

Chapter 1

On the Problem of Matching Database Schemas

Marco A. Casanova, Karin K. Breitman, Antonio L. Furtado, Vânia M.P. Vidal,
and José A. F. de Macêdo

Abstract This chapter introduces procedures to test strict satisfiability and decide logical implication for a family of database schemas called *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, datatypes, 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 chapter 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 *constraint graph*. The chapter includes a proof that the procedures work in polynomial time and are consistent and complete for a subfamily of the schemas that impose a restriction on the role hierarchy.

1.1 Introduction

The problem 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. 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.

We may repeat similar remarks for the problem of detecting redundancies in a schema, that is, the problem of detecting which constraints are logically implied by others. The situation is analogous if we replace the question of satisfiability by the question of logical implication.

A third similar, but more sophisticated problem is to automatically generate the constraints of a mediated schema from the sets of constraints of export schemas. The constraints of the mediated schema are relevant for a correct understanding of what the semantics of the external schemas have in common.

With this motivation, we focus on the crucial challenge of selecting a sufficiently expressive family of schemas that is useful for defining real-world schemas and yet is tractable, i.e., for which there are practical procedures to test the satisfiability of a schema, to detect redundancies in a schema, and to combine the constraints of export schemas into a single set of mediated schema constraints. The intuitive metrics for expressiveness here is that the family of schemas should account for the commonly used conceptual constructs of OWL, UML, and the ER model. By a practical procedure, we mean a procedure that is polynomial on the size of the set of constraints of the schema.

As an answer to this challenge, we first introduce a family of schemas that we call *extralite schemas with role hierarchies*. Using the OWL jargon, this family supports 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). We then introduce the subfamily of *restricted extralite schemas with role hierarchies*, which limits the interaction between role hierarchies and cardinality constraints.

Testing satisfiability for extralite schemas with role hierarchies turns out to be EXPTIME-hard, as a consequence of the results in [1]. However, for the restricted schemas, we show how to test strict satisfiability and logical implication in polynomial time. Strict satisfiability imposes the additional restriction that the constraints of a schema must not force classes or properties to be always empty, and is more adequate than the traditional notion of satisfiability in the context of database design.

The syntax and semantics of extralite schemas is that of Description Logics to facilitate the formal analysis of the problems we address. However, we depart from the tradition of Description Logics deduction services, which are mostly based on tableaux techniques [3]. The decision procedures outlined in the chapter are based on the satisfiability algorithm for Boolean formulas in conjunctive normal form with at most two literals per clause, described in [2]. The intuition is that the constraints of an extralite schema can be treated much in the same way as Boolean implications. Furthermore, the implicational structure of the constraints can be completely captured as a *constraint graph*. The results also depend on the notion of Herbrand interpretation for Description Logics.

The notion of constraint graph is the key to meet the challenge posed earlier. It permits expressiveness and decidability to be balanced, in the sense that it accounts for a useful family of constraints and yet leads to decision procedures, which are polynomial on the size of the set of constraints. This balancing is achieved by a careful analysis of how the constraints interact.

Constraint graphs can be used to help detect inconsistencies in a set of constraints and to suggest alternatives to fix the problem. They help solve the query containment and related problems in the context of schema constraints [10]. They can also be used to compute the greatest lower bound of two sets of constraints, which is the basic step of a strategy to automatically generate the constraints of a mediated schema from the sets of constraints of the export schemas [8]. The appendix illustrates, with the help of examples, how to use constraint graphs to address such problems.

The main contributions of the chapter are the family of extralite schemas with role hierarchies, the procedures to test strict satisfiability and logical implication, which explore the structure of sets of constraints, captured as a constraint graph, and the concept of Herbrand interpretation for Description Logics. The results in the chapter indicate that the procedures are consistent and complete for restricted extralite schemas with role hierarchies, and work in polynomial time. These results extend those published in [8] for extralite schemas without role hierarchies.

There is a vast literature on the formal verification of database schemas and on the formalization of ER and UML diagrams. We single out just a few references here. The problem of modeling conceptual schemas in DL is discussed in [4]. DL-Lite is used, for example, in [5, 6] to address schema integration and query answering. A comprehensive survey of the DL-Lite family can be found in [1]. Techniques from Propositional Logic to support the specification of Boolean and multivalued dependencies were addressed in [9].

When compared with the DL-Lite family [1], extralite schemas with role hierarchies are a subset $DL-Lite_{core}^{\mathcal{H}, \mathcal{N}}$ with role disjunctions. The restricted schemas in turn are a subset of $DL-Lite_{core}^{(\mathcal{H}, \mathcal{N})}$ with role disjunctions only, which limits the interaction between role inclusions and cardinality constraints. We emphasize that restricted extralite schemas are sufficiently expressive to capture the most familiar constructs of OWL, UML, and the ER model [4], and yet come equipped with useful decision procedures that explore the structure of sets of constraints.

The chapter is organized as follows. Section 1.2 reviews DL concepts and introduces the notion of extralite schemas with role hierarchies. Section 1.3 shows how to test strict satisfiability and logical implication for restricted extralite schemas with role hierarchies. It also outlines proofs for the major results of the chapter, whose details can be found in [7]. Section 1.4 contains examples of the concepts introduced in Sections 1.2 and 1.3, and briefly discusses two applications of the results of Section 1.3. Finally, Section 1.5 contains the conclusions.

1.2 A Class of Database Schemas

1.2.1 A Brief Review of Attributive Languages

We adopt a family of *attributive languages* [3] defined as follows. A *language* \mathcal{L} in the family is characterised 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 \perp , respectively, and the *universal role* and the *bottom role*, also denoted by \top and \perp , 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*
 $\exists p$: *existential quantification*
 $(\leq n p)$: *at-most restriction*
 $(\geq n p)$: *at-least restriction*

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

$s(\top) = \Delta^s$ if \top denotes the universal concept
 $s(\top) = \Delta^s \times \Delta^s$ if \top denotes the universal role
 $s(\perp) = \emptyset$ if \perp 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 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):

$s(p^-) = s(p)^-$: the *inverse* of $s(p)$
 $s(\neg p) = \Delta^s \times \Delta^s - s(p)$: the *complement* of $s(p)$ with respect to $\Delta^s \times \Delta^s$
 $s(\neg e) = \Delta^s - s(e)$: the *complement* of $s(e)$ with respect to Δ^s
 $s(\exists p) = \{I \in \Delta^s / (\exists J \in \Delta^s) / (I, J) \in s(p)\}$: the set of individuals that $s(p)$ relates to some individual
 $s(\geq n p) = \{I \in \Delta^s / |\{J \in \Delta^s / (I, J) \in s(p)\}| \geq n\}$: the set of individuals that $s(p)$ relates to at least n distinct individuals

$s(\leq n p) = \{I \in \Delta^s / \{J \in \Delta^s / (I, J) \in s(p)\} |\leq n\}$: the set of individuals that $s(p)$ 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 s for \mathcal{L} satisfies $u \sqsubseteq v$ iff $s(u) \subseteq s(v)$, s satisfies $u \mid v$ iff $s(u) \cap s(v) = \emptyset$, and s satisfies $u \equiv v$ iff $s(u) = s(v)$. A formula σ is a *tautology* iff any interpretation satisfies σ . Two formulas are *tautologically equivalent* iff any interpretation s that satisfies one formula also satisfies the other.

Given a set of formulas Σ , we say that an interpretation s is a *model* of Σ iff s satisfies all formulas in Σ , denoted $s \models \Sigma$. 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 s is a *strict model* of Σ iff 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 .

1.2.2 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 are atomic concepts, P and Q are 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 individuals)

maxCardinality constraint: $C \sqsubseteq (\leq k p)$ (p maps each individual in C to at most k individuals)

Concept Subset Constraint: $C \sqsubseteq D$ (C is a subset of D)

Concept Disjointness Constraint: $C \mid D$ (C and D are disjoint atomic concepts)

Role Subset Constraint: $P \sqsubseteq Q$ (P is a subset of Q)

Role Disjointness Constraint: $P \mid Q$ (P and Q are disjoint atomic roles)

We say that a formula of one of the above forms is an *extralite constraint*, the concept subset and disjointness constraints of S are the *concept hierarchy* of S , and the role subset and disjointness constraints of S are the *role hierarchy* of S .

We *normalize* a set of extralite constraints by rewriting:

$$\begin{aligned} \exists P \sqsubseteq C & \quad \text{as } (\geq 1 P) \sqsubseteq C \\ \exists P^- \sqsubseteq C & \quad \text{as } (\geq 1 P^-) \sqsubseteq C \\ C \sqsubseteq (\leq k P) & \quad \text{as } C \sqsubseteq \neg(\geq k+1 P) \\ C \sqsubseteq (\leq k P^-) & \quad \text{as } C \sqsubseteq \neg(\geq k+1 P^-) \\ C \mid D & \quad \text{as } C \sqsubseteq \neg D \\ P \mid Q & \quad \text{as } P \sqsubseteq \neg Q \end{aligned}$$

The formula on the right-hand side is called the *normal form* of the formula on the left-hand side. 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.

Furthermore, we *close* the set of extralite constraints by also considering as an extralite constraint any inclusion of one of the forms

$$\begin{aligned} C \sqsubseteq \perp \quad (\geq m p) \sqsubseteq \perp \quad (\geq m p) \sqsubseteq (\geq n q) \\ P \sqsubseteq \perp \quad (\geq m p) \sqsubseteq \neg C \quad (\geq m p) \sqsubseteq \neg(\geq n q) \end{aligned}$$

where C is an atomic concept, P is an atomic role, p and q both are atomic roles or both are the inverse of atomic roles, and m and n are positive integers.

Finally, a *restricted extralite schema with role hierarchies* is a schema $S = (\mathcal{A}, \Sigma)$ that satisfies the following restriction:

Restriction. Role Hierarchy Restriction: *If Σ contains a role subset constraint of the form $P \sqsubseteq Q$, then Σ contains no maxCardinality constraints of the forms $C \sqsubseteq (\leq k Q)$ or $C \sqsubseteq (\leq k Q^-)$, with $k \geq 1$.*

Note that the normalization process will rewrite the above constraints as $C \sqsubseteq \neg(\geq k+1 Q)$ and $C \sqsubseteq \neg(\geq k+1 Q^-)$, with $k \geq 1$.

1.3 Testing Strict Satisfiability and Logical Implication

This section first introduces the notion of constraint graph. Then, it defines Herbrand interpretations for Description Logics. Finally, it states results that lead to simple polynomial procedures to test strict satisfiability and logical implication for restricted extralite schemas with role hierarchies.

1.3.1 Representation Graphs

Let Σ be a finite set of normalized extralite constraints and Ω be a finite set of extralite 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 \bar{d} . Proposition 1.1 states properties of descriptions that will be used in the rest of this section.

Proposition 1.1. *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 $(\geq k P^-) \sqsubseteq (\geq k Q^-)$.
- (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 $P \sqsubseteq \neg Q$.
- (vi) If Σ logically implies $e \sqsubseteq f$ and $e \sqsubseteq \neg f$, then Σ logically implies $e \sqsubseteq \perp$.
- (vii) If Σ logically implies $(\geq 1 P) \sqsubseteq \perp$ or $(\geq 1 P^-) \sqsubseteq \perp$, then Σ logically implies $P \sqsubseteq \perp$.
- (viii) If Σ logically implies $P \sqsubseteq \perp$, then Σ logically implies $(\geq m P) \sqsubseteq \perp$, $(\geq m P^-) \sqsubseteq \perp$, $\top \sqsubseteq (\leq n P)$ and $\top \sqsubseteq (\leq n P^-)$, where $m > 0$ and $n \geq 0$.

In the next definitions, we introduce graphs whose nodes are labeled with expressions or sets of expressions. Then, we use such graphs to create efficient procedures to test if Σ is strictly satisfiable and to decide logical implication for Σ . Finally, it will become clear when we formulate Theorem 1.2 that the definitions must also consider an additional set Ω of constraint expressions.

To simplify the definitions, 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 1.1. 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 (\bar{N}, \bar{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 (\bar{L}, \bar{K}) , where K and L are the nodes labeled with P and $\neg Q$, respectively, if no such arcs exists.

Note that 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 1.2. The *constraint graph that represents* Σ and Ω is the labeled graph $G(\Sigma, \Omega)$, where each node is labeled with a set of expressions, 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 $G(\Sigma)$ is the *constraint graph that represents* Σ .

Note that Definition 1.2 reflects Proposition 1(iii).

Definition 1.3. Let $G(\Sigma, \Omega)$ be the constraint 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 holds:

- (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 $K \rightarrow N$.

- (ii) K is a \perp -node with level $n+1$ iff
- (a) There is a \perp -node M of level n , distinct from K , such that $K \rightarrow M$, and M is the \perp -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 \perp -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 \perp -node M of level n such that M is labeled with a minCardinality constraint of the form $(\geq 1 P)$ or of the form $(\geq 1 P^-)$.

Note that cases (i) and (ii-a) of Definition 1.3 correspond to Proposition 1(vi), case (ii-b) to Proposition 1(viii), and case (ii-c) to Proposition 1(vii).

Definition 1.4. A node K is a \perp -node of $G(\Sigma, \Omega)$ iff K is a \perp -node with level n , for some non-negative integer n . A node K is a \top -node of $G(\Sigma, \Omega)$ iff \bar{K} is a \perp -node.

To avoid repetitions, in what follows, let $g(\Sigma, \Omega)$ be the graph that captures Σ and Ω and $G(\Sigma, \Omega)$ 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 1.1 and Theorems 1 and 2. Finally, Proposition 4 relates the structure of $G(\Sigma, \Omega)$ with the logical consequences of Σ .

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

- (i) 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.
- (ii) 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.

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 \bar{K} .
- (iii) For any pair of nodes M and N of $G(\Sigma, \Omega)$, we have that $M \rightarrow N$ iff $\bar{N} \rightarrow \bar{M}$.
- (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 \perp -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 negated 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.
- (vii) Assume that Σ has no inclusions of the form $e \sqsubseteq \neg(\geq k P)$ or of the form $e \sqsubseteq \neg(\geq k P^-)$. Let M be the node labeled with $\neg(\geq k P)$ (or with $\neg(\geq k P^-)$). Then, for any node K of $G(\Sigma, \Omega)$, if there is a path $K \rightarrow M$ in $G(\Sigma, \Omega)$, then K is labeled only with negative concept expressions.

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 $\Sigma \models \top \sqsubseteq e$.

1.3.2 Herbrand Interpretations and Instance Labeling Functions

To prove the main results, we introduce in this section the notion of canonical Herbrand interpretation for a set of constraints. The definition mimics the analogous notion used in automated theorem proving strategies based on Resolution.

Definition 1.5.

- (i) A set Φ of distinct function symbols is a set of Skolem function symbols for $G(\Sigma, \Omega)$ iff Φ associates:
 - (a) n distinct unary function symbols with each node N of $G(\Sigma, \Omega)$ labeled with $(\geq n P)$, denoted $f_1[N, P], \dots, f_n[N, P]$ for ease of reference;
 - (b) n distinct unary function symbols with each node N of $G(\Sigma, \Omega)$ labeled with $(\geq n P^-)$, denoted $g_1[N, P], \dots, g_n[N, P]$ for ease of reference;
 - (c) a distinct constant with each node N of $G(\Sigma, \Omega)$ labeled with an atomic concept or with $(\geq 1 P)$, denoted $c[N]$ for ease of reference.
- (ii) 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*.

In the next definition, recall that use $Q \rightarrow P$ to indicate that there is a path from a node Q to a node P in $G(\Sigma, \Omega)$.

Definition 1.6.

- (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 \perp -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 \perp -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 $(\geq k Q)$ such that $m \leq k$, $Q \rightarrow P$ 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 \perp -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 $(\geq k Q^-)$ such that $n \leq k$, $Q \rightarrow P$ 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 $r \in [1, n]$, and
 - (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 .

Definition 1.7. 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 $G(\Sigma, \Omega)$:

- (a) Assume that K is a concept expression node, and that K is neither a \perp -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 .
- (b) Assume that K is a role expression node and is neither a \perp -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 .
- (c) Assume that K is a \perp -node. Then, $s'(K) = \emptyset$.

- (d) Assume that K is a concept expression node and is a \top -node. Then, $s'(K) = \Delta[\Phi]$.
- (e) Assume that K is a role expression node and is 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 both are concept expression nodes or both are role expression nodes, $s'(M) \cap s'(N) \neq \emptyset$ 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 $t \in s'(N)$.

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 an interpretation for such alphabet.

Definition 1.8. 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:

- (a) $\Delta[\Phi]$ is the domain of s .
- (b) $s(C) = s'(M)$, for each atomic concept C , where M is the node of $G(\Sigma, \Omega)$ labeled with C (there is just one such node).
- (c) $s(P) = s'(N)$, for each atomic role P , where N is the node of $G(\Sigma, \Omega)$ labeled with P (again, there is just one such node).

1.3.3 Strict Satisfiability and Logical Implication for Extralite Schemas with Restricted Role Hierarchies

We now ready to prove the main results of the chapter that lead to efficient decision procedures to test strict satisfiability and logical implication for restricted extralite schemas with role hierarchies.

In what follows, let Σ be a finite set of normalized extralite constraints and Ω be a finite set of extralite constraint expressions. Let $G(\Sigma, \Omega)$ be the graph that represents Σ and Ω .

Lemma 1.1. *Assume that Σ satisfies the role hierarchy restriction. 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 Sketch 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. 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 \perp -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 four cases to consider.

Case 1.1: e is an atomic concept C .

By Def. 1.8(b), $s(C) = s'(N)$.

Case 1.2: e is an atomic role P .

By Def. 1.8(c), $s(P) = s'(N)$.

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 \perp -node. Assume otherwise. Then, by Def. 1.3(ii-b) and Def. 1.4, the node L labeled with $(\geq 1 P)$ would be a \perp -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 \perp -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 \perp -node nor a \top -node, Def. 1.7(b) applies to $s'(N_P)$.

Recall that N is the node labeled with $(\geq n P)$ and that N is neither a \perp -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 integer greater than n . 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. 1.8(c), $s(P) = s'(N_P)$. 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 $a \in s'(N)$

Since Σ satisfies the role hierarchy restriction, there are two cases to consider.

Case 1.3.1: Σ defines no subroles for P .

Let $a \in s(\geq n P)$. By definition of minCardinality , there must be n distinct pairs $(a, b_1), \dots, (a, b_n)$ in $s(P)$ and, consequently, in $s'(N_P)$, since $s(P) = s'(N_P)$, by Def. 1.8(c).

Recall that N_P is neither a \perp -node nor a \top -node. Then, by Def. 1.7(b) and Def. 1.6(iii), possibly by reordering b_1, \dots, b_n , we then have that there are nodes L_0, L_1, \dots, L_v such that

- (3) (a, b_1) is a seed pair of N_P of the form $(g_{i_0}[L_0, P](u), u)$, 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 $(a, f_{l_1}[L_1, P](a))$, 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 $(a, f_{w_j}[L_i, P](a))$, 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 $l_1 > l_2 > \dots > l_v$. 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 $(a, f_{w_j}[L_j, P](a))$ is a seed pair of N_P triggered by $a \in s'(L_j)$ since, by Def. 1.6(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 there is just one node, L_1 , that satisfies (5).

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

Case 1.3.1.1: $n = 1$.

Case 1.3.1.1.1: a is of the form $g_{i_0}[L_0, P](u)$.

Recall that N_P is not a \perp -node. Then, by Def. 1.6(iv), $g_{i_0}[L_0, P](u)$ 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 \perp -node or a \top -node, by Def. 1.7(a), $a \in s'(N)$.

Case 1.3.1.1.2: a is not of the form $g_{i_0}[L_0, P](u)$.

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 1.3.1.2: $n > 1$.

We first show that $n \leq l_1$. First observe that, by (5) and $n > 1$, $s'(N_P)$ contains a seed pair $(a, f_{w_j}[L_1, P](a))$ 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 $(a, b_1), \dots, (a, b_n) \in s'(N_P)$ and $(a, b_1), \dots, (a, b_n)$ are triggered by $a \in s'(L_1)$. Therefore, either $(a, b_1), \dots, (a, b_n)$ are all pairs triggered by $a \in s'(L_1)$, in which case $n = l_1$, or $(a, b_1), \dots, (a, b_n), (a, b_{n+1}), \dots, (a, b_{l_1})$, 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 1.3.2: Σ defines subroles for P .

Since Σ satisfies the role hierarchy restriction and defines subroles for P , then Σ has no constraint of the form $e \sqsubseteq \neg(\geq 1 P)$ or of the form $e \sqsubseteq \neg(\geq 1 P^-)$. The proof of this case is a variation of that of Case 1.3.1.

Case 1.4: e is of the form $(\geq n P^-)$.

The proof of this case is entirely similar to that of Case 1.3.

Case 2: N is a \perp -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 four cases to consider.

Case 2.1: e is an atomic concept C .

Then, by Def. 1.8(b), we trivially have that $s(C) = s'(N) = \emptyset$.

Case 2.2: N is an atomic node P .

Then, by Def. 1.8(c), 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 \perp -node

Then, by Def. 1.7(c) and Def. 1.8(c), $s(P) = s'(N_P) = \emptyset$. Hence, $s((\geq n p)) = \emptyset$.

Case 2.1.2.2: N_P is not a \perp -node.

By Prop. 3(vi-b), N_P cannot be a \top -node. Then, Def. 1.7(b) 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 $(a, b_1), \dots, (a, b_n)$ 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 $(g_{i_0}[L_0, P](u), u)$, 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 $(a, f_{w_j}[L_1, P](a))$, 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 $(a, b_1), \dots, (a, b_n) \in s'(N_P)$ and $(a, b_1), \dots, (a, b_n)$ are triggered by $a \in s'(L_1)$. Therefore, either $(a, b_1), \dots, (a, b_n)$ are all pairs triggered by $a \in s'(L_1)$, in which case $n = l_1$, or $(a, b_1), \dots, (a, b_n), (a, b_{n+1}), \dots, (a, b_{l_1})$, 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 $s'(N) = \emptyset$.

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

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

Therefore, we established, in all cases, that Lemma 1.1(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 \perp -node. By Prop 3(vi-a), \perp -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 1.1(i), $t \in s'(M)$. That is, $t \in s'(M) \cap s'(N)$.

Note that M and N are dual nodes since M is labeled with e and N is labeled with $\neg e$. Therefore, since N is neither a \perp -node nor a \top -node, M is also neither a \top -node nor a \perp -node, by definition of \top -node.

Since Σ satisfies the role hierarchy restriction, there are two cases to consider.

Case 1.1.1: $\neg e$ is not of the form $\neg(\geq n P)$ or $\neg(\geq n P^-)$.

Then, by Def. 1.7(a), $t \in s'(N)$ iff t is a seed term of a node J and there is a path from J to K . Furthermore, by Prop. 5(ii), there is a seed node K such that $K \rightarrow M$ and $K \rightarrow N$ and $t \in s'(K)$. But this is impossible. 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 \perp -node. Hence, by Def. 1.7(c), $s'(K) = \emptyset$, which implies that $t \notin s'(K)$. Therefore, we established that, for all terms t , if $t \in s'(N)$ then $t \in s(\neg e)$.

Case 1.1.2: $\neg e$ is of the form $\neg(\geq n P)$ or $\neg(\geq n P^-)$.

Case 1.1.2.1: Σ defines no subroles for P .

Follows as in Case 1.1.1, again using Def. 1.7(a) and Prop. 5(ii).

Case 1.1.2.2: Σ defines subroles for P .

Since Σ satisfies the role hierarchy restriction, Σ has no constraint of the form $h \sqsubseteq \neg(\geq n P)$ or of the form $h \sqsubseteq \neg(\geq n P^-)$. Then, by Prop. 3(vii), for any node K , if $K \rightarrow N$, then K is labeled only with negative concept expressions. Therefore, there

could be no seed node K such that $K \rightarrow N$. Hence, by Def. 1.7(a), there is no term t such that $s'(N) = \emptyset$, which contradicts the assumption that $t \in s'(N)$.

Therefore, in all cases, 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. 1.7(b).

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

Case 2: N is a \top -node.

Let \bar{N} be the dual node of N . Since N is a \top -node, we have that \bar{N} is a \perp -node. Furthermore, since $\neg e$ labels N , e labels \bar{N} . Since e is a positive expression, by Lemma 1.1(i), $s'(\bar{N}) = s(e) = \emptyset$.

Case 2.1: N is a concept expression node.

By Def. 1.7(d) 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. 1.7(e) 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 1.1(ii) holds.

We are now ready to state the first result of the chapter.

Theorem 1.1. *Assume that Σ satisfies the role hierarchy restriction. 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 . Then, N is a \perp -node iff $s(e) = \emptyset$.
- (iii) Let e be a minCardinality constraint of the form $(\geq k P)$, with $k > 1$. Assume that $G(\Sigma, \Omega)$ has a node labeled with e . Then, N is a \perp -node iff $s(e) = \emptyset$.
- (iv) Let P be an atomic role. Let N be the node of $G(\Sigma, \Omega)$ labeled with P . Then, N is a \perp -node iff $s(P) = \emptyset$.

Proof Sketch 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 1.1(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 1.1(i), $s'(M) = s(e)$. and, by Lemma 1.1(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 $e \sqsubseteq f \in \Sigma$, 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. 1.1, $G(\Sigma, \Omega)$ always has a node N labeled with e . Since e is positive, by Lemma 1.1(i), $s(e) = s'(N)$.

Assume that N is a \perp -node. Then, by Lemma 1.1(i) and Def. 1.7(c), $s(e) = s'(N) = \emptyset$.

Assume that N is not a \perp -node. Note that N cannot be a \top -node, since N is labeled with the positive expression e . Then, N is neither a \perp -node nor a \top -node. By Def. 1.6(ii) and Def. 1.7(a), the seed term $c[N]$ of N is such that $c[N] \in s'(N)$. Hence, trivially, $s(e) = s'(N) \neq \emptyset$.

(iii)(iv) Follows as for (ii).

Based on Theorem 1.1, we can then create a simple procedure to test strict satisfiability, which has polynomial time complexity on the size of Σ :

SAT(Σ)

input: a set Σ of extralite constraints that satisfies the role hierarchy restriction.

output: “YES - Σ is strictly satisfiable”

“NO - Σ is not strictly satisfiable”

- 1) Normalize the constraints in Σ , creating a set Σ' .
- 2) Construct the constraint graph $G(\Sigma')$ that represents Σ' .
- 3) **If** $G(\Sigma')$ has no \perp -node labeled with an atomic concept or an atomic role,

then return “YES - Σ is strictly satisfiable”;

else return “NO - Σ is not strictly satisfiable”.

From Theorem 1.1, we can also prove that:

Theorem 1.2. *Assume that Σ satisfies the role hierarchy restriction. Let σ be a normalized extralite constraint. Assume that σ is of the form $e \sqsubseteq f$ and let $\Omega = \{e, f\}$. Then, $\Sigma \models \sigma$ iff one of the following conditions holds:*

- (i) *The node of $G(\Sigma, \Omega)$ labeled with e is a \perp -node; or*
- (ii) *The node of $G(\Sigma, \Omega)$ labeled with f is a \top -node; or*
- (iii) *There is a path in $G(\Sigma, \Omega)$ from the node labeled with e to the node labeled with f .*

Proof Sketch Let Σ be a set of normalized constraints. Assume that Σ satisfies the role hierarchy restriction. 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.

(\Leftarrow) Follows directly from Prop. 4.

(\Rightarrow) 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 \perp -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 $e \not\rightarrow f$ 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 two 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 1.1, s is a model of Σ . We show that $s \not\models e \sqsubseteq f$.

Case 1.1: N is a \perp -node.

Since N is a \perp -node, by Prop. 4(iii), we have that $\Sigma \models f \sqsubseteq \perp$, 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 \perp -node. Hence, we have that $s(e) \neq \emptyset$, by Theorem 1.1(ii), (iii) and (iv). Hence, we trivially have that $s \not\models e \sqsubseteq f$.

Case 1.2: N is not a \perp -node.

Observe that M and N are neither a \perp -node nor a \top -node. By assumption of the case and by (4), N is neither a \perp -node nor a \top -node. Now, by (3), M is not a \perp -node.

Furthermore, by Prop. 3(iv-b), since M is labeled with a positive expression e , M cannot be a \top -node.

By Lemma 1.1(i), since e is positive by assumption, by Def. 1.6(ii), (iii) and (iv), and by Def. 1.7(a) and (b), since M is neither a \perp -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 \perp -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 \perp -node nor a \top -node, $(t, u) \in s'(K)$ iff there is a path from M to K

By (5), we have $e \not\rightarrow f$. Furthermore, N is neither a \perp -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 1.1(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), $s(e) \not\subseteq s(f)$, that is, $s \not\models e \sqsubseteq f$, 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 1.1(i), s is a model of Σ . We show that $s \not\models e \sqsubseteq f$.

By Prop. 4(i) and (ii), and since s is a model of Σ , we have that $s \models e \equiv g$, if e and g label the same node, and $s \models e \sqsubseteq g$, otherwise. Hence, we have that $s \not\models e \sqsubseteq \neg g$. Now, since f is $\neg g$, we have $s \not\models e \sqsubseteq f$, as desired.

Case 2.2: $e \not\rightarrow g$.

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 1.1(i), r is a model of Φ . We show that $r \not\models e \sqsubseteq f$.

We first observe that

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

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 \perp -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 \bar{N} , the dual node of N in $G(\Sigma, \Omega)$. Then, $g \rightarrow h$ and $g \rightarrow \neg h$ implies that \bar{N} is a \perp -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 \perp -node.

By Theorem 1.1(i), r is a model of Φ . Furthermore, by Theorem 1.1(ii) and (iv), and since K is not a \perp -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.

Based on Theorem 1.2, we can then create a simple procedure to test logical implication:

IMPLIES ($\Sigma, e \sqsubseteq f$)

input: a set Σ of constraints satisfies the role hierarchy restriction,
and a constraint $e \sqsubseteq f$

output: “YES - Σ logically implies $e \sqsubseteq f$ ”
“NO - Σ does not logically imply $e \sqsubseteq f$ ”

- 1) Normalize the constraints in Σ , creating a set Σ' .
- 2) Normalize $e \sqsubseteq f$, creating a normalized constraint $e' \sqsubseteq f'$.
- 3) Construct $G(\Sigma', \{e', f'\})$.
- 4) **If** the node of $G(\Sigma', \{e', f'\})$ labeled with e' is a \perp -node, or the node of $G(\Sigma', \{e', f'\})$ labeled with f' is a \top -node, or there is a path in $G(\Sigma', \{e', f'\})$ from the node labeled with e' to the node labeled with f' ,
then return “YES - Σ logically implies $e \sqsubseteq f$ ”;
else return “NO - Σ does not logically imply $e \sqsubseteq f$ ”.

Note that **IMPLIES** has polynomial time complexity on the size of $\Sigma \cup \{e \sqsubseteq f\}$.

1.4 Examples

1.4.1 Examples of Extralite Schemas

In this section, we introduce examples of concrete, albeit simple extralite schemas with role hierarchies to illustrate how to capture commonly used ER model and UML constructs as extralite constraints.

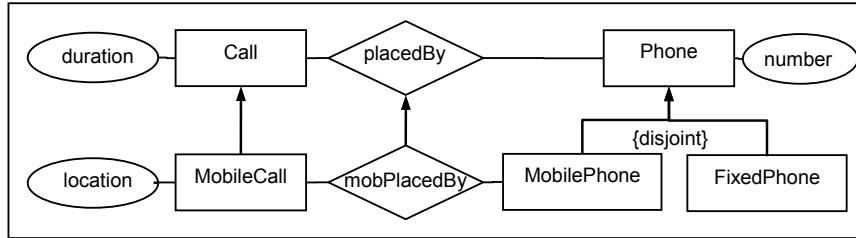


Fig. 1.1 ER diagram of the PhoneCompany1 schema (without cardinalities).

Example 1: Figure 1.1 shows the ER diagram of the PhoneCompany1 schema. Figure 1.2 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 1.2 indicates that:

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

Fig. 1.2 Formal definition of the constraints of the PhoneCompany1 schema.

- 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 MobilePhone

The second column of Figure 1.2 shows the cardinalities of the PhoneCompany1 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. MobilePhone, which need not be explicitly declared)

The third column of Figure 1.2 indicates that

- MobilePhone and FixedPhone are subsets of Phone
- MobilePhone and FixedPhone are disjoint
- 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 1.2.1.

Example 2: Fig. 1.3 shows the ER diagram of the PhoneCompany2 schema, and Figure 1.4 formalizes the constraints, following the same organization as that in Fig. 1.2). 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 MobilePhone
- the constraints of the schema imply that PlacedBy and mobPlacedBy are disjoint roles, by the disjunction-transfer rule introduced at the end of Section 1.2.1 (see also Example 3(b)).

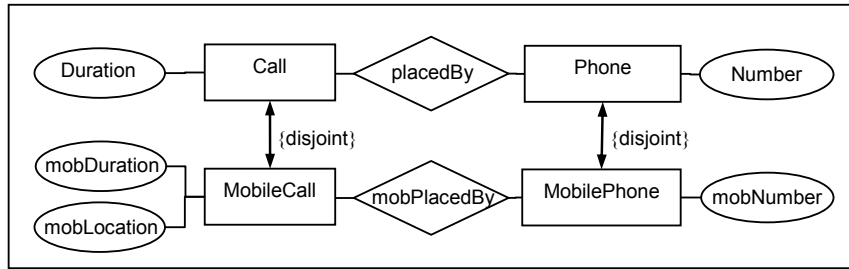


Fig. 1.3 ER diagram of the PhoneCompany2 schema (without card. and disjunctions).

$\exists \text{Number} \sqsubseteq \text{Phone}$	$\text{Phone} \sqsubseteq (\leq 1 \text{ number})$	MobilePhone $ \text{Phone}$ MobileCall $ \text{Call}$
$\exists \text{Number} \bar{\sqsubseteq} \text{String}$	$\text{Phone} \sqsubseteq (\geq 1 \text{ number})$	
$\exists \text{Duration} \sqsubseteq \text{Call}$	$\text{Call} \sqsubseteq (\leq 1 \text{ Duration})$	
$\exists \text{Duration} \bar{\sqsubseteq} \text{String}$	$\text{Call} \sqsubseteq (\geq 1 \text{ Duration})$	
$\exists \text{placedBy} \sqsubseteq \text{Call}$	$\text{Call} \sqsubseteq (\leq 1 \text{ placedBy})$	
$\exists \text{placedBy} \bar{\sqsubseteq} \text{Phone}$	$\text{Call} \sqsubseteq (\geq 1 \text{ placedBy})$	
$\exists \text{mobDuration} \sqsubseteq \text{MobileCall}$	$\text{MobileCall} \sqsubseteq (\leq 1 \text{ mobDuration})$	
$\exists \text{mobDuration} \bar{\sqsubseteq} \text{String}$	$\text{MobileCall} \sqsubseteq (\geq 1 \text{ mobDuration})$	
$\exists \text{mobLocation} \sqsubseteq \text{MobileCall}$	$\text{MobileCall} \sqsubseteq (\leq 1 \text{ mobLocation})$	
$\exists \text{mobLocation} \bar{\sqsubseteq} \text{String}$	$\text{MobileCall} \sqsubseteq (\geq 1 \text{ mobLocation})$	
$\exists \text{mobPlacedBy} \sqsubseteq \text{MobileCall}$	$\text{MobileCall} \sqsubseteq (\leq 1 \text{ mobPlacedBy})$	
$\exists \text{mobPlacedBy} \bar{\sqsubseteq} \text{MobilePhone}$	$\text{MobileCall} \sqsubseteq (\geq 1 \text{ mobPlacedBy})$	

Fig. 1.4 Formal definition of the constraints of the PhoneCompany2 schema.

1.4.2 Examples of Representation Graphs

In this section, we illustrate representation graphs and their uses in the decision procedures of Section 1.3.3.

Example 3: 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) $\exists \text{ placedBy} \sqsubseteq \text{Call}$ normalized as: $(\geq 1 \text{ placedBy}) \sqsubseteq \text{Call}$
- (2) $\exists \text{ placedBy}^- \sqsubseteq \text{Phone}$ normalized as: $(\geq 1 \text{ placedBy}^-) \sqsubseteq \text{Phone}$
- (3) $\exists \text{ mobPlacedBy} \sqsubseteq \text{MobileCall}$ normalized as: $(\geq 1 \text{ mobPlacedBy}) \sqsubseteq \text{MobileCall}$
- (4) $\exists \text{ mobPlacedBy}^- \sqsubseteq \text{MobilePhone}$ normalized as: $(\geq 1 \text{ mobPlacedBy}^-) \sqsubseteq \text{MobilePhone}$
- (5) $\text{Call} \sqsubseteq (\leq 1 \text{ placedBy})$ normalized as: $\text{Call} \sqsubseteq \neg (\geq 2 \text{ placedBy})$
- (6) $\text{MobilePhone} | \text{Phone}$ normalized as: $\text{MobilePhone} \sqsubseteq \neg \text{Phone}$
- (7) $\text{MobileCall} | \text{Call}$ normalized as: $\text{MobileCall} \sqsubseteq \neg \text{Call}$

Figure 1.5 depicts $G(\Sigma)$, the graph that represents Σ , using the normalized constraints. In special, the dotted arcs highlight the paths that correspond to the conditions of Stage 4 of Definition 1.1, and the dashed arcs indicate the arcs that Stage 4 of Definition 1.1 requires to exist, which capture the derived constraint:

- (8) $\text{mobPlacedBy} | \text{placedBy}$ normalized as: $\text{mobPlacedBy} \sqsubseteq \neg \text{placedBy}$

Since $G(\Sigma)$ has no \perp -node labeled with an atomic concept or an atomic role, Σ is strictly satisfiable, by Theorem 1.1. However note that $(\geq 2 \text{ placedBy})$ is a \perp -node of $G(\Sigma)$.

Example 4: Let Σ be the following subset of the constraints of the PhoneCompany1 schema, introduced in Example 1 (again we do not consider all constraints to reduce the size of the example):

- (1) $\exists \text{ placedBy} \sqsubseteq \text{Call}$ normalized as: $(\geq 1 \text{ placedBy}) \sqsubseteq \text{Call}$
- (2) $\exists \text{ placedBy}^- \sqsubseteq \text{Phone}$ normalized as: $(\geq 1 \text{ placedBy}^-) \sqsubseteq \text{Phone}$
- (3) $\text{Call} \sqsubseteq (\leq 1 \text{ placedBy})$ normalized as: $\text{Call} \sqsubseteq \neg (\geq 2 \text{ placedBy})$
- (4) $\text{MobileCall} \sqsubseteq \text{Call}$
- (5) $\text{mobPlacedBy} \sqsubseteq \text{placedBy}$

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

- (6) $\text{ConferenceCall} \sqsubseteq \text{Call}$
- (7) $\text{ConferenceCall} \sqsubseteq (\geq 2 \text{ placedBy})$

These new constraints intuitively say that conference calls are calls placed by at least two phones. However, this apparently correct modification applied to the PhoneCompany1 schema forces ConferenceCall to always have an empty interpretation. Example 5(c) will also show that (6) is actually redundant.

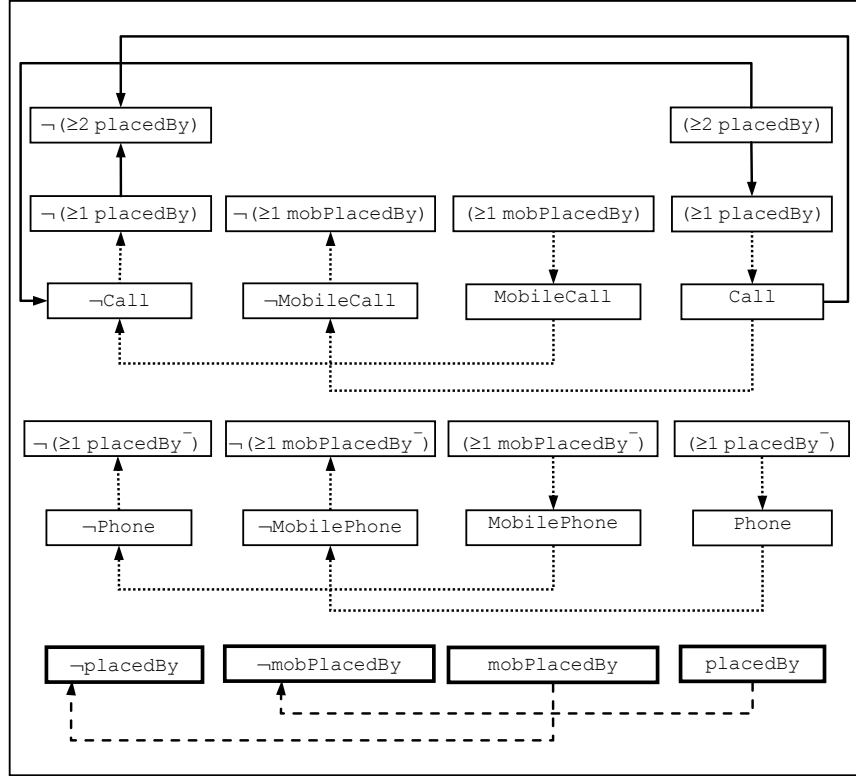


Fig. 1.5 The graph representing Σ .

Figure 1.6 depicts $G(\Psi)$, the graph that represents Ψ , using the normalized constraints. Note that there is a path from ConferenceCall to \neg ConferenceCall. Also note that there are paths from the node labeled with ConferenceCall to nodes labeled with Call and \neg Call, as well as to nodes labeled with $(\geq 2$ placedBy) and $\neg(\geq 2$ placedBy) and nodes labeled with $(\geq 1$ placedBy) and $\neg(\geq 1$ placedBy). The arcs of all such paths are shown in dashed lines in Figure 1.6.

Hence, the node labeled with ConferenceCall is a \perp -node of $G(\Psi)$, which implies that Ψ is not strictly satisfiable, by Theorem 1.1. Any interpretation s that satisfies Ψ is such that $s(\text{ConferenceCall}) \subseteq s(\neg\text{ConferenceCall})$ holds, which implies that $s(\text{ConferenceCall}) = \emptyset$.

Example 5: This example illustrates the three cases of Theorem 1.2. Let Ψ be the set of constraints considered in Example 4 and $G(\Psi)$ be the graph representing Ψ , shown in Figure 1.6.

- (a) Let σ be the constraint $\text{ConferenceCall} \sqsubseteq (\geq 1 \text{ placedBy}^-)$. Note that σ is of the form $e \sqsubseteq f$, where $e = \text{ConferenceCall}$ and $f = (\geq 1$

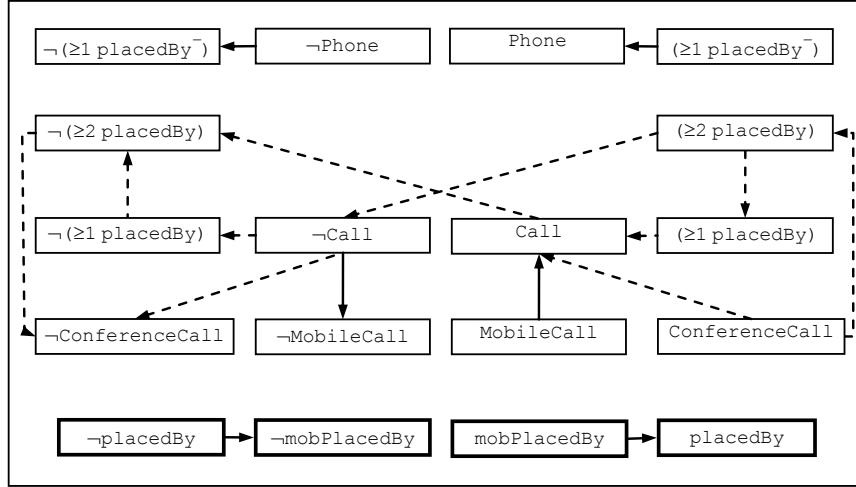


Fig. 1.6 The graph representing Ψ .

placedBy^-). Then, $G(\Psi, \{e, f\})$ is equal to $G(\Psi)$, since $G(\Psi)$ already contains nodes labeled with ConferenceCall and with $(\geq 1 \text{ placedBy}^-)$. Recall from Example 4 that the node labeled with ConferenceCall is a \perp -node of $G(\Psi)$, and hence of $G(\Psi, \{e, f\})$. Then, by Theorem 1.2(i), we trivially have

$$\Psi \models \text{ConferenceCall} \sqsubseteq (\geq 1 \text{ placedBy}^-)$$

- (b) Let σ be the constraint $\text{Phone} \sqsubseteq \neg\text{ConferenceCall}$. Note that σ is of the form $e \sqsubseteq f$, where $e = \text{Phone}$ and $f = \neg\text{ConferenceCall}$. Since the node labeled with ConferenceCall is a \perp -node of $G(\Psi, \{e, f\})$, the node labeled with $\neg\text{ConferenceCall}$ is a \top -node of $G(\Psi, \{e, f\})$. Hence, by Theorem 1.2(ii), we have

$$\Psi \models \text{Phone} \sqsubseteq \neg\text{ConferenceCall}$$

- (c) Let σ be the constraint $\text{ConferenceCall} \sqsubseteq \text{Call}$. Note that σ is of the form $e \sqsubseteq f$, where $e = \text{ConferenceCall}$ and $f = \text{Call}$. Since there is a path in $G(\Psi \cup \{e, f\})$ from the node labeled with ConferenceCall to the node labeled with Call passing through the nodes labeled with $(\geq 2 \text{ placedBy})$ and $(\geq 1 \text{ placedBy})$, by Theorem 1.2(iii), we have

$$\Sigma \models \text{ConferenceCall} \sqsubseteq \text{Call}$$

Hence, constraint (6) in Example 4 is actually redundant.

1.4.3 Two Applications of Representation Graphs

In this section, we briefly discuss two applications of representation graphs. The first application explores how to use representation graphs to suggest changes to a strictly unsatisfiable schema until it become strictly satisfiable.

Example 6: Consider again the modified set of constraints Ψ of Example 4. To simplify the discussion, given an expression e , when we refer to node e , we mean the node labeled with e . Recall that Fig. 1.6 shows the graph representing Ψ . Also recall that the sources of the strict unsatisfiability of Ψ are the paths shown in dashed lines in Fig. 1.6.

Note that the arc from node $(\geq 2 \text{ placedBy})$ to node $(\geq 1 \text{ placedBy})$ is in $G(\Psi)$ by virtue of the semantics of these minCardinality expressions and, hence, it cannot be dropped (and likewise for the arc from $\neg(\geq 1 \text{ placedBy})$ to $\neg(\geq 2 \text{ placedBy})$). Therefore, the simplest ways to break the faulty paths are:

- (a) Drop the arc from node `ConferenceCall` to node $(\geq 2 \text{ placedBy})$ (and consequently the dual arc from node $\neg(\geq 2 \text{ placedBy})$ to node `¬ConferenceCall`).
- (b) Drop the arc from node `Call` to node $\neg(\geq 2 \text{ placedBy})$ (and consequently the dual arc from node $(\geq 2 \text{ placedBy})$ to node `¬Call`).

Note that the strict satisfiability of the schema would not be restored by dropping just the arc from node `ConferenceCall` to node `Call` (and its dual arc), or the arc from node $(\geq 1 \text{ placedBy})$ to node `Call` (and its dual arc).

The representation graph is neutral as to which arc to drop. Thus, we must base our decision on some schema redesign heuristics. Both options are viable, but they obviously alter the semantics of the schema. Option (a) amounts to dropping constraint (7) of Example 4, which requires `ConferenceCall` to be a subset of $(\geq 2 \text{ placedBy})$. This option is not reasonable since it obliterates the very purpose of the redesign step, which was to model conference calls as calls placed by at least two phones. Option (b) means dropping constraint (3), which would alter the semantics of `Call`. However, it is consistent with the purpose of the redesign step and is better than Option (a).

A third option would be to create a second specialization of `Call`, say, `nonConferenceCall`, and alter constraint (3) of Example 4 accordingly. The constraints of Example 4 would now include:

- (8) $(\geq 1 \text{ placedBy}) \sqsubseteq \text{Call}$
- (9) $(\geq 1 \text{ placedBy}^-) \sqsubseteq \text{Phone}$
- (10) `MobileCall` \sqsubseteq `Call`
- (11) `mobPlacedBy` \sqsubseteq `placedBy`
- (12) `ConferenceCall` \sqsubseteq `Call`
- (13) `ConferenceCall` \sqsubseteq $(\geq 2 \text{ placedBy})$
- (14) `nonConferenceCall` \sqsubseteq `Call`
- (15) `nonConferenceCall` \sqsubseteq $\neg(\geq 2 \text{ placedBy})$

In view of (13) and (15), note that it would be redundant to include a constraint to force `ConferenceCall` and `nonConferenceCall` to be mutually exclusive. From the point of view of schema redesign practice, this would be the best alternative since it retains the information that there are calls with just one originating place.

The second application we briefly discuss is how to integrate two schemas, S_1 and S_2 , which use the same concepts and properties, but differ on their constraints [8]. More precisely, denote by $Th(\sigma)$ the set of all constraints which are logical consequences of a set of constraints σ . Let Σ_1 and Σ_2 be the sets of (normalized) constraints of two schemas, S_1 and S_2 , respectively. The goal now is to come up with a set of constraints Γ that conveys the common semantics of S_1 and S_2 , that is, a set of constraints Γ such that $Th(\Gamma) = Th(\Sigma_1) \cap Th(\Sigma_2)$.

Example 7: Let $G(\Sigma_1)$ and $G(\Sigma_2)$ be the graphs that represent the sets of constraints Σ_1 and Σ_2 . Denote their transitive closures by $G^*(\Sigma_1)$ and $G^*(\Sigma_2)$. Based on Theorem 1.2, we illustrate in this example how to use $G^*(\Sigma_1)$ and $G^*(\Sigma_2)$ to construct a set of constraints Γ such that $Th(\Gamma) = Th(\Sigma_1) \cap Th(\Sigma_2)$.

Suppose that Σ_1 is the following subset of the normalized constraints of the Phone-Company1 schema of Example 1 (again we do not consider all constraints to reduce the size of the example; we also abbreviate the names of the atomic concepts and roles in an obvious way, i.e., `pc` stands for `placedBy`, etc.):

- (1) $(\geq 1 \text{ pc}) \sqsubseteq C$
- (2) $(\geq 1 \text{ pc}^-) \sqsubseteq P$
- (3) $C \sqsubseteq \neg(\geq 2 \text{ pc})$
- (4) $(\geq 1 \text{ mpc}) \sqsubseteq MC$
- (5) $(\geq 1 \text{ mpc}^-) \sqsubseteq MP$
- (6) $MC \sqsubseteq C$
- (7) $MP \sqsubseteq P$
- (8) $\text{mpc} \sqsubseteq \text{pc}$

Suppose that Σ_2 is the following subset of normalized constraints of the Phone-Company2 schema of Example 2:

- (9) $(\geq 1 \text{ pc}) \sqsubseteq C$
- (10) $(\geq 1 \text{ pc}^-) \sqsubseteq P$
- (11) $C \sqsubseteq \neg(\geq 2 \text{ pc})$
- (12) $(\geq 1 \text{ mpc}) \sqsubseteq MC$
- (13) $(\geq 1 \text{ mpc}^-) \sqsubseteq MP$
- (14) $MC \sqsubseteq \neg C$
- (15) $MP \sqsubseteq \neg P$

For $i = 1, 2$, let $G(\Sigma_i)$ be the graph that represents Σ_i (Figure 1.5 depicts $G(\Sigma_2)$). We systematically construct Γ such that $Th(\Gamma) = Th(\Sigma_1) \cap Th(\Sigma_2)$ as follows. Tables 1.1(a) and 1.1(b) show the arcs of $G^*(\Sigma_1)$ and $G^*(\Sigma_2)$. Note that a tabular presentation of the arcs, as opposed to a graphical representation, is much more convenient since we are working with transitive closures. For example, line 3 of Table 1.1(a) indicates that $G^*(\Sigma_1)$ has an arc from the node labeled with $(\geq 1 \text{ pc})$ to the nodes labeled with C and $\neg(\geq 2 \text{ pc})$.

Table 1.1 Construction of the set of constraints Γ that generates $\Sigma\Delta\Phi$.

	(a) $G^*(\Sigma_1)$		(b) $G^*(\Sigma_2)$		(c) $G^*(\Gamma)$	
1	$\neg MC$	$\neg(\geq 1 \text{ mpc})$	$\neg MC$	$\neg(\geq 1 \text{ mpc})$		
2	C	$\neg(\geq 2 \text{ pc})$	C	$\neg MC$ $\neg(\geq 1 \text{ mpc})$ $\neg(\geq 2 \text{ pc})$	C	$\neg(\geq 2 \text{ pc})$
3	$(\geq 1 \text{ pc})$	C $\neg(\geq 2 \text{ pc})$	$(\geq 1 \text{ p})$	C $\neg MC$ $\neg(\geq 1 \text{ mpc})$ $\neg(\geq 2 \text{ pc})$	$(\geq 1 \text{ pc})$	C $\neg(\geq 2 \text{ pc})$
4	$(\geq 2 \text{ pc})$	\perp	$(\geq 2 \text{ pc})$	\perp	$(\geq 2 \text{ pc})$	\perp
5	$\neg(\geq 1 \text{ pc})$	$\neg(\geq 2 \text{ pc})$	$\neg(\geq 1 \text{ pc})$	$\neg(\geq 2 \text{ pc})$		
6	$\neg C$	$\neg(\geq 1 \text{ pc})$ $\neg(\geq 2 \text{ pc})$ $\neg MC$ $\neg(\geq 1 \text{ mpc})$	$\neg C$	$\neg(\geq 1 \text{ pc})$ $\neg(\geq 2 \text{ pc})$		
7	MC	C $\neg(\geq 2 \text{ pc})$	MC	$\neg C$ $\neg(\geq 1 \text{ pc})$ $\neg(\geq 2 \text{ pc})$	MC	$\neg(\geq 2 \text{ pc})$
8	$(\geq 1 \text{ mpc})$	MC C $\neg(\geq 2 \text{ pc})$	$(\geq 1 \text{ mpc})$	MC $\neg C$ $\neg(\geq 1 \text{ pc})$ $\neg(\geq 2 \text{ pc})$	$(\geq 1 \text{ mpc})$	MC $\neg(\geq 2 \text{ pc})$
9	$\neg MP$	$\neg(\geq 1 \text{ mpc}^-)$	$\neg MP$	$\neg(\geq 1 \text{ mpc}^-)$		
10			P	$\neg MP$ $\neg(\geq 1 \text{ mpc}^-)$		
11	$(\geq 1 \text{ pc}^-)$	P	$(\geq 1 \text{ pc}^-)$	P $\neg MP$ $\neg(\geq 1 \text{ mpc}^-)$	$(\geq 1 \text{ pc}^-)$	P
12	$\neg P$	$\neg(\geq 1 \text{ pc}^-)$ $\neg MP$ $\neg(\geq 1 \text{ mpc}^-)$	$\neg P$	$\neg(\geq 1 \text{ pc}^-)$		
13	MP	P	MP	$\neg P$ $\neg(\geq 1 \text{ pc}^-)$		
14	$(\geq 1 \text{ mpc}^-)$	MP P	$(\geq 1 \text{ mpc}^-)$	MP $\neg P$ $\neg(\geq 1 \text{ pc}^-)$	$(\geq 1 \text{ mpc}^-)$	MP
15			pc	$\neg \text{mpc}$		
16			mpc	$\neg \text{pc}$		

In this specific example, Table 1.1(c) induces Γ as follows:

- Lines 10, 15 and 16 are discarded since they correspond to arcs in just $G^*(\Sigma_2)$.
- Lines 1, 5, 6, 9 and 12 are discarded since they have a negated expression on the left-hand side cell.
- Line 4 corresponds to a special case of a \perp -node (c.f. Theorem 1.2(i)).
- The other lines retain just the arcs that are simultaneously in $G^*(\Sigma_1)$ and $G^*(\Sigma_2)$.

Table 1.1 shows the final set of constraints in Γ :

- (16) $C \sqsubseteq \neg(\geq 2 \text{ pc})$ from line 2
- (17) $(\geq 1 \text{ pc}) \sqsubseteq C$ from line 3
- (18) $(\geq 1 \text{ pc}) \sqsubseteq \neg(\geq 2 \text{ pc})$ from line 3
- (19) $(\geq 2 \text{ pc}) \sqsubseteq \perp$ from line 4
- (20) $MC \sqsubseteq \neg(\geq 2 \text{ pc})$ from line 7
- (21) $(\geq 1 \text{ mpc}) \sqsubseteq MC$ from line 8
- (22) $(\geq 1 \text{ mpc}) \sqsubseteq \neg(\geq 2 \text{ pc})$ from line 8
- (23) $(\geq 1 \text{ pc}^-) \sqsubseteq P$ from line 11
- (24) $(\geq 1 \text{ mpc}^-) \sqsubseteq MP$ from line 14

Note that it is not entirely obvious that constraints (18), (19) and (22) are in $Th(\Sigma_1) \cap Th(\Sigma_2)$. We refer the reader to [8] for a detailed proof that this construction leads to a set of constraints Γ such that $Th(\Gamma) = Th(\Sigma_1) \cap Th(\Sigma_2)$. Roughly, it corresponds to the saturation strategy in binary resolution.

1.5 Conclusions

We first introduced extralite schemas with role hierarchies, which are sufficiently expressive to encode commonly used ER model and UML constructs, including relationship hierarchies. Then, we showed how to efficiently test strict satisfiability and logical implication for restricted extralite schemas with role hierarchies. The procedures have low time complexity, and they retain and explore the constraint structure, which is a useful feature for a number of problems, as pointed out in the introduction.

Finally, as future work, we plan to investigate the problem of efficiently testing extralite schemas with role hierarchies for finite satisfiability [11].

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

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The dl-lite family and relations. *J. Artif. Int. Res.* **36**, 1–69 (2009)
2. Aspövall, B., Plass, M.F., Tarjan, R.E.: A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.* **8**(3), 121–123 (1979)
3. Baader, F., Nutt, W.: The description logic handbook. chap. Basic description logics, pp. 43–95. Cambridge University Press, New York, NY, USA (2003)
4. Borgida, A., Brachman, R.J.: The description logic handbook. chap. Conceptual modeling with description logics, pp. 349–372. Cambridge University Press, New York, NY, USA (2003)
5. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rosati, R., Ruzzi, M.: Data integration through DL-Lite-A ontologies. In: *Proceedings of the Third International Workshop on Semantics in Data and Knowledge Bases (SDKB 2008)*, pp. 26–47 (2008)
6. Calvanese, D., Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Autom. Reason.* **39**, 385–429 (2007)
7. Casanova, M.A., Furtado, A.L., Macedo, J.A., Vidal, V.M.: Extralite schemas with role hierarchies. Tech. Rep. MCC09/10, Department of Informatics, PUC-Rio (2010)
8. Casanova, M.A., Lauschner, T., Leme, L.A.P.P., Breitman, K.K., Furtado, A.L., Vidal, V.M.: Revising the constraints of lightweight mediated schemas. *Data & Knowledge Engineering* **69**(12), 1274 – 1301 (2010). Special issue on 28th International Conference on Conceptual Modeling (ER 2009)
9. Hartmann, S., Link, S., Trinh, T.: Constraint acquisition for entity-relationship models. *Data Knowl. Eng.* **68**, 1128–1155 (2009)
10. Lauschner, T., Casanova, M.A., Vidal, V.M.P., de Macêdo, J.A.F.: Efficient decision procedures for query containment and related problems. In: A. Brayner (ed.) *XXIV Simpósio Brasileiro de Banco de Dados*, 05-09 de Outubro, Fortaleza, Ceará, Brasil, Anais, pp. 1–15 (2009)
11. Rosati, R.: Finite model reasoning in DL-Lite. In: *Proceedings of the 5th European semantic web conference on The semantic web: research and applications, ESWC’08*, pp. 215–229. Springer-Verlag, Berlin, Heidelberg (2008)

Index

- Alphabet
 - Attributive Language, 4
 - At-Most Restriction, 4
 - Atomic Concepts, 4
 - Atomic Negation, 4
 - Atomic Roles, 4
 - Attributive Language, 4
 - Alphabet, 4
 - At-Most Restriction, 4
 - Atomic Negation, 4
 - Atomic Role, 4
 - Axiom
 - Equality, 5
 - Inclusion, 5
 - Bottom Concept, 4
 - Concept Description, 4
 - Interpretation, 4
 - Domain, 4
 - Equality, 5
 - Full Existential Quantification, 4
 - Inclusion, 5
 - Individual
 - Attributive Language, 4
 - Interpretation, 4
 - Domain, 4
 - Individual, 4
 - Interpretation function, 4
 - Satisfies
 - Axiom, 5
 - Terminological Axiom, *see* Axiom
 - Universal Concept, 4
- Axiom
 - Equality, 5
 - Inclusion, 5
 - Bottom Concept, 4
 - Concept Description, 4
 - Interpretation, 4
 - Domain, 4
 - Equality, 5
 - Full Existential Quantification, 4
 - Inclusion, 5
 - Individual
 - Attributive Language, 4
 - Interpretation, 4
 - Domain, 4
 - Individual, 4
 - Interpretation function, 4
 - Satisfies
 - Axiom, 5
 - Terminological Axiom, *see* Axiom
 - Universal Concept, 4