \{e,f\}$. By (i), $e \sqsubseteq f$ is in $\Sigma_1 \triangle \Sigma_2$ iff $e \sqsubseteq f$ is in $Th(\Gamma)$. But, by Theorem 2, $e \sqsubseteq f$ is in $Th(\Gamma)$ iff one of the conditions (a), (b) or (c) holds. □