PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 07/10

Extralite Schemas with Role Hierarchies

Marco Antonio Casanova, Antonio Luz Furtado
Departamento de Informática

José Antonio F. de Macêdo, Vânia Maria Ponte Vidal
Departamento de Computação - UFC

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. 07/10 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena June, 2010

Extralite Schemas with Role Hierarchies*

Marco Antonio Casanova, Antonio Luz Furtado

{casanova, furtado}@inf.puc-rio.br,

José Antonio F. de Macêdo, Vânia Maria Ponte Vidal

Departamento de Computação, Universidade Federal do Ceará

{jose.macedo, vvidal}@lia.ufc.br

Abstract. This paper addresses the problems of testing strict satisfiability and deciding logical implication for extralite schemas with role hierarchies. Using the OWL jargon, extralite schemas support named classes, datatype and object properties, minCardinalities and maxCardinalities, InverseFunctionalProperties, class subset constraints, and class disjointness constraints. Extralite schemas with role hierarchies also support subset and disjointness constraints defined for datatype and object properties. Strict satisfiability imposes the additional restriction that the constraints of a schema must not force classes or datatype or object properties to be always empty, and is therefore more adequate than the traditional notion of satisfiability in the context of database design. The decision procedures outlined in the paper are based on the satisfiability algorithm for Boolean formulas in conjunctive normal form with at most two literals per clause, and explore the structure of a set of constraints, captured as a graph.

Keywords: schema satisfiability, schema redesign, ER schema, Description Logics.

* 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

The question of satisfiability is often taken for granted when designing database schemas, perhaps based on the implicit assumption that real data provides a consistent database state. However, this implicit assumption is unwarranted when the schema results from the integration of several data sources, as in a data warehouse or in a mediation environment. Indeed, when we have to combine semantically heterogeneous data sources, we should expect conflicting data or, equivalently, mutually inconsistent sets of integrity constraints. The same problem also occurs during schema redesign, when changes in some constraints might create conflicts with other parts of the database schema. Naturally, the satisfiability problem is aggravated when the schema integration process has to deal with a large number of source schemas, or when the schema to be redesigned is complex.

In this paper, we show how to test strict satisfiability and decide logical implication for extralite schemas with role hierarchies. Using the OWL jargon, extralite schemas support named classes, datatype and object properties, minCardinalities and maxCardinalities, InverseFunctionalProperties, which capture simple keys, class subset constraints, and class disjointness constraints. Extralite schemas with role hierarchies also support subset and disjointness constraints defined for datatype and object properties (formalized as *atomic roles* in Description Logics). Strict satisfiability imposes the additional restriction that the constraints of a schema must not force classes or datatype or object properties to be always empty, and is more adequate than the traditional notion of satisfiability in the context of database design.

Albeit the syntax and semantics of extralite schemas is that of Description Logics, we depart from the tradition of Description Logics deduction services, which are mostly based on tableaux techniques [Baader and Nutt 2003]. The decision procedures outlined in the paper are based on the satisfiability algorithm for Boolean formulas in conjunctive normal form with at most two literals per clause, described in [Aspvall et al. 1979]. 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 procedures depend on a *constraint graph* that captures the structure of a set of constraints.

The results in this paper extend those presented in [Casanova et al. 2010] for extralite schemas without subset and disjointness constraints defined for datatype and object properties, and have not been presented elsewhere. They are motivated by the problem of computing the constraints of mediated schemas [Casanova et al. 2009], as well as traditional database problems [Lauschner et al. 2009]. For example, the constraint graph introduced in this paper can be used to create a procedure that detects inconsistencies in a set of constraints and suggests alternatives to fix the problem. It also provides the basis for computing the greatest lower bound of two sets of constraints, which is behind a strategy to change the constraints of a mediated schema to accommodate the set of constraints of a new export schema.

There is a vast literature on the formal verification of database schemas and on the formalization of ER and UML diagrams. Space limitations force us to single out just a few references. The problem of modeling conceptual schemas in DL is discussed in [Borgida and Brachman 2003]. DL-Lite was introduced in [Artale et al. 2009; Calvanese et al. 2007; Calvanese et al. 2008] to address schema integration and query answering. Techniques from Propositional Logic to support the specification of Boolean and multivalued dependencies were addressed in [Hartmanna et al. 2009].

When compared with DL-Lite, extralite schemas with role hierarchies treat maxCardinality as a negated form of minCardinality and allow concept and role inclusions and disjunctions in such a way that negated descriptions occur only on the left-hand side of inclusions. While retaining expressiveness, this feature permits using a novel approach to the deductive services for a class of OWL schemas, which is the main contribution of the paper.

The paper is organized as follows. Section 2 reviews DL concepts and introduces the notion of extralite schemas with role hierarchies. Section 3 illustrates the problems that arise with the interaction of concept and role hierarchies. Section 4 shows how to test strict satisfiability and decide logical implication for extralite schemas with role hierarchies. Section 5 contains the conclusions.

2. A CLASS OF DATABASE SCHEMAS

2.1 A Brief Review of Attributive Languages

We adopt a family of *attributive languages* [Baader and Nutt 2003] 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, and the *universal role* and the *bottom role*, also denoted by $\top$ and $\bot$, respectively.

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 is a role description, then the following expressions are role descriptions

p^-	(the *inverse* of p)	$\neg p$	(the *negation* of p)

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 e is a concept description, p is a role description and n is a positive integer, then the following expressions are concept descriptions

$\neg e$	(negation)	$(\leq n\ p)$	(at-most restriction)
$\exists p$	(existential quantification)	$(\geq n\ p)$	(at-least restriction)

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

$\boldsymbol{s}(\top) = \Delta^{\boldsymbol{s}}$	if $\top$ denotes the universal concept
$\boldsymbol{s}(\top) = \Delta^s \times \Delta^s$	if $\top$ denotes the universal role
$\boldsymbol{s}(\bot) = \varnothing$	if $\bot$ denotes the bottom concept or the bottom role
$s(A) \subseteq \Delta^s$	for each atomic concept A of $\mathcal{A}$
$s(P) \subseteq \Delta^s \times \Delta^s$	for each atomic role P of $\mathcal{A}$

The function $\boldsymbol{s}$ is extended to role and concept descriptions of $\mathcal{L}$ as follows (where e is a concept description and p is a role description):

$\boldsymbol{s}(p^-) = \boldsymbol{s}(p)^-$	(the inverse of $\boldsymbol{s}(p)$)
$\boldsymbol{s}(\neg p) = \Delta^s \times \Delta^s - \boldsymbol{s}(p)$	(the complement of $\boldsymbol{s}(p)$ with respect to $\Delta^s \times \Delta^s$)
$\boldsymbol{s}(\neg e) = \Delta^s - \boldsymbol{s}(e)$	(the complement of $\boldsymbol{s}(e)$ with respect to Δ^s)
$\boldsymbol{s}(\exists p) = \{I \in \Delta^s \,/\, (\exists J \in \Delta^s)((I,J) \in \boldsymbol{s}(p))\}$	(the set of individuals that $\boldsymbol{s}(p)$ relates to some individual)
$\boldsymbol{s}(\geq n\ p) = \{I \in \Delta^s \,/\, \lvert\{J \in \Delta^s \,/\, (I,J) \in \boldsymbol{s}(p)\}\rvert \geq n\}$	(the set of individuals that $\boldsymbol{s}(p)$ relates to at least n distinct individuals)
$\boldsymbol{s}(\leq n\ p) = \{I \in \Delta^s \,/\, \lvert\{J \in \Delta^s \,/\, (I,J) \in \boldsymbol{s}(p)\}\rvert \leq n\}$	(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 both u and v are concept descriptions or both u and v are role descriptions of $\mathcal{L}$. We also say that $u \sqsubseteq v$ is a *concept inclusion* iff both u and v are concept descriptions, and that $u \sqsubseteq v$ is a *role inclusion* iff both u and v are role descriptions; and likewise for the other types of formulas.

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 $\boldsymbol{s}(u) \cap \boldsymbol{s}(v) = \emptyset$, and $\boldsymbol{s}$ *satisfies* $u \equiv v$ iff $\boldsymbol{s}(u) = \boldsymbol{s}(v)$. A formula σ is a *tautology* iff any interpretation satisfies σ. Two formulas are *tautologically equivalent* iff any interpretation $\boldsymbol{s}$ that satisfies one formula also satisfies the other.

Given a set of formulas Σ, we say that an interpretation $\boldsymbol{s}$ is a *model* of Σ iff $\boldsymbol{s}$ satisfies all formulas in Σ, denoted $\boldsymbol{s} \models \Sigma$. We say that Σ is *satisfiable* iff there is a model of Σ. However, this notion of satisfiability is not entirely adequate in the context of database design since it allows the constraints of a schema to force atomic concepts or atomic roles to be always empty. Hence, we define that an interpretation $\boldsymbol{s}$ is a *strict model* of Σ iff $\boldsymbol{s}$ satisfies all formulas in Σ and $s(C) \neq \emptyset$, for each atomic concept C, and $s(P) \neq \emptyset$, for each atomic role P; we say that Σ is *strictly satisfiable* iff there is a strict model for Σ. In addition, we say that Σ *logically implies* a formula σ, denoted $\Sigma \models \sigma$, iff any model of Σ satisfies σ.

The inference rules below capture some of the interactions between concept inclusions and disjunctions, on one hand, and role inclusions and disjunctions, on the other hand (where P and Q are atomic roles):

Inclusion-transfer rules $\dfrac{P \sqsubseteq Q}{(\geq n\ P) \sqsubseteq (\geq n\ Q)}$ $\qquad \dfrac{P \sqsubseteq Q}{(\geq n\ P^-) \sqsubseteq (\geq n\ Q^-)}$

Disjunction-transfer rules $\dfrac{(\geq 1\ P) \mid (\geq 1\ Q)}{P \mid Q}$ $\qquad \dfrac{(\geq 1\ P^-) \mid (\geq 1\ Q^-)}{P \mid Q}$

We do not claim that these rules form a complete set of rules. In fact, they just help formulating the definitions in Section 4.1. We also note that concept inclusions do not induce role inclusions, and role disjunctions do not induce concept disjunctions. That is, the following rules are unsound:

(Unsound rules) $\dfrac{(\geq 1\ P) \sqsubseteq (\geq 1\ Q) \quad (\geq 1\ P^-) \sqsubseteq (\geq 1\ Q^-)}{P \sqsubseteq Q}$ $\qquad \dfrac{P \mid Q}{(\geq 1\ P) \mid (\geq 1\ Q)}$ $\qquad \dfrac{P \mid Q}{(\geq 1\ P^-) \mid (\geq 1\ Q^-)}$

We will retake the discussion about the interaction of concept inclusions and disjunctions and role inclusions and disjunctions in the examples of Section 3 and in the formal development of Section 4.

2.2 Extralite Schemas with Role Hierarchies

An *extralite schema with role hierarchies* is a pair $S=(\mathcal{A},\Sigma)$ such that

- $\mathcal{A}$ is an alphabet, called the *vocabulary* of S
- Σ is a set of formulas, called the *constraints* of S, which must be of one the forms (where C and D denote atomic concepts, P and Q denote atomic roles, p denotes P or its inverse P^-, and k is a positive integer):

Domain Constraint:	$\exists P \sqsubseteq C$	(the domain of P is a subset of C)
Range Constraint:	$\exists P^- \sqsubseteq C$	(the range of P is a subset of C)
minCardinality constraint:	$C \sqsubseteq (\geq k\ p)$	(p maps each individual in C to at least k distinct individuals)
maxCardinality constraint:	$C \sqsubseteq (\leq k\ p)$	(p maps each individual in C to at most k distinct individuals)
Concept Subset Constraint:	$C \sqsubseteq D$	(C is a subset of D)
Concept Disjointness Constraint:	$C \mid D$	(C and D are disjoint concepts)
Role Subset Constraint:	$P \sqsubseteq Q$	(P is a subset of Q)
Role Disjointness Constraint:	$P \mid Q$	(P and Q are disjoint roles)

We loosely refer to the concept subset and disjointness constraints of S as the *concept hierarchy* of S, and to the role subset and disjointness constraints of S as the *role hierarchy* of S.

The following are examples of inclusions that are not acceptable constraints:

$\neg C \sqsubseteq D$	(negated atomic concept on the left-hand side of the concept inclusion)
$\neg C \mid D$	(negated atomic concept on the left-hand side of the concept disjunction)
$\neg C \sqsubseteq (\leq k\ P^-)$	(negated atomic concept on the left-hand side of the concept inclusion)
$\neg P \sqsubseteq Q$	(negated atomic role on the left-hand side of the role inclusion)
$P^- \sqsubseteq Q$	(inverse atomic role on the left-hand side of the role inclusion)
$C \sqsubseteq (\leq k\ \neg P)$	(negated atomic role on the at-most restriction)

We *normalize* a set of constraints by rewriting:

$\exists P \sqsubseteq C$	as	$(\geq 1\ P) \sqsubseteq C$	
$\exists P^- \sqsubseteq C$	as	$(\geq 1\ P^-) \sqsubseteq C$	
$C \sqsubseteq (\leq k\ P)$	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, $D \sqsubseteq \neg C$)
$P \mid Q$	as	$P \sqsubseteq \neg Q$	(or, equivalently, $Q \sqsubseteq \neg P$)

The formula on the right-hand column is called the *normal form* of the formula on the left-hand column. Observe that: a formula and its normal form are tautologically equivalent; the normal forms avoid the use of existential quantification and at-most restrictions; negated descriptions occur only on the right-hand side of the normal forms; inverse roles do not occur in role subset or role disjoint constraints.

Finally, we say that the concept descriptions $\exists P$ and ($\geq 1\ P$) are *descriptions of the domain* of *P*, and that the concept descriptions $\exists P^{-}$ and ($\geq 1\ P^{-}$) are *descriptions of the range* of *P*.

3. EXAMPLES

We first introduce examples of concrete, albeit simple extralite schemas with role hierarchies to illustrate the definitions in Section 2. Then, we present more abstract examples that bring forward some of the interactions between concept and role hierarchies.

Example 1: Figure 1(a) shows the ER diagram of the PhoneCompany schema. Fig. 1(b) formalizes the constraints: the first column shows the domain and range constraints; the second column depicts the cardinality constraints; and the third column contains the subset and disjointness constraints.

The first column of Figure 1(b) indicates that:

- `number` is an atomic role modeling an attribute of `Phone` with range `String`
- `duration` is an atomic role modeling an attribute of `Call` with range `String`
- `location` is an atomic role modeling an attribute of `Call` with range `String`
- `placedBy` is an atomic role modeling a binary relationship from `Call` to `Phone`
- `mobPlacedBy` is an atomic role modeling a binary relationship from `MobileCall` to `Mobile-Phone`

The second column of Figure 1(b) shows the cardinalities of the PhoneCompany schema:

- `number` has maxCardinality and minCardinality both equal to 1 w.r.t. `Phone`
- `duration` has maxCardinality and minCardinality both equal to 1 w.r.t. `Call`
- `location` has maxCardinality and minCardinality both equal to 1 w.r.t. `MobileCall`
- `placedBy` has maxCardinality and minCardinality both equal to 1 w.r.t. `Call`
- (`placedBy`$^{-}$ has unbounded maxCardinality and minCardinality equal to 0 w.r.t. `Phone`, which need not be explicitly declared)
- `mobPlacedBy` has maxCardinality and minCardinality both equal to 1 w.r.t. `MobileCall`
- (`mobPlacedBy`$^{-}$ has unbounded maxCardinality and minCardinality equal to 0 w.r.t. `Mobile-Phone`, which need not be explicitly declared)

The third column of Figure 1(b) indicates that

- `MobilePhone` and `FixedPhone` are subsets of `Phone`
- `MobilePhone` and `FixedPhone` are disjoint atomic concepts
- `MobileCall` is a subset of `Call`
- `mobPlacedBy` is a subset of `placedBy`

Note that the constraints saying that `MobilePhone` is a subset of `Phone` and that `MobileCall` is a subset of `Call` do not imply that `mobPlacedBy` is a subset of `placedBy`. In general, concept inclusions do not imply role inclusions, as already discussed at the end of Section 2.1 (see also Example 3(a)). □

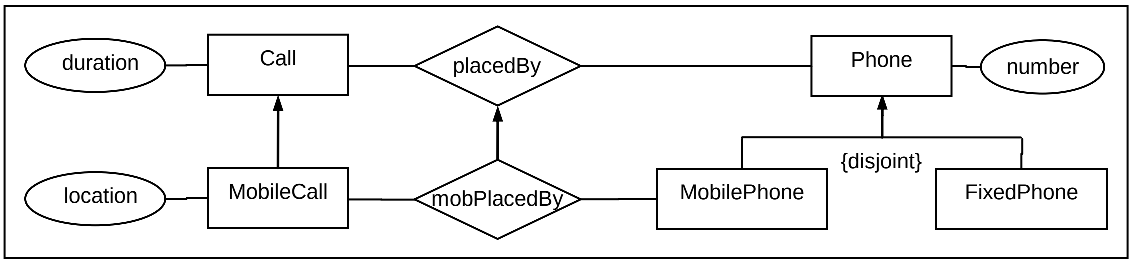

Fig. 1(a). ER diagram of the PhoneCompany schema (without cardinalities).

$\exists\,\mathtt{number} \sqsubseteq \mathtt{Phone}$	$\mathtt{Phone} \sqsubseteq (\leq 1\ \mathtt{number})$	$\mathtt{FixedPhone} \sqsubseteq \mathtt{Phone}$
$\exists\,\mathtt{number}^{-} \sqsubseteq \mathtt{String}$	$\mathtt{Phone} \sqsubseteq (\geq 1\ \mathtt{number})$	$\mathtt{MobilePhone} \sqsubseteq \mathtt{Phone}$
$\exists\,\mathtt{duration} \sqsubseteq \mathtt{Call}$	$\mathtt{Call} \sqsubseteq (\leq 1\ \mathtt{duration})$	$\mathtt{MobilePhone} \mid \mathtt{FixedPhone}$
$\exists\,\mathtt{duration}^{-} \sqsubseteq \mathtt{String}$	$\mathtt{Call} \sqsubseteq (\geq 1\ \mathtt{duration})$	$\mathtt{MobileCall} \sqsubseteq \mathtt{Call}$
$\exists\,\mathtt{location} \sqsubseteq \mathtt{MobileCall}$	$\mathtt{MobileCall} \sqsubseteq (\leq 1\ \mathtt{location})$	$\mathtt{mobPlacedBy} \sqsubseteq \mathtt{placedBy}$
$\exists\,\mathtt{location}^{-} \sqsubseteq \mathtt{String}$	$\mathtt{MobileCall} \sqsubseteq (\geq 1\ \mathtt{location})$	
$\exists\,\mathtt{placedBy} \sqsubseteq \mathtt{Call}$	$\mathtt{Call} \sqsubseteq (\leq 1\ \mathtt{placedBy})$	
$\exists\,\mathtt{placedBy}^{-} \sqsubseteq \mathtt{Phone}$	$\mathtt{Call} \sqsubseteq (\geq 1\ \mathtt{placedBy})$	
$\exists\,\mathtt{mobPlacedBy} \sqsubseteq \mathtt{MobileCall}$	$\mathtt{MobileCall} \sqsubseteq (\leq 1\ \mathtt{mobPlacedBy})$	
$\exists\,\mathtt{mobPlacedBy}^{-} \sqsubseteq \mathtt{MobilePhone}$	$\mathtt{MobileCall} \sqsubseteq (\geq 1\ \mathtt{mobPlacedBy})$	

Fig. 1(b). Formal definition of the constraints of the PhoneCompany schema.

Example 2: Figure 2(a) shows the ER diagram of the PhoneCompany2 schema, and Figure 2(b) formalizes the constraints, following the same organization as that in Figure 1(b). Note that:

- `MobilePhone` and `Phone` are disjoint atomic concepts
- `MobileCall` and `Call` are disjoint atomic concepts
- `PlacedBy` is an atomic role modeling a binary relationship from `Call` to `Phone`
- `mobPlacedBy` is an atomic role modeling a binary relationship from `MobileCall` to `Mobile-Phone`
- the constraints of the schema imply that `PlacedBy` and `mobPlacedBy` are disjoint roles, by the disjunction-transfer rule introduced at the end of Section 2.1 (see also Example 3(b)). □

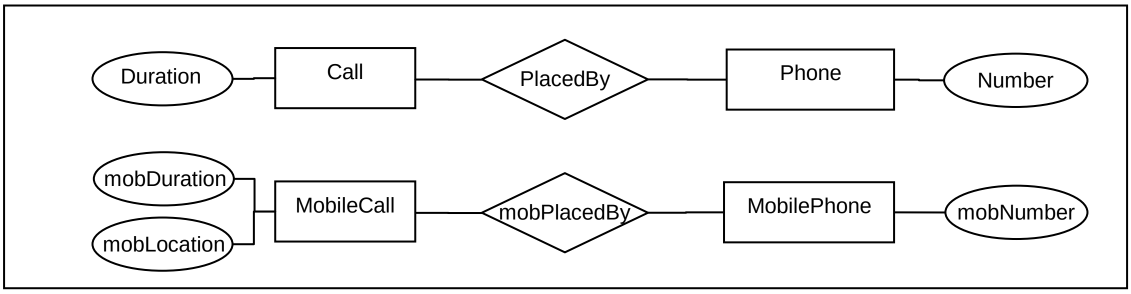

Fig. 2(a). ER diagram of the PhoneCompany2 schema (without cardinalities and disjunctions).

$\exists$ Number $\sqsubseteq$ Phone $\exists$ Number$^-$ $\sqsubseteq$ String $\exists$ Duration $\sqsubseteq$ Call $\exists$ Duration$^-$ $\sqsubseteq$ String $\exists$ placedBy $\sqsubseteq$ Call $\exists$ placedBy$^-$ $\sqsubseteq$ Phone $\exists$ mobDuration $\sqsubseteq$ MobileCall $\exists$ mobDuration$^-$ $\sqsubseteq$ String $\exists$ mobLocation $\sqsubseteq$ MobileCall $\exists$ mobLocation$^-$ $\sqsubseteq$ String $\exists$ mobPlacedBy $\sqsubseteq$ MobileCall $\exists$ mobPlacedBy$^-$ $\sqsubseteq$ MobilePhone	Phone $\sqsubseteq$ ($\leq$ 1 number) Phone $\sqsubseteq$ ($\geq$ 1 number) Call $\sqsubseteq$ ($\leq$ 1 Duration) Call $\sqsubseteq$ ($\geq$ 1 Duration) Call $\sqsubseteq$ ($\leq$ 1 placedBy) Call $\sqsubseteq$ ($\geq$ 1 placedBy) MobileCall $\sqsubseteq$ ($\leq$ 1 mobDuration) MobileCall $\sqsubseteq$ ($\geq$ 1 mobDuration) MobileCall $\sqsubseteq$ ($\leq$ 1 mobLocation) MobileCall $\sqsubseteq$ ($\geq$ 1 mobLocation) MobileCall $\sqsubseteq$ ($\leq$ 1 mobPlacedBy) MobileCall $\sqsubseteq$ ($\geq$ 1 mobPlacedBy)	MobilePhone | Phone MobileCall | Call

Fig. 2(b). Formal definition of the constraints of the PhoneCompany2 schema.

Example 3: This example illustrates how the interaction between the concept and the role hierarchies may lead to unanticipated consequences. Let *A* and *B* be atomic concepts and *P* and *Q* be atomic roles.

(a) Let Φ be the following set of constraints:

(1) $A \mid B$ (*A* and *B* are disjoint concepts)
(2) $A \sqsubseteq \exists P$ (*A* is a subset of the description of the domain of *P*)
(3) $\exists Q \sqsubseteq B$ (the description of the domain of *Q* is a subset of *B*)
(4) $P \sqsubseteq Q$ (*P* is a subset of *Q*)

Then, we have:

(5) $\exists P \sqsubseteq \exists Q$ by (4) and definition of existential quantification
(6) $A \sqsubseteq B$ by (2), (5), (3)

Hence, any model ***s*** of Φ is such that $\boldsymbol{s}(A)=\emptyset$, in view of (1) and (6). Therefore, Φ has no strict model.

(b) Let Σ be the following set of constraints:

(7) $A \mid B$ (*A* and *B* are disjoint)
(8) $\exists P \sqsubseteq A$ (the description of the domain of *P* is a subset of *A*)
(9) $\exists Q \sqsubseteq B$ (the description of the domain of *Q* is a subset of *B*)
(10) $P \sqsubseteq Q$ (*P* is a subset of *Q*)

Then, we have:

(11) $\exists P \mid \exists Q$ by (7), (8), (9)
(12) $P \mid Q$ by (11) and definition of existential quantification

Hence, any model ***s*** of Σ is such that $\boldsymbol{s}(P)=\emptyset$, in view of (10) and (12). Thus, Σ has no strict model. □

4. TESTING EXTRALITE SCHEMAS WITH ROLE HIERARCHIES FOR STRICT SATISFIABILITY

4.1 The General Case

Let Σ be a finite set of normalized constraints and Ω be a finite set of constraint expressions, that is, expression that may occur on the right- or left-hand sides of a normalized constraint. The alphabet is understood as the (finite) set of atomic concepts and roles that occur in Σ and Ω.

We say that the *complement* of a non-negated description c is $\neg c$, and vice-versa. We denote the complement of a description d by $\overline{d}$. Recall that Σ logically implies $e \sqsubseteq \bot$ iff any model of Σ must assign an empty set to the description e, and that Σ logically implies $\top \sqsubseteq e$ iff any model of Σ must assign the set of all individuals to e, if e is a concept description, and the set of all pairs of individuals, if e is a role description. Proposition 1 states properties of descriptions that will be used in the rest of this section.

Proposition 1: Let e, f and g be concept or role descriptions, P and Q be atomic roles, and p be either P or P^-. Then, we have:

(i) $(\geq n\ p) \sqsubseteq (\geq m\ p)$ is a tautology, where $0<m<n$

(ii) $e \sqsubseteq f$ is tautologically equivalent to $\bar{f} \sqsubseteq \bar{e}$

(iii) if Σ logically implies $e \sqsubseteq f$ and $f \sqsubseteq g$, then Σ logically implies $e \sqsubseteq g$

(iv) if Σ logically implies $P \sqsubseteq Q$, then Σ logically implies $(\geq k\ P) \sqsubseteq (\geq k\ Q)$ and $(\geq k\ P^-) \sqsubseteq (\geq k\ Q^-)$ (soundness of the inclusion-transfer rules)

(v) if Σ logically implies $(\geq 1\ P) \sqsubseteq \neg(\geq 1\ Q)$ or $(\geq 1\ P^-) \sqsubseteq \neg(\geq 1\ Q^-)$, then Σ logically implies $P \sqsubseteq \neg Q$ (soundness of the disjunction-transfer rules)

(vi) if Σ logically implies $e \sqsubseteq f$ and $e \sqsubseteq \neg f$, then Σ logically implies $e \sqsubseteq \bot$

(vii) if Σ logically implies $(\geq 1\ P) \sqsubseteq \bot$ or $(\geq 1\ P^-) \sqsubseteq \bot$, then Σ logically implies $P \sqsubseteq \bot$

(viii) if Σ logically implies $P \sqsubseteq \bot$, then Σ logically implies $(\geq k\ P) \sqsubseteq \bot$, $(\geq k\ P^-) \sqsubseteq \bot$, $\top \sqsubseteq (\leq k\ P)$ and $\top \sqsubseteq (\leq k\ P^-)$. □

In the next definitions, we will introduce graphs whose nodes are labeled with expressions or sets of expressions. Then, we use such graphs to create an efficient procedure to test if Σ is strictly satisfiable. Finally, we show how to use the graphs to decide logical implication for Σ. These results extend similar results for concept hierarchies described in [Casanova et al. 2010].

To simplify the definitions, if a node K is labeled with an expression e, then $\overline{K}$ denotes the node labeled with $\bar{e}$. We will also use $K \rightarrow M$ to indicate that there is a path from a node K to a node M, and $K \nrightarrow M$ to indicate that no such path exists; we will use $e \rightarrow f$ to denote that there is a path from a node labeled with e to a node labeled with f, and $e \nrightarrow f$ to indicate that no such path exists.

Definition 1: The labeled graph $g(\Sigma,\Omega)$ that *captures* Σ and Ω, where each node is labeled with an expression, is defined in four stages as follows:

Stage 1:

Initialize $g(\Sigma,\Omega)$ with the following nodes and arcs:

(i) For each atomic concept C, $g(\Sigma,\Omega)$ has exactly one node labeled with C.

(ii) For each atomic role P, $g(\Sigma,\Omega)$ has exactly one node labeled with P, one node labeled with $(\geq 1\ P)$, and one node labeled with $(\geq 1\ P^-)$.

(iii) For each expression e that occurs on the right- or left-hand side of an inclusion in Σ, or that occurs in Ω, other than those in (i) or (ii), $g(\Sigma,\Omega)$ has exactly one node labeled with e.

(iv) For each inclusion $e \sqsubseteq f$ in Σ, $g(\Sigma,\Omega)$ has an arc (M,N), where M and N are the nodes labeled with e and f, respectively.

Stage 2:

Until no new node or arc can be added to $g(\Sigma,\Omega)$,

For each role inclusion $P \sqsubseteq Q$ in Σ,

For each node K,

(i) if K is labeled with ($\geq k\ P$), for some $k>0$, then add a node L labeled with ($\geq k\ Q$), and an arc (K,L), if no such node and arc exists.

(ii) if K is labeled with ($\geq k\ P^-$), for some $k>0$, then add a node L labeled with ($\geq k\ Q^-$), and an arc (K,L), if no such node and arc exists.

(iii) if K is labeled with ($\geq k\ Q$), for some $k>0$, then add a node L labeled with ($\geq k\ P$), and an arc (L,K), if no such node and arc exists.

(iv) if K is labeled with ($\geq k\ Q^-$), for some $k>0$, then add a node L labeled with ($\geq k\ P^-$), and an arc (L,K), if no such node and arc exists.

Stage 3:

Until no new node or arc can be added to $g(\Sigma,\Omega)$,

(i) If $g(\Sigma,\Omega)$ has a node labeled with an expression e, then add a node labeled with $\bar{e}$, if no such node exists.

(ii) If $g(\Sigma,\Omega)$ has a node M labeled with ($\geq m\ p$) and a node N labeled with ($\geq n\ p$), where p is either P or P^- and $0<m<n$, then add an arc (N,M), if no such arc exists.

(iii) If $g(\Sigma,\Omega)$ has an arc (M,N), then add an arc $(\overline{N}, \overline{M})$, if no such arc exists.

Stage 4:

Until no new node or arc can be added to $g(\Sigma,\Omega)$,

for each pair of nodes M and N such that M and N are labeled with ($\geq 1\ P$) and $\neg$($\geq 1\ Q$), respectively, and there is a path from M to N,

add arcs (K,L) and $(\overline{L}, \overline{K})$, where K and L are the nodes labeled with P and $\neg Q$, respectively, if no such arcs exists. □

Note that Stage 1(iv) reflects Proposition 1 (ii), Stage 2 corresponds to Proposition 1 (iv), Stage 3(ii) to Proposition 1(i), Stage 3(iii) to Proposition 1(ii), and Stage 4 to Proposition 1(v).

Definition 2: The labeled graph $G(\Sigma,\Omega)$ that *represents* Σ and Ω, where each node is labeled 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. □

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

(i) K is a $\perp$*-node with level 0* iff there are nodes M and N, not necessarily distinct from K, and a positive expression h such that M and N are respectively labeled with h and $\neg h$, and $K \rightarrow M$ and $K \rightarrow N$.

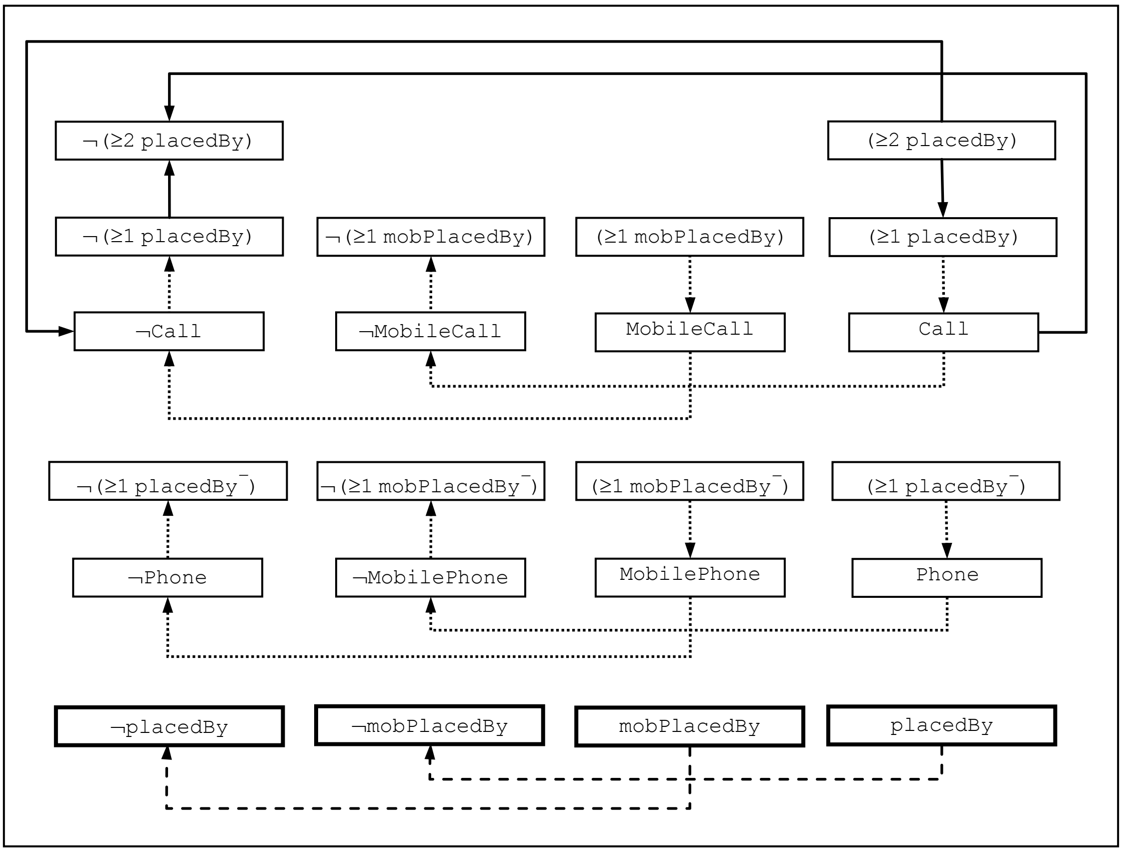

Fig. 3. The graph representing Σ.

(ii) K 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 $(\geq k\ P)$ or of the form $(\geq k\ P^{-})$ and there is a ⊥-node M of level n such that M is labeled with P, or

(c) K is labeled with an atomic role P and there is a ⊥-node M of level n such that M is labeled with a minCardinality constraint of the form $(\geq 1\ P)$ or of the form $(\geq 1\ P^{-})$. □

In case (i), 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 cases (i) and (ii-a) of Definition 3 correspond to Proposition 1(vi), case (ii-b) to Proposition 1(vii), and case (ii-c) to Proposition 1(viii).

We are now ready to state the first result of the paper (see the Appendix for a proof).

Theorem 1: Let Σ be a set of normalized constraints. Let $G(\Sigma)$ be the graph that represents Σ. Then, Σ is strictly satisfiable iff $G(\Sigma)$ has no ⊥-node labeled with an atomic concept or an atomic role. □

Based on Theorem 1, we can then create a procedure that receives as input a set Σ of constraints, normalizes the constraints in Σ, constructs the graph $G(\Sigma)$ that represents Σ, tests if $G(\Sigma)$ has no ⊥-node labeled with an atomic concept or an atomic role, and outputs "YES - Σ is strictly satisfiable", if the test succeeds, and "NO - Σ is not strictly satisfiable", otherwise. Furthermore, we note that the procedure is quadratic on the size of Σ.

Example 4: Let Σ be the following subset of the constraints of the PhoneCompany2 schema, introduced in Example 2 (we do not consider all constraints to reduce the size of the example):

(1)	∃ placedBy ⊑ Call	normalized as: (≥1 placedBy) ⊑ Call
(2)	∃ placedBy⁻ ⊑ Phone	normalized as: (≥1 placedBy⁻) ⊑ Phone
(3)	∃ mobPlacedBy ⊑ MobileCall	normalized as: (≥1 mobPlacedBy) ⊑ MobileCall
(4)	∃ mobPlacedBy⁻ ⊑ MobilePhone	normalized as: (≥1 mobPlacedBy⁻) ⊑ MobilePhone
(5)	Call ⊑ (≤1 placedBy)	normalized as: Call ⊑ ¬(≥2 placedBy)
(6)	MobilePhone | Phone	normalized as: MobilePhone ⊑ ¬Phone
(7)	MobileCall | Call	normalized as: MobileCall ⊑ ¬Call

Figure 3 depicts $G(\Sigma)$, the graph that represents Σ, using the normalized constraints. In special, the dotted arrows highlight the paths that correspond to the conditions of Definition 2(ii), and dashed arrows indicate the arcs that Definition 2(ii) requires to exist, which capture the derived constraint:

(8) mobPlacedBy | placedBy normalized as: mobPlacedBy ⊑ ¬placedBy

Since $G(\Sigma)$ has no ⊥-node, Σ is strictly satisfiable, by Theorem 1. □

Example 5: Let Σ be the following subset of the constraints of the PhoneCompany schema, introduced in Example 1 (again we do not consider all constraints to reduce the size of the example):

(1)	∃ placedBy ⊑ Call	normalized as: (≥1 placedBy) ⊑ Call
(2)	∃ placedBy⁻ ⊑ Phone	normalized as: (≥1 placedBy⁻) ⊑ Phone
(3)	Call ⊑ (≤1 placedBy)	normalized as: Call ⊑ ¬(≥2 placedBy)
(4)	MobileCall ⊑ Call	
(5)	mobPlacedBy ⊑ placedBy	

Let Ψ be defined by adding to Σ a new atomic concept, ConferenceCall, and two new constraints:

(6) ConferenceCall ⊑ Call
(7) ConferenceCall ⊑ (≥2 placedBy)

These new constraints intuitively say that conference calls are calls placed by at least two phones. However, this apparently correct modification applied to the PhoneCompany schema forces ConferenceCall to always have an empty interpretation. Example 6 will also show that (6) is actually redundant.

Indeed, Figure 4 depicts $G(\Psi)$, the graph that represents Ψ, using the normalized constraints. Note that there are paths from the node labeled with ConferenceCall to nodes labeled with Call and ¬Call, as well as to nodes labeled with (≥2 placedBy) and ¬(≥2 placedBy) and nodes labeled with (≥1 placedBy) and ¬(≥1 placedBy) (all arcs of such paths are shown in dashed lines in Figure 4). Hence, the node labeled with ConferenceCall is a ⊥-node of $G(\Psi)$, which implies that Ψ is not strictly satisfiable, by Theorem 1.

Indeed, any interpretation ***s*** that satisfies Ψ is such that the following implications hold

s(ConferenceCall) ⊆ ***s***(Call) and ***s***(ConferenceCall) ⊆ ***s***(¬Call)

which implies that ***s***(ConferenceCall)=∅. □

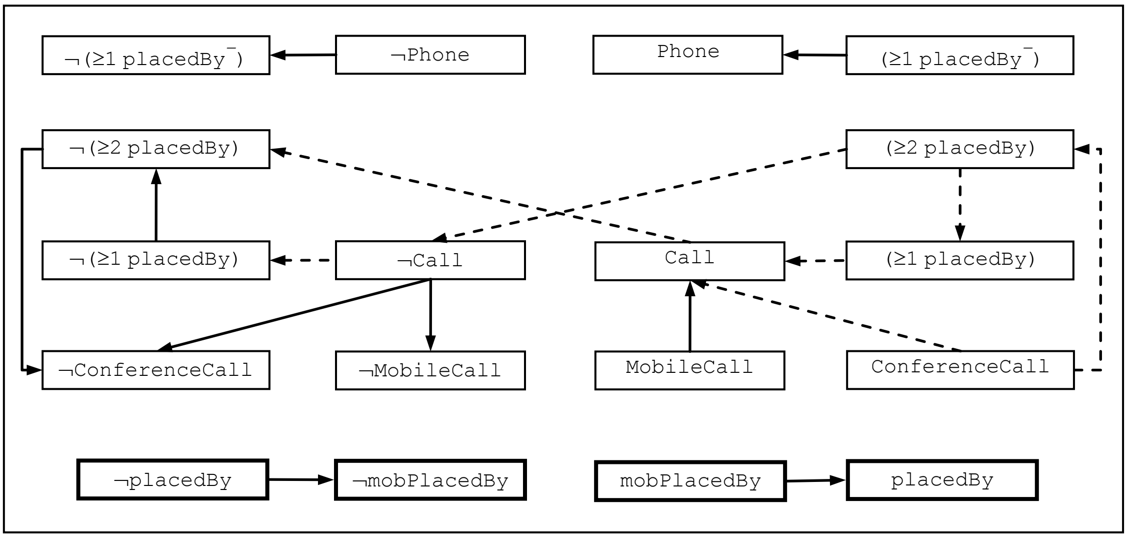

Fig. 4. The graph representing Ψ.

Let $G(\Sigma \cup \{e, f\})$ denote the graph that represents the set of constraints Σ with new nodes labeled with descriptions e and f (that is, $\Omega = \{e, f\}$ in Definitions 1 and 2). From Theorem 1, we can also prove that:

Theorem 2. Let Σ be a set of normalized constraints and σ be a normalized constraint. Assume that σ is of the form $e \sqsubseteq f$. Then, $\Sigma \models \sigma$ iff one of the following conditions holds:

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

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

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

Based on Theorem 2, we can then create a procedure that receives as input a set Σ of constraints and a constraint $e \sqsubseteq f$, and decides whether or not $\Sigma \models e \sqsubseteq f$. The procedure is again quadratic on the size of Σ.

Example 6: This example illustrates the three cases of Theorem 2.

(a) Let Ψ be the set of constraints considered in Example 5. Let $G(\Psi)$ be the graph representing Ψ, shown in Figure 4. Let σ be the constraint `ConferenceCall` $\sqsubseteq$ `(≥1 placedBy⁻)`. Note that σ is of the form $e \sqsubseteq f$, where e = `ConferenceCall` and f = `(≥1 placedBy⁻)`. Then, $G(\Psi \cup \{e, f\})$ is equal to $G(\Psi)$, since $G(\Psi)$ already contains nodes labeled with `ConferenceCall` and with `(≥1 placedBy⁻)`. Recall from Example 5 that the node labeled with `ConferenceCall` is a $\bot$-node of $G(\Psi)$, and hence of $G(\Psi \cup \{e, f\})$. Then, by Theorem 2(i), $\Psi \models$ `ConferenceCall` $\sqsubseteq$ `(≥1 placedBy⁻)`.

(b) Let Ψ and $G(\Psi)$ be as before. Let σ be the constraint `Phone` $\sqsubseteq$ `¬ConferenceCall`. Note that σ is of the form $e \sqsubseteq f$, where e = `Phone` and f = `¬ConferenceCall`. Since the node labeled with `Confe-renceCall` is a $\bot$-node of $G(\Psi \cup \{e, f\})$, the node labeled with `¬ConferenceCall` is $\top$-node of $G(\Psi \cup \{e, f\})$. Hence, by Theorem 2(ii), $\Psi \models$ `Phone` $\sqsubseteq$ `¬ConferenceCall`.

(c) Let Ψ and $G(\Psi)$ be as before. Let σ be the constraint `ConferenceCall` ⊑ `Call`. Note that σ is of the form $e \sqsubseteq f$, where e = `ConferenceCall` and f = `Call`. Since there is a path in $G(\Psi \cup \{e, f\})$ from the node labeled with `ConferenceCall` to the node labeled with `Call`, by Theorem 2(iii), we have that $\Sigma \models$ `ConferenceCall` ⊑ `Call`.

(d) Let Σ be the subset of the constraints of the PhoneCompany2 schema introduced in Example 4. Let σ be the constraint `Call` ⊑ ¬`(≥2 mobPlacedBy)`. Note that σ is of the form $e \sqsubseteq f$, where e = `Call` and f = ¬`(≥2 mobPlacedBy)`. Then, $G(\Sigma \cup \{e, f\})$ is the graph in Figure 5, with two new nodes and two new arcs (highlighted in dashed lines in Figure 5), by virtue of Definition 1(iii) and (v). Since there is a path in $G(\Sigma \cup \{e, f\})$ from the node labeled with `Call` to the node labeled with ¬`(≥2 mobPlacedBy)`, by Theorem 2(iii), we have that $\Sigma \models$ `Call` ⊑ ¬`(≥2 mobPlacedBy)`. □

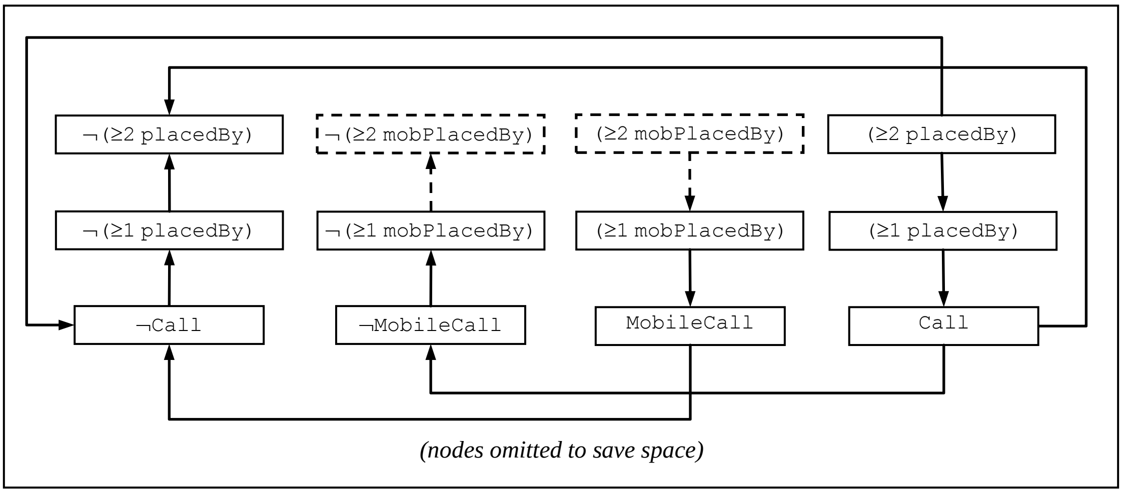

Fig. 5. Partial representation of the graph $G(\Sigma \cup \{e, f\})$.

4.2 Testing Strict Satisfiability for a Class of Extralite Schemas

Let Γ be a set of normalized constraints which contains neither role disjunctions nor maxCardinality constraints. We show how to reduce the problem of testing the strict satisfiability of Γ to testing the strict satisfiability of a set of concept inclusions derived from Γ, using the inclusion-transfer rules introduced at the end of Section 2.1.

A *role chain* in Γ of length n is a sequence of role inclusions of the form $P_0 \sqsubseteq P_1$, $P_1 \sqsubseteq P_2$,…, $P_{n-1} \sqsubseteq P_n$. A role chain $P_0 \sqsubseteq P_1$, $P_1 \sqsubseteq P_2$,…, $P_{n-1} \sqsubseteq P_n$ is a *role cycle* iff $P_n = P_0$. We say that Γ is *role acyclic* iff Γ has no role cycle. Since all atomic roles in a role cycle denote the same binary relation, we may always assume that Γ is role acyclic (by collapsing all atomic roles in a role cycle into a single atomic role).

Assume that Γ is role acyclic and let P be an atomic role. We say that P *has level 0 in* Γ iff P does not occur on the right-hand side of a role inclusion in Γ, and P *has level m in* Γ iff the longest role chain ending on P has length m. The *height* of Γ w.r.t. roles is N iff the longest role chain in Γ has size N.

Definition 5: Let Γ be a set of normalized constraints and assume that Γ is role acyclic. The set $C(\Gamma)$ *capturing* Γ is the set of concept inclusions defined as follows:

(i) All concept inclusions in Γ are in $C(\Gamma)$.
(ii) For each concept description of the form $(\geq k\ Q)$ that occurs on the right- or left-hand side of an inclusion in Γ, for each role chain $P_0 \sqsubseteq P_1$, $P_1 \sqsubseteq P_2$,…, $P_{n-1} \sqsubseteq P_n$ in Γ such that $P_n = Q$, the concept inclusions $(\geq k\ P_i) \sqsubseteq (\geq k\ P_{i+1})$ and $(\geq k\ P_i^-) \sqsubseteq (\geq k\ P_{i+1}^-)$ are in $C(\Gamma)$, for each $i \in [0,n)$.
(iii) These are the only formulas in $C(\Gamma)$. □

Theorem 3. Let Γ be a set of normalized constraints that contains neither role disjunctions nor maxCardinality constraints and that Γ is role acyclic. Then, Γ is strictly satisfiable iff $C(\Gamma)$ is strictly satisfiable.

(see the Appendix for a proof).

The interested reader might verify that the construction of the interpretation ***r*** in the above proof fails for role disjunctions and maxCardinality constraints. Indeed, the construction of ***r****(P)* in (2) guarantees that *r* satisfies role inclusions (Case 3), but not role disjunctions. Furthermore, the construction of ***r****(P)* in (2.2) does not imply that $\boldsymbol{s}((\leq k\ P)) \subseteq \boldsymbol{r}((\leq k\ P))$, which means that step (10) in the argument of Case 6 cannot be adapted to maxCardinality constraints.

In summary, Theorem 3 implies that the problem of testing the strict satisfiability of extralite schemas with role hierarchies, but without role disjunctions and maxCardinality constraints, reduces to the problem of testing the strict satisfiability of extralite schemas without role hierarchies. This reduction then permits ignoring the role hierarchy when testing strict satisfiability. For a theoretic perspective, it allows the direct use of the results in [Casanova et al. 2010] for extralite schemas without role hierarchies, and avoids going through the rather complex proof of Theorem 1, stated in Section 4.1.

5. CONCLUSIONS

We first introduced extralite schemas with role hierarchies that are sufficiently expressive to encode commonly used ER model and UML constructs, including relationship hierarchies. Then, we illustrated what problems arise when the concept and role hierarchies interact. Finally, we showed how to efficiently test strict satisfiability and decide logical implication for extralite schemas with role hierarchies.

The results in Section 4 are novel and cover a technically complex issue, which was overcome basically by the definition of a graph that represents the set of concept and role inclusions and disjunctions of a schema. We stress that the question of strict satisfiability becomes a serious issue when the schema results from the integration of several data sources, or when the schema to be redesigned is complex.

Finally, as future work, we plan to investigate the problem of efficiently testing extralite schemas for finite satisfiability [Rosati 2008].

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

ARTALE, A.; CALVANESE, D.; KONTCHAKOV, R.; ZAKHARYASCHEV, M. 2009. "The DL-Lite family and relations". *J. of Artificial Intelligence Research* 36, pp. 1–69.

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.

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 U. Press, Cambridge, UK.

BORGIDA, A. AND BRACHMAN, R.J. 2003. "Conceptual modeling with 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.

BREITMAN, K.; CASANOVA, M.; TRUSZKOWSKI, W. 2007. *Semantic web: concepts, technologies, and applications*. Springer, London.

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.

CALVANESE, D.; DE GIACOMO, G.; LEMBO, D.; LENZERINI, M.; ROSATI, R. 2007. "Tractable reasoning and efficient query answering in description logics: The DL-Lite family". *J. of Automated Reasoning* 39(3), pp. 385–429.

CASANOVA, M.A.; LAUSCHNER, T.; LEME. L.A.P.P; BREITMAN, K.K.; FURTADO, A.L.; VIDAL, V.M.P. 2009. "A Strategy to Revise the Constraints of the Mediated Schema". In: Proc. 28th Int. Conf. on Conceptual Modeling. Springer, 2009.

CASANOVA, M.A.; LAUSCHNER, T.; LEME. L.A.P.P; BREITMAN, K.K.; FURTADO, A.L.; VIDAL, V.M.P. 2010. "Revising the Constraints of the Mediated Schema". Technical Report MCC06/10, Department of Informatics, PUC-Rio, May 2010.

LAUSCHNER, T.; CASANOVA, M.A.; VIDAL, V.M.P.; MACEDO, J.A. (2009) "Efficient Decision Procedures for Query Containment and Related Problems". In: Anais do XXIV Simpósio Brasileiro de Banco de Dados. 2009.

HARTMANNA, S.; LINKB, S.; TRINHA, T. 2009. "Constraint acquisition for Entity-Relationship models", *Data & Knowledge Engineering*, Vol. 68, Issue 10, Oct. 2009, pp. 1128-1155.

ROSATI, R. 2008. "Finite model reasoning in DL-Lite". In: Proc 5th European Semantic Web Conference, Berlin, Heidelberg: Springer Verlag , June (2008).

APPENDIX - PROOFS OF THE MAIN RESULTS

A.1 Proof of Theorem 1

Let Σ be a finite set of normalized constraints and Ω be a finite set of constraint expressions, that is, expression that may occur on the right- or left-hand sides of a normalized constraint. The alphabet is understood as the (finite) set of atomic concepts and roles that occur in Σ and Ω.

Recall that we say that the *complement* of a non-negated description *c* is $\neg c$, and vice-versa. We denote the complement of a description *d* by $\bar{d}$. Also recall that Σ logically implies $e \sqsubseteq \bot$ iff any model of Σ must assign an empty set to the description *e*, and that Σ logically implies $\top \sqsubseteq e$ iff any model of Σ must assign the set of all individuals to *e*, if *e* is a concept description, and the set of all pairs of individuals, if *e* is a role description. Proposition 1 states properties of descriptions that will be used in the rest of this section.

Proposition 1: Let *e*, *f* and *g* be concept or role descriptions, *P* and *Q* be atomic roles, and *p* be either *P* or P^-. Then, we have:

(i) $(\geq n\ p) \sqsubseteq (\geq m\ p)$ is a tautology, where $0<m<n$

(ii) $e \sqsubseteq f$ is tautologically equivalent to $\bar{f} \sqsubseteq \bar{e}$

(iii) If Σ logically implies $e \sqsubseteq f$ and $f \sqsubseteq g$, then Σ logically implies $e \sqsubseteq g$

(iv) If Σ logically implies $P \sqsubseteq Q$, then Σ logically implies $(\geq k\ P) \sqsubseteq (\geq k\ Q)$ and $(\geq k\ P^-) \sqsubseteq (\geq k\ Q^-)$ (soundness of the inclusion-transfer rules)

(v) If Σ logically implies $(\geq 1\ P) \sqsubseteq \neg(\geq 1\ Q)$ or $(\geq 1\ P^-) \sqsubseteq \neg(\geq 1\ Q^-)$, then Σ logically implies $P \sqsubseteq \neg Q$ (soundness of the disjunction-transfer rules)

(vi) If Σ logically implies $e \sqsubseteq f$ and $e \sqsubseteq \neg f$, then Σ logically implies $e \sqsubseteq \bot$

(vii) If Σ logically implies $(\geq 1\ P) \sqsubseteq \bot$ or $(\geq 1\ P^-) \sqsubseteq \bot$, then Σ logically implies $P \sqsubseteq \bot$

(viii) If Σ logically implies $P \sqsubseteq \bot$, then Σ logically implies $(\geq k\ P) \sqsubseteq \bot$, $(\geq k\ P^-) \sqsubseteq \bot$, $\top \sqsubseteq (\leq k\ P)$ and $\top \sqsubseteq (\leq k\ P^-)$. □

Proof

(The proof follows directly from the definitions in Section 2).

In the next definitions, we will introduce graphs whose nodes are labeled with expressions or sets of expressions. To simplify the definition, if a node *K* is labeled with an expression *e*, then $\bar{K}$ denotes the node labeled with $\bar{e}$. We will also use $K \rightarrow M$ to indicate that there is a path from a node *K* to a node *M*, and $K \nrightarrow M$ to indicate that no such path exists; we will use $e \rightarrow f$ to denote that there is a path from a node labeled with *e* to a node labeled with *f*, and $e \nrightarrow f$ to indicate that no such path exists.

Definition 1: The labeled graph $g(\Sigma,\Omega)$ that *captures* Σ and Ω, where each node is labeled with an expression, is defined in four stages as follows:

Stage 1:

Initialize $g(\Sigma,\Omega)$ with the following nodes and arcs:

(i) For each atomic concept C, $g(\Sigma,\Omega)$ has exactly one node labeled with C.

(ii) For each atomic role P, $g(\Sigma,\Omega)$ has exactly one node labeled with P, one node labeled with ($\geq 1\ P$), and one node labeled with ($\geq 1\ P^{-}$).

(iii) For each expression e that occurs on the right- or left-hand side of an inclusion in Σ, or that occurs in Ω, other than those in (i) or (ii), $g(\Sigma,\Omega)$ has exactly one node labeled with e.

(iv) For each inclusion $e \sqsubseteq f$ in Σ, $g(\Sigma,\Omega)$ has an arc (M,N),where M and N are the nodes labeled with e and f, respectively.

Stage 2:

Until no new node or arc can be added to $g(\Sigma,\Omega)$,

For each role inclusion $P \sqsubseteq Q$ in Σ,

For each node K,

(v) if K is labeled with ($\geq k\ P$), for some $k>0$, then add a node L labeled with ($\geq k\ Q$), and an arc (K,L), if no such node and arc exists.

(vi) if K is labeled with ($\geq k\ P^{-}$), for some $k>0$, then add a node L labeled with ($\geq k\ Q^{-}$), and an arc (K,L), if no such node and arc exists.

(vii) if K is labeled with ($\geq k\ Q$), for some $k>0$, then add a node L labeled with ($\geq k\ P$), and an arc (L,K), if no such node and arc exists.

(viii) if K is labeled with ($\geq k\ Q^{-}$), for some $k>0$, then add a node L labeled with ($\geq k\ P^{-}$), and an arc (L,K), if no such node and arc exists.

Stage 3:

Until no new node or arc can be added to $g(\Sigma,\Omega)$,

(iv) If $g(\Sigma,\Omega)$ has a node labeled with an expression e, then add a node labeled with $\bar{e}$, if no such node exists.

(v) If $g(\Sigma,\Omega)$ has a node M labeled with ($\geq m\ p$) and a node N labeled with ($\geq n\ p$), where p is either P or P^{-} and $m<n$, then add an arc (N,M)**,** if no such arc exists.

(vi) If $g(\Sigma,\Omega)$ has an arc (M,N), then add an arc $(\overline{N}, \overline{M})$, if no such arc exists.

Stage 4:

Until no new node or arc can be added to $g(\Sigma,\Omega)$,

for each pair of nodes M and N such that M and N are labeled with ($\geq 1\ P$) and $\neg(\geq 1\ Q)$, respectively, and there is a path from M to N,

add arcs (K,L) and $(\overline{L}, \overline{K})$,where K and L are the nodes labeled with P and $\neg Q$, respectively, if no such arcs exists. □

Definition 2: The labeled graph $G(\Sigma,\Omega)$ that *represents* Σ and Ω, where each node is labeled 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. □

Definition 3: Let $G(\Sigma,\Omega)$ 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 there are nodes M and N, not necessarily distinct from K, and a positive 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 ($\geq k\ P$) or of the form ($\geq k\ P^-$) and there is a ⊥-node M of level n such that M is labeled with P, or

(c) K is labeled with an atomic role P and there is a ⊥-node M of level n such that M is labeled with a minCardinality constraint of the form ($\geq 1\ P$) or of the form ($\geq 1\ P^-$). □

In Case (i), 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.

Definition 4: Let $G(\Sigma,\Omega)$ 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 *⊥-node* iff K is a ⊥-node with level n, for some non-negative integer n.

(ii) K is a *⊤-node* iff $\overline{K}$ is a ⊥-node. □

To avoid repetitions, in what follows, let $G(\Sigma,\Omega)$ be the graph that represents Σ and Ω . Proposition 2 lists properties of $g(\Sigma,\Omega)$ that directly reflect the structure of the set of constraints Σ. Proposition 3 applies the results in Proposition 2 to obtain properties of $G(\Sigma,\Omega)$ that are fundamental to establish Lemma 1 and Theorems 1 and 2. Finally, Proposition 4 relates the structure of $G(\Sigma,\Omega)$ with the theory of Σ, i.e., the logical consequences of Σ.

Proposition 2: For any pair of nodes K and M of $g(\Sigma,\Omega)$,

(i) If (K,M) is an arc of $g(\Sigma,\Omega)$ and if M is labeled with a positive expression, then K is labeled with a positive expression.

(ii) If (K,M) is an arc of $g(\Sigma,\Omega)$ and if K is labeled with a negative expression, then M is labeled with a negative expression.

(iii) If there is a path $K \rightarrow M$ in $g(\Sigma,\Omega)$ and if M is labeled with a positive expression, then K is labeled with a positive expression.

(iv) If there is a path $K \rightarrow M$ in $g(\Sigma,\Omega)$ and if K is labeled with a negative expression, then M is labeled with a negative expression.

Proof

First observe that

(1) if $a \sqsubseteq b$ is a normalized constraint in Σ, then a is positive and b is either positive or negative. Furthermore, $\bar{b} \sqsubseteq \bar{a}$ is such that $\bar{b}$ positive or negative and $\bar{a}$ is negative.

Let K and M be a pair of nodes of $g(\Sigma,\Omega)$.

(i) Assume that (K,M) is an arc of $g(\Sigma,\Omega)$ and that M is labeled with a positive expression. Then, by (1) and Def. 1, K is labeled with a positive expression.

(ii) Assume that (K,M) is an arc of $g(\Sigma,\Omega)$ and that K is labeled with a negative expression. Then, by (1) and Def. 1, M is labeled with a negative expression.

(iii) Assume that there is a path $K \rightarrow M$ in $g(\Sigma,\Omega)$ and that M is labeled with a positive expression f. Suppose that K is labeled with an expression e. Since $K \rightarrow M$, there is a sequence of nodes $N_0, N_1, \ldots, N_m$ of $g(\Sigma,\Omega)$ respectively labeled with $h_0, h_1, \ldots, h_m$ such that $h_0 = e$, $h_m = f$ and (h_{i-1}, h_i) is an arc of $g(\Sigma,\Omega)$, for $i \in [0,m]$. By (backward) induction on the length of the path and using (i), since $h_m = f$ is a positive expression by assumption, we may conclude that $h_0 = e$ is a positive expression, as desired.

(iv) Assume that there is a path $K \rightarrow M$ in $g(\Sigma,\Omega)$ and that K is labeled with a negative expression e. Suppose that M is labeled with an expression f. Since $K \rightarrow M$, there is a sequence of nodes $P_0, P_1, \ldots, P_n$ of $g(\Sigma,\Omega)$ respectively labeled with $l_0, l_1, \ldots, l_n$ such that $l_0 = e$, $l_n = f$ and (l_{i-1}, l_i) is an arc of $g(\Sigma,\Omega)$, for $i \in [0,n]$. By (forward) induction on the length of the path and using (ii), since $l_0 = e$ is a negative expression by assumption, we may conclude that $l_n = f$ is a negative expression. □

Proposition 3:

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

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

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

(iv) For any node K of $G(\Sigma,\Omega)$, one of the following conditions holds:

- (a) K is labeled only with atomic concepts or minCardinality constraints of the form $(\geq m\ p)$, where p is either P or P^- and $m \geq 1$, or
- (b) K is labeled only with atomic roles, or
- (c) K 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 $m \geq 1$, or
- (d) K is labeled only with negated atomic roles.

(v) For any pair of nodes K and M of $G(\Sigma,\Omega)$,

- (a) If there is a path $K \rightarrow M$ in $G(\Sigma,\Omega)$ and if M is labeled with a positive expression, then K is labeled only with positive expressions.
- (b) If there is a path $K \rightarrow M$ in $G(\Sigma,\Omega)$ and if K is labeled with a negative expression, then M is labeled only with negative expressions.

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

(a) If K is a $\bot$-node, 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 $m \geq 1$, or K is labeled only with atomic roles.

(b) If K is a $\top$-node, then K is labeled only with negated atomic concepts or minCardinality constraints of the form $\neg(\geq m\ p)$, where p is either P or P^{-} and $m \geq 1$, or K is labeled only with negated atomic roles.

Proof

(i)(ii)(iii) Follow from the definition of $G(\Sigma,\Omega)$.

(iv-a) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is labeled with a positive concept expression e. Then, by construction of $G(\Sigma,\Omega)$, K is labeled only with concept expressions.

Suppose that K is labeled with a negative concept expression f. 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 Prop. 2(iii), since $f \rightarrow e$ is a path in $g(\Sigma,\Omega)$ and f is negative, e must be negative. Contradiction. Therefore, we may conclude that f cannot be a negative expression.

Thus, if K is labeled with a positive concept expression, then K is labeled only with positive concept expressions. Since such positive concept 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-b) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is labeled with a positive role expression e. Then, by construction of $G(\Sigma,\Omega)$, K is labeled only with role expressions, which are in fact atomic roles or negated atomic roles. By an argument similar to that in Case (iv-a), K can only be labeled with atomic roles.

(iv-c) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is labeled with a negative concept expression e. By (iii), $\overline{K}$ is labeled with $\overline{e}$, which is positive. Therefore, by (iv-a), $\overline{K}$ is labeled only with positive concept expressions. Hence, by (iii) again, K is labeled only with negative concept expressions, which must be negated atomic concepts or negated minCardinality constraints of the form $\neg(\geq m\ p)$, where p is either P or P^{-} and $m \geq 1$.

(iv-d) Let K be a node of $G(\Sigma,\Omega)$. Assume that K is labeled with a negative role expression e. Then, by construction of $G(\Sigma,\Omega)$, K is labeled only with role expressions, which are in fact atomic roles or negated atomic roles. By an argument similar to that in Case (iv-c), K can only be labeled with negative atomic roles.

(v-a) Let K and M be nodes of $G(\Sigma,\Omega)$. Assume that $K \rightarrow M$ and that M is labeled with a positive expression f. Let e be a label of K. Since, $K \rightarrow M$, there is a path $e \rightarrow f$ in $g(\Sigma,\Omega)$. By Prop. 2(iii), e must be a positive expression. Hence, K is labeled only with positive expressions.

(v-b) Let K and M be nodes of $G(\Sigma,\Omega)$. Assume that $K \rightarrow M$ and that K is labeled with a negative expression e. Let f be a label of M. Since, $K \rightarrow M$, there is a path $e \rightarrow f$ in $g(\Sigma,\Omega)$. By Prop. 2(iv), f must be a negative expression. Hence, M is labeled only with negative expressions.

(vi-a) Let K be a $\perp$-node $G(\Sigma,\Omega)$. The proof follows by induction on the $\perp$-level of K.

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

There are nodes M and N and a positive expression h such that M and N are respectively labeled with h and $\neg h$, and $K \rightarrow M$ and $K \rightarrow N$. Then, since $K \rightarrow M$ and M is labeled with a positive expression h, by (iv-a), (iv-b) and (v-a), K is labeled only with atomic concepts or minCardinalities of the form ($\geq m\ p$), where p is either P or P^{-} and $m \geq 1$, or K is labeled only with atomic roles.

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

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

Case I.1: There is a $\perp$-node M with level n such that $K \rightarrow M$. Then, by the induction hypothesis, the property holds for M. That is, M is labeled only with positive expressions. Hence, since $K \rightarrow M$, by (iv-a), (iv-b) and (v-a), K is labeled only with atomic concepts or minCardinality constraints of the form ($\geq m$ p), where p is either P or P^{-} and $m \geq 1$, or K is labeled only with atomic roles.

Case I.2: K is labeled with a minCardinality constraint of the form ($\geq 1\ P$) or ($\geq 1\ P^{-}$) and there is a $\perp$-node M with level n such that M is labeled with P. Then, by (iv-c), K is labeled only with atomic concepts or minCardinalities of the form ($\geq m\ p$), where p is either P or P^{-} and $m \geq 1$.

Case I.3: K is labeled with an atomic role P and there is $\perp$-node M with level n such that M is labeled with a minCardinality constraint of the form ($\geq 1\ P$) or ($\geq 1\ P^{-}$). Then, by (iv-d), K is labeled only with atomic roles.

(vi-b) Let L be a node of $G(\Sigma,\Omega)$. Assume that L is a $\top$-node. Then, by definition of $\top$-node, $\overline{L}$ is a $\perp$-node. Thus, by (vi-a), $\overline{L}$ is labeled only with atomic concepts or minCardinality constraints of the form ($\geq m\ p$), where p is either P or P^{-} and $m \geq 1$, or $\overline{L}$ is labeled only with atomic roles. Therefore, by (iii), 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 $m \geq 1$, or L is labeled only with negated atomic roles. □

Proposition 4:

(i) 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 $\Sigma \models e \sqsubseteq f$.

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

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

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

Proof

(i), (ii) First observe that, by construction of $g(\Sigma,\Omega)$, 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 $g(\Sigma,\Omega)$, if M and N are labeled with e and f, respectively, and 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 (i) and (ii) follow by the construction of $G(\Sigma,\Omega)$.

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

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

There are nodes M and N and a non-negative 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 $\Sigma \models e \sqsubseteq \perp$.

Induction hypothesis: Assume that the property holds when $\perp$-nodes with level n.

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

Case I.1: There is a $\perp$-node M with level n such that $K \rightarrow M$. Then, by the induction hypothesis and (i), $\Sigma \models e \sqsubseteq \perp$.

Case I.2: K is labeled with a minCardinality constraint of the form ($\geq 1\ P$) or ($\geq 1\ P^{-}$) and there is a $\perp$-node M with level n such that M is labeled with P. Assume that K is labeled with ($\geq 1\ P$) (the other case is identical). Then, by the induction hypothesis, $\Sigma \models P \sqsubseteq \perp$. But this implies that $\Sigma \models (\geq 1\ P) \sqsubseteq \perp$. Since K is labeled with e and ($\geq 1\ P$), by (vi), $\Sigma \models e \equiv (\geq 1\ P)$. Hence, $\Sigma \models e \sqsubseteq \perp$.

Case I.3: K is labeled with an atomic role P and there is $\perp$-node M with level n such that M is labeled with a minCardinality constraint of the form ($\geq 1\ P$) or ($\geq 1\ P^{-}$). Assume that M is labeled with ($\geq 1\ P$) (the other case is identical). Then, by the induction hypothesis, $\Sigma \models (\geq 1\ P) \sqsubseteq \perp$. But this implies that $\Sigma \models P \sqsubseteq \perp$. Since K is labeled with e and P, by (vi), $\Sigma \models e \equiv P$. Hence, $\Sigma \models e \sqsubseteq \perp$.

(iv) 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 $\perp$-node and $\bar{e}$ labels $\overline{K}$, by (iii). Hence, by (vii), $\Sigma \models \bar{e} \sqsubseteq \perp$. Thus, we have that $\Sigma \models \top \sqsubseteq e$. □

The next sequence of definitions leads to the notion of canonical Herbrand interpretation for a set of constraints, used to prove Lemma 1, which is the basis for Theorems 1 and 2.

Definition 5: Let Φ be a set of distinct *Skolem function symbols for* $G(\Sigma,\Omega)$ as follows:

(i) For each node N of $G(\Sigma,\Omega)$ labeled with ($\geq n\ P$), associate n distinct unary *Skolem function symbols* $f_1[N,P],\ldots, f_n[N,P]$

(ii) For each node N of $G(\Sigma,\Omega)$ labeled with ($\geq n\ P^{-}$), associate n distinct unary *Skolem function symbols* $g_1[N,P],\ldots, g_n[N,P]$.

(iii) For each node N of $G(\Sigma,\Omega)$ labeled with an atomic concept or with ($\geq 1\ P$), associate a distinct *Skolem constant* $c[N]$ (a constant is a 0-ary function symbol).

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

We say that a node K of $G(\Sigma,\Omega)$ is a *(positive or negative) concept expression node* iff K is labeled only with (positive or negative) concept expressions, and that K is a *(positive or negative) role expression node* iff K is labeled only with (positive or negative) role expressions. In view of Proposition 3(iv), any node K of $G(\Sigma,\Omega)$ is either a (positive or negative) concept expression node or a (positive or negative) role expression node.

Assume that K is and Again, to avoid repetitions, let Φ be a set of distinct Skolem function symbols for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$ 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 $G(\Sigma,\Omega)$ labeled with concept expressions, and a set of pairs of individuals in $\Delta[\Phi]$ to each node of $G(\Sigma,\Omega)$ labeled with role expressions.

(ii) Let N be a node of $G(\Sigma,\Omega)$ labeled with an atomic concept or with $(\geq k\ P)$. 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 $G(\Sigma,\Omega)$ labeled with the atomic role P. Assume that N_P is not a $\bot$-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

   a. the pair $(a, f_r[M,P](a))$ is called a *seed pair* of N_P *triggered by* $a \in s'(M)$, for $r \in [1,m]$.

   b. the term $f_r[M,P](a)$ is a *seed term* of the node L labeled with $(\geq 1\ P^-)$, 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 $G(\Sigma,\Omega)$ labeled with the atomic role P. Assume that N_P is not a $\bot$-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

   a. the pair $(g_r[N,P](b),b)$ is called a *seed pair* of N_P *triggered by* $b \in s'(N)$, for $r \in [1,n]$.

   b. the term $g_r[N,P](b)$ is a *seed term* of the node L labeled with $(\geq 1\ P)$, 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. □

Intuitively, the seed term of a node N will play the role of a unique signature of N, and likewise for a seed pair of a node N_P. The next definition captures this intuition.

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 a concept expression node, 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.

(ii) Assume that K is a role expression node, and that K is neither a $\bot$-node nor a $\top$-node. Then, $(t,u) \in s'(K)$ iff (t,u) is a seed pair of a node J and there is a path from J to K.

(iii) Assume that K is a $\bot$-node. Then, $s'(K) = \emptyset$.

(iv) Assume that K is a concept expression node and a $\top$-node. Then, $s'(K) = \Delta[\Phi]$.

(v) Assume that K is a role expression node and a $\top$-node. Then, $s'(K) = \Delta[\Phi] \times \Delta[\Phi]$. □

Proposition 5: 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 $G(\Sigma,\Omega)$, 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 a $\perp$-node and that both are concept expression nodes or both are role expression nodes, $s'(M) \cap s'(N) \neq \varnothing$ iff there is a seed node K such that $K\rightarrow M$ and $K\rightarrow N$.

(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

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

(i) Let M and N be a pair of nodes of $G(\Sigma,\Omega)$. Suppose that $M\rightarrow N$. There are 4 cases to consider.

Case 1: M is a $\perp$-node. Then, by Def. 7(iii), $s'(M)=\varnothing$, which trivially implies $s'(M) \subseteq s'(N)$.

Case 2: N is a $\top$-node. If N is a concept expression node, by Def. 7(iv), $s'(N) = \Delta[\Phi]$, which trivially implies $s'(M) \subseteq s'(N)$. If N is a role expression node, by Def. 7(v), $s'(N) = \Delta[\Phi] \times \Delta[\Phi]$, which also trivially implies $s'(M) \subseteq s'(N)$.

Case 3: M is a $\top$-node. By definition of $\top$-node, $\overline{M}$ is a $\perp$-node. Since by assumption $M\rightarrow N$, by Prop. 3(ii), $\overline{N} \rightarrow \overline{M}$. Then, by definition of $\perp$-node, $\overline{N}$ is also a $\perp$-node. Hence, N is also a $\top$-node. Thus, if N is a concept expression node, by Def. 7(iv), $s'(M) = \Delta[\Phi] = s'(N)$, and if N is a role expression node, by Def. 7(v), $s'(M) = \Delta[\Phi] \times \Delta[\Phi] = s'(N)$.

Case 4: M is neither a $\perp$-node nor a $\top$-node, and N is not a $\top$-node. Since M is not a $\perp$-node and $M\rightarrow N$, by definition of $\perp$-node, N is also not a $\perp$-node.

Case 4.1: Assume that M is a concept expression node. By construction of $G(\Sigma,\Omega)$, since $M\rightarrow N$, node N is also a concept expression node. We then have that M and N are concept expression nodes and they are not a $\perp$-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 $s'(M) \subseteq s'(N)$.

Case 4.2: Assume that M is a role expression node. By construction of $G(\Sigma,\Omega)$, since $M\rightarrow N$, node N is also a role expression node. We then have that M and N are role expression nodes and they are not a $\perp$-node or a $\top$-node. Hence, the conditions of Def. 7(ii) apply to both M and N. Let $(t,u)\in s'(M)$. By Def. 7(ii), (t,u) is a seed pair of a node J and $J\rightarrow M$. Since $M\rightarrow N$, we then have $J\rightarrow N$. Hence, by Def. 7(ii), $(t,u)\in s'(N)$. 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 $\bot$-node and that both are concept expression nodes or both are role expression nodes.

Then, since M and N are not a $\bot$-node, by Def. 7(i), $s'(M) \neq \emptyset$ and $s'(N) \neq \emptyset$.

Case 1: Either M or N is a $\top$-node. Assume that M and N are concept expression nodes. Then, either $s'(M)=\Delta[\Phi]$ or $s'(N)=\Delta[\Phi]$. Hence, since $s'(M) \neq \emptyset$ and $s'(N) \neq \emptyset$, we trivially have that $s'(M) \cap s'(N) \neq \emptyset$. Assume that M and N are role expression nodes. Then, either $s'(M)=\Delta[\Phi]\times\Delta[\Phi]$ or $s'(N)=\Delta[\Phi]\times\Delta[\Phi]$. Hence, since $s'(M) \neq \emptyset$ and $s'(N) \neq \emptyset$, we trivially have that $s'(M)\cap s'(N)\neq\emptyset$.

Case 2: Neither M nor N is a $\top$-node. By the assumptions, M and N are neither a $\bot$-node nor a $\top$-node. Assume that M and N are concept expression nodes. Hence, the conditions of Def. 7(i) apply to both M and N. By Def. 6, a term t cannot be a seed term of two distinct nodes. Then, $t\in s'(M)\cap s'(N)$ iff t is a seed term of a node J and $J\rightarrow M$ and $J\rightarrow N$. Assume that M and N are role expression nodes. Hence, the conditions of Def. 7(ii) apply to both M and N. By Def. 6, a pair (t,u) cannot be a seed pair of two distinct nodes. Then, $(t,u)\in s'(M)\cap s'(N)$ iff (t,u) is a seed pair of a node J and $J\rightarrow M$ and $J\rightarrow N$.

(iii) This property follows directly from Def. 7(ii), 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). □

Recall that the alphabet is understood as the (finite) set of atomic concepts and roles that occur in Σ and Ω. Hence, in the context of Σ and Ω, when we refer to an *interpretation*, we mean and interpretation for such alphabet.

Definition 8: Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and $\Delta[\Phi]$. *The canonical Herbrand interpretation induced by s'* is the interpretation s 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 canonical Herbrand interpretation induced by s'. Then, we have:

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

(ii) For each node N of $G(\Sigma,\Omega)$, for each negative 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 positive expression that labels N.

First observe that N cannot be a $\top$-node. Indeed, by Prop 3(vi-b), $\top$-nodes are labeled only with negative expressions, which contradicts the assumption that e is a positive expression. Then, there are two cases to consider.

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

We have to prove that $s(e) = s'(N)$.

By the restrictions on constraints and constraint expressions, since e is a positive expression, there are 4 cases to consider.

Case 1.1: e is an atomic concept C.

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

Case 1.2: e is an atomic role P.

By Def. 8(iii), $s(P) = s'(N)$.

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

Let N_P be the node labeled with P. Then, N_P is not a $\bot$-node. Indeed, assume otherwise. Then, by Def. 3(iv-b) and Def. 4, the node L labeled with ($\geq 1\ P$) would be a $\bot$-node. But, by construction of $G(\Sigma,\Omega)$, there is 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. Furthermore, since N_P is labeled with the positive atomic role P, by Prop. 3(vi-b), N_P cannot be a $\top$-node.

Then, since N_P is neither a $\bot$-node nor a $\top$-node, Def. 7(ii) applies to $s'(N_P)$.

Recall that N is the node labeled with ($\geq n\ P$) and that N is neither a $\bot$-node nor 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. 5(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 $(a,b_1),\ldots,(a,b_n)$ in $s(P)$ and, consequently, in $s'(N_P)$, since $s(P)=s'(N_P)$, by Def. 8(iii).

Recall that N_P is neither a $\bot$-node nor a $\top$-node. Then, by Def. 7(ii) and Def. 6(iii), possibly by reordering $b_1,\ldots,b_n$, we then have that there are nodes $L_0,L_1,\ldots L_v$ 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 1.3.1: $n=1$.

Case 1.3.1.1: a is of the form $g_{i0}[L_0,P](u)$.

Recall that N_P is not a $\bot$-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 1.3.1.2: 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. 5(i), for the second alternative. Therefore, $a \in s'(N)$ as desired, since $a \in s'(L_1)$.

Case 1.3.2: $n>1$.

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. 5(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. 5(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 1.4: 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 $\bot$-node.

We have to prove that $s(e) = s'(N) = \emptyset$.

Again, by the restrictions on constraints and constraint expressions, since e is a positive expression, there are 4 cases to consider.

Case 2.1: e is an atomic concept C.

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

Case 2.2: N is an atomic node P.

Then, by Def. 8(iii), we trivially have that $s(P) = s'(N) = \emptyset$.

Case 2.3: 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 1.3.

Let N_P be the node labeled with P.

Case 2.1.2.1: N_P is a $\bot$-node

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

Case 2.1.2.2: N_P is not a $\bot$-node.

By Prop. 3(vi-b), N_P cannot be a $\top$-node. Then, Def. 7(ii) applies to $s'(N_P)$.

We proceed by contradiction. So, 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 1.3, 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. 5(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. 5(i), for the second alternative. Therefore, $a \in s'(N)$, since $a \in s'(L_1)$. But this is impossible, since $s'(N)=\varnothing$.

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

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

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

(ii) Let N be a node of $G(\Sigma,\Omega)$. Let $\neg e$ be a negative expression that labels N.

First observe that N cannot be a $\bot$-node. Indeed, by Prop 3(vi-a), $\bot$-nodes are labeled only with positive expressions, which contradicts the assumption that $\neg e$ is a negative expression. Then, there are two cases to consider.

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

We have to prove that $s'(N) \subseteq s(\neg e)$.

Case 1.1: N is a concept expression 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 since M is labeled with e and N is labeled with $\neg e$. Therefore, since N is neither a $\bot$-node nor a $\top$-node, M is also neither a $\top$-node nor a $\bot$-node, by definition of $\top$-node. Hence, by Prop. 5(ii) and Def. 7(i), 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), $s'(K)=\varnothing$, which implies that $t \notin s'(K)$. Therefore, we established that, for all terms t, if $t \in s'(N)$ then $t \in s(\neg e)$.

Case 1.2: N is a role expression node.

Follows likewise, using Prop. 5(ii) again and Def. 7(ii).

Therefore, in both cases, we established that $s'(N) \subseteq s(\neg e)$, as desired.

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

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), $s'(\overline{N})=s(e)=\varnothing$.

Case 2.1: N is a concept expression node.

By Def. 7(iv) and definition of $s(\neg e)$, we have $s'(N)=\Delta[\Phi]=s(\neg e)$, which trivially implies $s'(N) \subseteq s(\neg e)$.

Case 2.2: N is a role expression node.

By Def. 7(v) and definition of $s(\neg e)$, we then have $s'(N) = \Delta[\Phi] \times \Delta[\Phi] = s(\neg e)$, which trivially implies $s'(N) \subseteq s(\neg e)$.

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

Theorem 1: Let s be the canonical Herbrand interpretation 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 e be an atomic concept or a minCardinality constraint of the form $(\geq 1\ P)$. Let N be the node of $G(\Sigma,\Omega)$ labeled with e. Assume that N is not a $\bot$-node. Then $s(e)\neq\emptyset$.

(iii) Let e be a minCardinality constraint of the form $(\geq k\ P)$, with $k > 1$. If $G(\Sigma,\Omega)$ has a node labeled with e which is not a $\bot$-node, then $s(e)\neq\emptyset$.

(iv) Let P be an atomic role. Let N be the node of $G(\Sigma,\Omega)$ labeled with P. Assume that N is not a $\bot$-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 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 positive and f can be positive or negative. Therefore, there are two cases to consider.

Case 1: e and f are both positive.

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. 5(i), we then have $s'(M) \subseteq s'(N)$. Hence, $s(e) = s'(M) \subseteq s'(N) = s(f)$.

Case 2: e is positive and f is negative.

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 negative 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. 5(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 e be an atomic concept or a minCardinality constraint of the form $(\geq 1\ P)$. By Stage 1 of Def. 1, $G(\Sigma,\Omega)$ always has a node N labeled with e. Assume that N is not a $\bot$-node. Then, by Lemma 1(i), $s(e)=s'(N)$.

Note that N cannot be a $\top$-node, since N is labeled with the positive expression e. 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 $c[N] \in s'(N)$. Hence, trivially, $s(e)=s'(N)\neq\emptyset$.

(iii) Let e be a minCardinality constraint of the form $(\geq k\ P)$, with $k > 1$. Assume that $G(\Sigma,\Omega)$ has a node N labeled with e which is not a $\bot$-node. Then, by Lemma 1(i), $s(e)=s'(N)$.

Note that N cannot be a $\top$-node, since N is labeled with the positive expression e. 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 $c[N] \in s'(N)$. Hence, trivially, $s(e)=s'(N)\neq\emptyset$.

(iv) Let P be an atomic role. By Stage 1 of Def. 1, $G(\Sigma,\Omega)$ always has a node N labeled with P and a node M labeled with $(\geq 1\ P)$. Assume that N is not a $\bot$-node. Hence, by Definition 3(ii-c), M is not a $\bot$-node. Then, by Theorem 1(i), $s((\geq 1\ P)) \neq \emptyset$, which implies that $s(P) \neq \emptyset$. □

A.2 Proof of Theorem 2

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. 4 (iii), $\Sigma \models e \sqsubseteq \bot$, which trivially implies that $\Sigma \models e \sqsubseteq f$.

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

Then, by Prop. 4 (iv), $\Sigma \models \top \sqsubseteq f$, 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. 4(i) and (ii), we have that $\Sigma \models e \sqsubseteq f$.

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

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

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

(2) f is either an atomic concept C, a negated atomic concept $\neg D$, an atomic role P, a negated atomic role Q, a minCardinality constraint of the form $(\geq k\ p)$, or a negated minCardinality constraint 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 $\Sigma \not\models e \sqsubseteq f$, it suffices to exhibit a model r of Σ such that $r \not\models e \sqsubseteq f$. Recall that $r \not\models e \sqsubseteq f$ iff (i) if e and f are concept expressions, there is an individual t such that $t \in r(e)$ and $t \notin r(f)$ or, equivalently, $t \in r(\neg f)$; (ii) if e and f are role expressions, there is a pair of individuals (t,u) such that $(t,u) \in r(e)$ and $(t,u) \notin r(f)$ or, equivalently, $(t,u) \in r(\neg f)$;

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-negative and f can be negative or not. Hence, there are 2 cases to consider.

Case 1: e and f are both positive.

Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and s be the interpretation induced by s'. By Theorem 1, s is a model of Σ. We show that $s \not\models e \sqsubseteq f$.

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

Since N is a $\bot$-node, by Prop. 4(iii), we have that $\Sigma \models f \sqsubseteq \bot$, which implies that $s(f)=\emptyset$, since s is a model of Σ.

By (1), e is either an atomic concept C, an atomic role P or a minCardinality of the form $(\geq k\ p)$, where p is either P or P^{-}. By (3), M is not a $\bot$-node. Hence, we have that $s(e) \neq \emptyset$, by Theorem 1 (ii), (iii) and (iv). Hence, we trivially have that $s \not\models e \sqsubseteq f$.

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, by Prop. 2(iv-b), since M is labeled with a positive expression e, M cannot be a $\top$-node.

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

(6) $s'(M)=s(e)$ and there is a seed term $c[M] \in s'(M)$, if M is a concept expression node
$s'(M)=s(e)$ and there is a seed pair $(t,u) \in s'(M)$, if M is a role expression node

By definition of canonical instance labeling function, we have:

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

For each role expression node K of $G(\Sigma,\Omega)$ that is neither a $\bot$-node nor a $\top$-node, $(t,u)\in s'(K)$ 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) $c[M]\notin s'(N)$, if N is a concept expression node

$(t,u)\notin s'(N)$, if N is a role expression node

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

(9) $c[M]\notin s(f)$, if f is a concept expression

$(t,u)\notin s(f)$, if f is a role expression

Therefore, by (6) and (9), $s(e)\nsubseteq s(f)$, that is, $s \nvDash e \sqsubseteq f$, as desired.

Case 2: e is positive and f is negative.

Assume that f is a negative expression of the form $\neg g$, where g is positive.

Case 2.1: $e\rightarrow g$.

Let s' be a canonical instance labeling function for $G(\Sigma,\Omega)$ and s be the interpretation induced by s'. By Theorem 1(i), s is a model of Σ. We show that $s \nvDash e \sqsubseteq f$.

By Prop. 4(i) and (ii), and since s is a model of Σ, we have that $s \vDash e \equiv g$, if e and g label the same node, and $s \vDash e \sqsubseteq g$, otherwise. Hence, we have that $s \nvDash e \sqsubseteq \neg g$. Now, since f is $\neg g$, we have $s \nvDash e \sqsubseteq f$, 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, if e and g are concept expressions, or H is a new atomic role, if e and g are role expressions

Let r' be a canonical instance labeling function for $G(\Phi,\Omega)$ and r be the interpretation induced by r'. By Theorem 1(i), r is a model of Φ. We show that $r \nvDash e \sqsubseteq f$.

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 positive expression h such that $H\rightarrow h$ and $H\rightarrow\neg h$ are paths in $G(\Phi,\Omega)$

Assume otherwise. Let h be a positive 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 Theorem 1(i), r is a model of Φ. Furthermore, by Theorem 1(ii) and (iv), 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 $r \not\models e \sqsubseteq f$, as desired.

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

Corollary 1: Let Σ be a set of normalized constraints. Let $e \sqsubseteq f$ be a constraint. Let $G(\Sigma)$ 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 $\bot$-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, an atomic role or 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 $G(\Sigma,\Omega)$, recalling that $\Omega = \{e,f\}$. Assume that e does not label a node of $G(\Sigma)$. Then, by construction of $G(\Sigma,\Omega)$, if M is a $\bot$-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 $G(\Sigma,\Omega)$, recalling that $\Omega = \{e,f\}$. 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 $m<n$. □

A.3 Proof of Theorem 3

Theorem 3. Let Γ be a set of normalized constraints that contains neither role disjunctions nor maxCardinality constraints and that Γ is role acyclic. Then, Γ is strictly satisfiable iff $C(\Gamma)$ is strictly satisfiable.

Proof

Let Γ be a set of normalized constraints. Assume that Γ is role acyclic and that Γ contains neither role disjunctions nor maxCardinality constraints.

($\Rightarrow$) Assume that Γ is strictly satisfiable. Recall that $P \sqsubseteq Q$ logically implies ($\geq k\ P$) $\sqsubseteq$ ($\geq k\ Q$) and ($\geq k\ P^-$) $\sqsubseteq$ ($\geq k\ Q^-$). Then, we immediately have that $C(\Gamma)$ is strictly satisfiable.

($\Leftarrow$) Assume that $C(\Gamma)$ is strictly satisfiable. Then, there is a strict model $\boldsymbol{s}$ for $C(\Gamma)$. Construct an interpretation $\boldsymbol{r}$ for Γ as follows:

(1) For each atomic concept C, let $\boldsymbol{r}(C) = \boldsymbol{s}(C)$

(2) For each atomic role P, let $\boldsymbol{r}(P)$ be as follows:

(2.1) if P has level 0, then $\boldsymbol{r}(P) = \boldsymbol{s}(P)$

(2.2) if P has level $m > 0$, then $\boldsymbol{r}(P) = \boldsymbol{s}(P) \cup \boldsymbol{r}(P_1) \cup \ldots \cup \boldsymbol{r}(P_n)$, where $P_1 \sqsubseteq P, \ldots, P_n \sqsubseteq P$ is the set of role inclusions in Γ such that P occurs on the right-hand side of the inclusion (note that $P_1, \ldots, P_n$ must then have level m-1, since P has level m)

We first show that $\boldsymbol{r}$ is a strict interpretation for Γ. Indeed, observe that, since $\boldsymbol{s}$ is a strict model for $C(\Gamma)$, we have $\boldsymbol{s}(C) \neq \emptyset$ and $\boldsymbol{s}(P) \neq \emptyset$. Hence, $\boldsymbol{r}(C) = \boldsymbol{s}(C) \neq \emptyset$ and $\boldsymbol{r}(P) = \boldsymbol{s}(P) \cup \boldsymbol{r}(P_1) \cup \ldots \cup \boldsymbol{r}(P_n) \neq \emptyset$. We now show that $\boldsymbol{r}$ is a model for Γ, that is, we show that $\boldsymbol{r}$ satisfies each formula γ in Γ. Since Γ does not contain role disjunctions or maxCardinality constraints, we only have to analyse the following cases:

Case 1: γ is a concept inclusion of the form $C \sqsubseteq D$. By Def. 5(i), γ is in $C(\Gamma)$. Then, since $\boldsymbol{s}$ is a model of $C(\Gamma)$, $\boldsymbol{s}(C) \subseteq \boldsymbol{s}(D)$. Hence, by (1), we have that $\boldsymbol{r}(C) = \boldsymbol{s}(C) \subseteq \boldsymbol{s}(D) = \boldsymbol{r}(D)$. Thus, $\boldsymbol{r}$ satisfies $C \sqsubseteq D$.

Case 2: γ is a concept disjunction of the form $C \mid D$. Follows as in Case 1.

Case 3: γ is a role inclusion of the form $P \sqsubseteq Q$. Follows as in Case 1, but using (2).

Case 4: γ is of the form $\exists P \sqsubseteq C$, normalized as $(\geq 1\ P) \sqsubseteq C$. By Def. 5(i), γ is in $C(\Gamma)$. Then, since $\boldsymbol{s}$ is a model of $C(\Gamma)$

(3) $\boldsymbol{s}((\geq 1\ P)) \subseteq \boldsymbol{s}(C)$

Assume that P has level m=0. Then, by (2.1), we trivially have that

(4) $\boldsymbol{r}((\geq 1\ P)) = \boldsymbol{s}((\geq 1\ P))$

Assume that P has level m>0. Let $P_1 \sqsubseteq P, \ldots, P_n \sqsubseteq P$ be the set of all role inclusions in Γ with P on the right-hand side. Using Def. 5(ii), with k=1, by construction of $C(\Gamma)$, we have

(5) $C(\Gamma)$ logically implies $(\geq 1\ P_i) \sqsubseteq (\geq 1\ P)$, for $i \in [1,n]$

Then, since $\boldsymbol{s}$ is a model of $C(\Gamma)$, by (5), we have

(6) $\boldsymbol{s}((\geq 1\ P_i)) \subseteq \boldsymbol{s}((\geq 1\ P))$, for $i \in [1,n]$

Therefore, by (2.2), (6) and definition of the interpretation of $(\geq 1\ P)$ and $(\geq 1\ P_i)$, for $i \in [1,n]$, we have

(7) $\boldsymbol{r}((\geq 1\ P)) = \boldsymbol{s}((\geq 1\ P)) \cup \boldsymbol{s}((\geq 1\ P_1)) \cup \ldots \cup \boldsymbol{s}((\geq 1\ P_n)) = \boldsymbol{s}((\geq 1\ P))$

Hence, $\boldsymbol{r}$ satisfies $(\geq 1\ P) \sqsubseteq C$ since, by (4), (7), (3) and (1), we have

(8) $\boldsymbol{r}((\geq 1\ P)) = \boldsymbol{s}((\geq 1\ P)) \subseteq \boldsymbol{s}(C) = \boldsymbol{r}(C)$

Case 5: γ is of the form $\exists P^- \sqsubseteq C$, normalized as $(\geq 1\ P^-) \sqsubseteq C$. Follows as in Case 4, replacing (5) by

(5') $C(\Gamma)$ logically implies $(\geq 1\ P_i^-) \sqsubseteq (\geq 1\ P^-)$, for $i \in [1,n]$

Case 6: γ is of the form $C \sqsubseteq (\geq k\ P)$. By Def. 5(i), γ is in $C(\Gamma)$. Then, since $\boldsymbol{s}$ is a model of $C(\Gamma)$

(9) $\boldsymbol{s}(C) \subseteq \boldsymbol{s}((\geq k\ P))$

By (2), we have $\boldsymbol{s}(P) \subseteq \boldsymbol{r}(P)$, which implies that

(10) $\boldsymbol{s}((\geq k\ P)) \subseteq \boldsymbol{r}((\geq k\ P))$

Hence, $\boldsymbol{r}$ satisfies $C \sqsubseteq (\geq k\ P)$ since, by (1), (9) and (10), we have

(11) $\boldsymbol{r}(C) = \boldsymbol{s}(C) \subseteq \boldsymbol{s}((\geq k\ P)) \subseteq \boldsymbol{r}((\geq k\ P))$

Case 7: γ is of the form $C \sqsubseteq (\geq k\ P^-)$. Follows as in Case 6, replacing (10) by

(10') $\boldsymbol{s}((\geq k\ P^-)) \subseteq \boldsymbol{r}((\geq k\ P^-))$. □