On the Finite Controllability of Conjunctive Query Answering in Databases under Open-World Assumption

Riccardo Rosati

Dipartimento di Informatica e Sistemistica
Sapienza Università di Roma
Via Ariosto 25, 00185 Roma, Italy
rosati@dis.uniroma1.it

Abstract

In this paper we study queries over relational databases with integrity constraints (ICs). The main problem we analyze is *OWA query answering*, i.e., query answering over a database with ICs under open-world assumption. The kinds of ICs that we consider are inclusion dependencies and functional dependencies, in particular key dependencies; the query languages we consider are conjunctive queries and unions of conjunctive queries. We present results about the decidability of OWA query answering under ICs. In particular, we study OWA query answering both over finite databases and over unrestricted databases, and identify the cases in which such a problem is *finitely controllable*, i.e., when OWA query answering over finite databases coincides with OWA query answering over unrestricted databases. Moreover, we are able to easily turn the above results into new results about implication of ICs and query containment under ICs, due to the deep relationship between OWA query answering and these two classical problems in database theory. In particular, we close two long-standing open problems in query containment, since we prove finite controllability of containment of conjunctive queries both under arbitrary inclusion dependencies and under key and foreign key dependencies. The results of our investigation are very relevant in many research areas which have recently dealt with databases under an incomplete information assumption: e.g., data integration, data exchange, view-based information access, ontology-based information systems, and peer data management systems.

Preprint submitted to Journal of Computer and System Sciences *April 12, 2010*

1. Introduction

The problem. In this paper we study queries and integrity constraints (ICs) over relational databases. We consider the most common forms of relational queries, i.e., *conjunctive queries* (CQs) and *unions of conjunctive queries* (UCQs), and the most important forms of relational integrity constraints, i.e., *functional dependencies* (FDs), and in particular *key dependencies* (KDs), and *inclusion dependencies* (IDs). The main problem studied in this paper is *OWA query answering*, which corresponds to query answering over a database with integrity constraints under open-world assumption (OWA), i.e., under the assumption that the facts stored in the database are only an *incomplete* specification of the data [1, 2, 3]. Under this assumption, the actual meaning of a database $\mathcal{D}$ with integrity constraints $\mathcal{C}$ is represented by the set of all databases $\mathcal{B}$ such that $\mathcal{B}$ contains all the facts in $\mathcal{D}$ and $\mathcal{B}$ satisfies the integrity constraints in $\mathcal{C}$.

The significance of the OWA query answering problem is witnessed by the following considerations:

- OWA query answering is deeply related to several classical problems in database theory, in particular: *implication of integrity constraints* (a.k.a. database dependencies) [4]; *OWA-consistency*, i.e., consistency of a database instance with respect to a set of ICs under open-world assumption [1, 2]; and *query containment under integrity constraints* [5].
- Many research areas are currently studying problems that are tightly related with databases under incomplete information: e.g., data integration [6], data exchange [7], view-based information access [8], ontology-based information systems [9], mapping composition [10], consistent query answering [11], and peer data management systems [12]. In all such scenarios, the problem of OWA query answering (or problems very closely related to it) is studied under various forms. Therefore, results about OWA query answering are in principle very relevant in all these areas.

Results for OWA query answering. We develop our analysis both under the assumption that a database must be a *finite* structure, and under the assumption of *unrestricted* databases (i.e., a database may be infinite). In this respect, we identify the cases when the OWA query answering problem is *finitely controllable*, i.e., when OWA query answering over finite databases coincides with OWA query answering over unrestricted databases.

We present a set of decidability, complexity, and finite controllability results for OWA query answering under ICs. More precisely: (i) we identify the cases in which such a problem is finitely controllable; (ii) we identify the decidability/undecidability frontier for the query languages and the ICs above mentioned; (iii) for the decidable cases, we study the computational complexity of OWA query answering (both data complexity and combined complexity).

In a nutshell, our results provide a clear picture of the frontier between decidability and undecidability of OWA-answering of conjunctive queries (and unions of conjunctive queries) under key dependencies and inclusion dependencies. In particular, our results show that:

- OWA-answering of unions of conjunctive queries under IDs (Theorem 2) and under keys and foreign keys (Theorem 3) is finitely controllable;
- OWA-answering of conjunctive queries under so-called *non-conflicting* KDs and IDs is not finitely controllable and is undecidable over finite databases (Theorems 4 and 5).

Results for query containment. Moreover, we are able to easily turn the above results into new results about implication of ICs and query containment under ICs, due to the deep relationship between OWA query answering and these two classical problems in database theory. In particular, we close two long-standing open problems in query containment which date back to the mid-80's [5]:

- we prove finite controllability of containment of conjunctive queries under arbitrary inclusion dependencies (Corollary 1);
- we prove finite controllability of containment of conjunctive queries under keys and foreign keys (Corollary 2).

Relevance of our results. Besides its theoretical interest, we believe that the analysis presented in this paper is very relevant in all the above mentioned areas dealing with data under open-world assumption, i.e., when it is not possible/realistic to assume that the database constitutes a complete specification of the information of interest.

Structure of the paper. In the next section, we present some preliminary definitions and define the problem studied. In Section 3 we present our main result, i.e., the proof of finite controllability of OWA-answering CQs under IDs. Then, in Section 4 we present our further results for OWA-answering of UCQs. In Section 5 we describe the relationship between OWA-answering and query containment, and point out two notable consequences of our results for OWA-answering in query containment. Finally, we analyze related work in Section 6 and conclude in Section 7.

The results presented in this paper appeared in a preliminary form in [13].

2. Definitions

We start from: (i) a relational signature, i.e., a set of relation symbols in which each relation is associated with an arity, i.e., a positive integer; (ii) a countably infinite alphabet of constant symbols; (iii) an alphabet of variable symbols. An *attribute* of a relation r is an integer b such that $1 \leq b \leq n$, where n is the arity of r.

A *fact* is an expression of the form $r(\mathbf{c})$, where r is a relation symbol and $\mathbf{c}$ is a tuple of constants. An *atom* is an expression of the form $r(\mathbf{v})$, where r is a relation symbol and $\mathbf{v}$ is a tuple of constants and/or variables. A *substitution* is a function mapping variables to constants.

A *database instance* (or simply *database*) is a (possibly numerable infinite) set of facts.

Given an n-tuple $\mathbf{v} = \langle v_1, \ldots, v_n \rangle$ and a sequence of integers $A = i_1, \ldots, i_k$ where $1 \leq i_j \leq n$ for each j, we denote by $\mathbf{v}[A]$ the projection of $\mathbf{v}$ over A, i.e., the k-tuple $\langle v_{i_1}, \ldots, v_{i_k} \rangle$.

2.1. Integrity constraints

Key dependencies. A key dependency (KD) is an expression of the form $key(r) = A$, where r is a relation symbol and A is a non-empty sequence of attributes, i.e., a sequence of integers ranging from 1 to the arity of r. The number of attributes in A is called the *arity* of the KD. We say that a set of KDs $\mathcal{K}$ is a set of *single* KDs if, for each relation r, there is at most one KD for r in $\mathcal{K}$. (We assume that a KD for a relation is always present, since in the case when there are no KDs for r we can consider the trivial KD $key(r) = U$ where U is the set of all attributes of r.) A database $\mathcal{D}$ satisfies a KD $key(r) = A$ if, for every pair of facts of the form $r(\mathbf{c})$ $r(\mathbf{c}')$ in $\mathcal{D}$, if $\mathbf{c}[A] = \mathbf{c}'[A]$ then $\mathbf{c} = \mathbf{c}'$.

Inclusion dependencies. An inclusion dependency (ID) is an expression of the form $r[A] \subseteq s[B]$, where r and s are relation symbols and A and B are sequences of attributes, i.e., sequences of integers ranging from 1 to the arity of the respective relations. The number of attributes in A (which is the same as the number of attributes in B) is called the *arity* of the ID. We do not allow multiple occurrences of the same attribute in A and in B. A database $\mathcal{D}$ satisfies an ID of the form $r[A] \subseteq s[B]$ if, for every fact in $\mathcal{D}$ of the form $r(\mathbf{c})$, there exists a fact in $\mathcal{D}$ of the form $s(\mathbf{c}')$ such that $\mathbf{c}[A] = \mathbf{c}'[B]$.

KDs and IDs are the main ICs studied in this paper. However, we will also mention functional dependencies. A *functional dependency* (FD) is an expression of the form $r : A \rightarrow b$ where r is a relation, A is a set of attributes of r and b is an attribute of r. A database $\mathcal{D}$ satisfies a FD $r : A \rightarrow b$ if, for every pair of facts of the form $r(\mathbf{c})$, $r(\mathbf{c}')$ in $\mathcal{D}$, if $\mathbf{c}[A] = \mathbf{c}'[A]$ then $\mathbf{c}[b] = \mathbf{c}'[b]$.

Classes of KDs and IDs. Given a set of KDs $\mathcal{K}$ and a set of IDs $\mathcal{I}$, we say that $\mathcal{I} \cup \mathcal{K}$ is a set of *non-conflicting* KDs and IDs if $\mathcal{K}$ is a set of single KDs and for each ID in I of the form $r[A] \subseteq s[B]$, B is not a *proper* superset of the key of s, i.e., if $\mathcal{K}$ contains the KD $key(s) = C$, then $B \not\supset C$.

Moreover, given a set of single KDs $\mathcal{K}$ and a set of IDs $\mathcal{I}$, we say that $\mathcal{I}$ is a set of *foreign keys* (FKs) for $\mathcal{K}$ if for each ID in $\mathcal{I}$ of the form $r[A] \subseteq s[B]$, B is a (not necessarily strict) subset of the key of s, i.e., if $\mathcal{K}$ contains the KD $key(s) = C$, then $B \subseteq C$.[1] Of course, if $\mathcal{I}$ is a set of FKs for $\mathcal{K}$ then $\mathcal{I} \cup \mathcal{K}$ is a set of non-conflicting KDs and IDs, but not vice versa.

Finally, given a set of ICs $\mathcal{C}$, we say that a database instance $\mathcal{B}$ satisfies $\mathcal{C}$ if $\mathcal{B}$ satisfies every IC in $\mathcal{C}$.

2.2. Queries

A *union of conjunctive queries* (UCQ) is an expression of the form

$$\{\mathbf{x} \mid conj_1(\mathbf{x}, \mathbf{c}) \vee \ldots \vee conj_m(\mathbf{x}, \mathbf{c})\} \qquad (1)$$

[1]Notice that this definition of foreign key differs from the more common assumption in which a foreign key refers exactly to a key, which corresponds to $B = C$ in the above definition.

where each $conj_i(\mathbf{x}, \mathbf{c})$ is an expression of the form

$$conj_i(\mathbf{x}, \mathbf{c}) = \exists \mathbf{y}.a_1 \wedge \ldots \wedge a_n$$

in which each a_i is an atom whose arguments are terms from the disjoint tuples of variables $\mathbf{x}$, $\mathbf{y}$, and from the tuple of constants $\mathbf{c}$ and such that each variable from $\mathbf{x}$ and $\mathbf{y}$ occurs in at least one atom a_i. The variables $\mathbf{x}$ are called the *distinguished* variables of the UCQ, while the other variables are called *existential* variables.

We call a UCQ a *conjunctive query* (CQ) when $m = 1$ in (1).

A *Boolean* UCQ is a UCQ without distinguished variables, i.e., an expression of the form $conj_1(\mathbf{c}) \vee \ldots \vee conj_m(\mathbf{c})$. Being a sentence, i.e., a closed first-order formula, such a query is either true or false in a database. Given a non-Boolean query q and a tuple of constants $\mathbf{c}$, we denote by $q(\mathbf{c})$ the Boolean query obtained from q by replacing the distinguished variables of q with the corresponding constants in $\mathbf{c}$.

Finally, the *size* of a conjunctive query q is the number of atoms in the body of q.

2.3. Problems studied

As a preliminary definition, we introduce the notion of homomorphism between a query and a database. Given a Boolean conjunctive query q and a database $\mathcal{D}$, a function h mapping every variable occurring in q to a constant occurring in $\mathcal{D}$ is a *homomorphism* from q to $\mathcal{D}$ if: (i) for every constant a occurring in q, $h(a) = a$; (ii) for every atom $r(t_1, \ldots, t_m)$ occurring in q, the fact $r(h(t_1), \ldots, h(t_m))$ belongs to $\mathcal{D}$.

Given a conjunctive query q and a database $\mathcal{D}$, we denote by $q^{\mathcal{D}}$ the set of tuples corresponding to the standard evaluation of the query over $\mathcal{D}$, i.e.:

$$q^{\mathcal{D}} = \{\mathbf{c} \mid \text{ there exists a homomorphism } h : q(\mathbf{c}) \rightarrow \mathcal{D}\}.$$

Moreover, if Q is the UCQ $\{\mathbf{x} \mid q_1(\mathbf{x}) \vee \ldots \vee q_m(\mathbf{x})\}$, then

$$\{\mathbf{c} \mid \text{ there exists } q_i(\mathbf{x}) (1 \leq i \leq m) \text{ and a homomorphism } h : q_i(\mathbf{c}) \rightarrow \mathcal{D}\}.$$

It is immediate to verify that $Q^{\mathcal{D}} = \bigcup_{i=1}^{m} q_i^{\mathcal{D}}$.

Given a set of ICs $\mathcal{I}$ and a database $\mathcal{D}$, we denote by $sem(\mathcal{I}, \mathcal{D})$ the set of databases $sem(\mathcal{I}, \mathcal{D}) = \{\mathcal{B} \mid \mathcal{B} \supseteq \mathcal{D} \text{ and } \mathcal{B} \text{ satisfies } \mathcal{I}\}$, while $sem_f(\mathcal{I}, \mathcal{D})$ denotes the subset of *finite* databases contained in $sem(\mathcal{I}, \mathcal{D})$.

Then, we define $ans(q, \mathcal{I}, \mathcal{D})$ and $ans_f(q, \mathcal{I}, \mathcal{D})$ as follows:

$$ans(q, \mathcal{I}, \mathcal{D}) = \{\mathbf{c} \mid \mathbf{c} \in q^{\mathcal{B}} \text{ for every } \mathcal{B} \in sem(\mathcal{I}, \mathcal{D})\}$$
$$ans_f(q, \mathcal{I}, \mathcal{D}) = \{\mathbf{c} \mid \mathbf{c} \in q^{\mathcal{B}} \text{ for every } \mathcal{B} \in sem_f(\mathcal{I}, \mathcal{D})\}$$

The above definition of $ans(q, \mathcal{I}, \mathcal{D})$ corresponds to the notion of *certain answers* in indefinite databases.

We now introduce the main problems studied in the paper, i.e., implication of ICs [14], OWA-consistency, OWA-answering, and finite controllability of OWA-answering [15, 2].

Implication of ICs. For unrestricted databases: given a set of ICs $\mathcal{I}$ and an IC I, we say that $\mathcal{I}$ implies I (and write $\mathcal{I} \models I$) if for every database $\mathcal{D}$ such that $\mathcal{D}$ satisfies $\mathcal{I}$, $\mathcal{D}$ satisfies I. For finite databases: $\mathcal{I}$ finitely implies I (and write $\mathcal{I} \models_f I$) if for every finite database $\mathcal{D}$ such that $\mathcal{D}$ satisfies $\mathcal{I}$, $\mathcal{D}$ satisfies I.

OWA-consistency. For unrestricted databases: given a set of ICs $\mathcal{I}$ and a database $\mathcal{D}$, we say that $\mathcal{D}$ is OWA-consistent with $\mathcal{I}$ if $sem(\mathcal{I}, \mathcal{D}) \neq \emptyset$. For finite databases: $\mathcal{D}$ is OWA$_f$-consistent with $\mathcal{I}$ if $sem_f(\mathcal{I}, \mathcal{D}) \neq \emptyset$.

OWA-answering. For unrestricted databases: given a set of ICs $\mathcal{I}$, a database $\mathcal{D}$, and a query q, compute $ans(q, \mathcal{I}, \mathcal{D})$. For finite databases (finite OWA-answering): compute $ans_f(q, \mathcal{I}, \mathcal{D})$.

The *decision problem* associated with OWA-answering is the following: given a query q and a tuple $\mathbf{c}$, decide whether $\mathbf{c} \in ans(q, \mathcal{I}, \mathcal{D})$, i.e., decide whether the Boolean query $q(\mathbf{c})$ is true in all databases in $sem(\mathcal{I}, \mathcal{D})$ (resp., $sem_f(\mathcal{I}, \mathcal{D})$). In the following, when we talk about (un)decidability of OWA-answering we actually refer to (un)decidability of the decision problem associated with OWA-answering.

Finite controllability. Finally, given a class of queries $\mathcal{Q}$ and a class of ICs $\mathcal{C}$, we say that OWA-answering $\mathcal{Q}$ under $\mathcal{C}$ is finitely controllable if, for every set of ICs $\mathcal{I} \subseteq \mathcal{C}$, for every query $q \in \mathcal{Q}$, and for every database $\mathcal{D}$, $ans(q, \mathcal{I}, \mathcal{D}) = ans_f(q, \mathcal{I}, \mathcal{D})$. In an analogous way, we define finite controllability of implication of ICs of a class of ICs $\mathcal{C}_1$ from a class of ICs $\mathcal{C}_2$.

3. Finite controllability of OWA-answering CQs under IDs

In this section we study finite controllability of OWA-answering of CQs in the presence of IDs, and prove the following result, which in fact closes a problem left open by Johnson and Klug [5], as we will explain in Section 5.

Theorem 1. *OWA-answering CQs under IDs is finitely controllable.*

The proof of the above theorem, which is the main result of the present paper, is actually very involved and requires the introduction of several auxiliary definitions, as well as the proof of several lemmas. We will start by recalling the well-known notion of chase (which we will call *canonical* chase) for inclusion dependencies. Such a chase may be infinite in the presence of cyclic IDs. Then, we will introduce a notion of *finite chase*, which is a modification of the chase procedure based on the idea of using a finite number of Skolem terms (i.e., labeled null values) in the construction of the chase, which guarantees termination of the chase even in the presence of cyclic IDs. The key idea of the construction is to use Skolem terms as labeled null values, limiting the nesting of Skolem functions in such terms to a number which is actually a parameter for the finite chase procedure. Finally, we will show that, in practice, the finite chase using m nesting levels of Skolem functions evaluates conjunctive queries with a number of atoms (and a number of existential variables) less than or equal to m in exactly the same way as the canonical (infinite) chase. This implies that OWA-answering conjunctive queries under IDs is finitely controllable.

3.1. The canonical chase

We start by recalling known results on OWA-answering of CQs for unrestricted databases. From now on, we extend the notion of fact given in Section 2, and say that a fact is an expression of the form $r(t_1, \ldots, t_n)$ where each t_i is either a constant or a functional term.

Definition 1 (canonical chase). *Given a set of IDs $\mathcal{I}$ and a database instance $\mathcal{D}$, we denote by chase$(\mathcal{I}, \mathcal{D})$ the (possibly infinite) database obtained starting from $\mathcal{D}$ and closing the database with respect to the following ID-chase rule:*

> ***if*** $I \in \mathcal{I}$, *with* $I = r[A] \subseteq s[B]$
> *and* $r(\mathbf{t}) \in$ *chase*$(\mathcal{I}, \mathcal{D})$

then *add to chase*$(\mathcal{I}, \mathcal{D})$ *a fact* $s(\mathbf{t}')$ *such that* $\mathbf{t}'[B] = \mathbf{t}[A]$ *and for each attribute* p *of* s *and such that* $p \notin B$,

$$\mathbf{t}'[p] = \phi_{I,p}(\mathbf{t}'[B])$$

Moreover, we say that the fact $s(\mathbf{t}')$ *in the above definition is* generated by $r(\mathbf{t})$ and I *in chase*$(\mathcal{I}, \mathcal{D})$. *Finally, we call* Skolem functions *the functions of the form* $\phi_{I,p}$ *used by the chase rule, and call* Skolem terms *the terms of the form* $\phi_{I,p}(\alpha)$ *introduced by the above chase rule, and distinguish them from the constants occurring in* $\mathcal{D}$.

Notice that the above construction may generate Skolem terms whose size is unbounded (in the presence of cyclic IDs). However, it is immediate to see that the set of such symbols that can be generated is countably infinite, and thus *chase*$(\mathcal{I}, \mathcal{D})$ is always either a finite set or a countably infinite set, i.e., it is actually a database according to the definitions provided in Section 2.

Given a fact f of *chase*$(\mathcal{I}, \mathcal{D})$, a *branch* for f in *chase*$(\mathcal{I}, \mathcal{D})$ is any sequence of facts $f_0, \ldots, f_m$ with $m \geq 0$ such that: (i) $f_0 \in \mathcal{D}$; (ii) $f_m = f$; (iii) for every i such that $1 \leq i \leq m$, the fact f_i can be generated by the ID-chase rule from the fact f_{i-1} and from some ID $I \in \mathcal{I}$.

From now on, we call *term* every constant or Skolem term.

We now extend the notion of homomorphism given in Section 2 to pairs of databases containing Skolem terms as follows. Given two databases $\mathcal{D}_1, \mathcal{D}_2$, a function h mapping every term occurring in $\mathcal{D}_1$ to a term occurring in $\mathcal{D}_2$ is a *homomorphism* from $\mathcal{D}_1$ to $\mathcal{D}_2$ if: (i) for every constant a occurring in $\mathcal{D}_1$, $h(a) = a$; (ii) for every fact $r(t_1, \ldots, t_m)$ occurring in $\mathcal{D}_1$, the fact $r(h(t_1), \ldots, h(t_m))$ belongs to $\mathcal{D}_2$.

It is known that the database *chase*$(\mathcal{I}, \mathcal{D})$ constitutes a *canonical model* for OWA-answering conjunctive queries, in the following sense:

Proposition 1 ([5, 16]). *Let* $\mathcal{D}$ *be a database instance and let* $\mathcal{I}$ *be a set of IDs. Then, chase*$(\mathcal{I}, \mathcal{D}) \in$ *sem*$(\mathcal{I}, \mathcal{D})$*. Moreover, for every* $\mathcal{D}' \in$ *sem*$(\mathcal{I}, \mathcal{D})$*, there exists a homomorphism from chase*$(\mathcal{I}, \mathcal{D})$ *to* $\mathcal{D}'$.

The above property immediately implies the following proposition.

Proposition 2 ([5, 16]). *Let* $\mathcal{D}$ *be a database instance and let* $\mathcal{I}$ *be a set of IDs. For every CQ* q *and for every tuple* $\mathbf{c}$, $\mathbf{c} \in$ *ans*$(q, \mathcal{I}, \mathcal{D})$ *iff* $\mathbf{c} \in q^{chase(\mathcal{I}, \mathcal{D})}$.

3.2. The finite chase

Now, in order to prove finite controllability of CQs under IDs, we modify the canonical chase above recalled. The modified version always produces a finite database. However, differently from the canonical chase, in this case we have to fix a priori the maximum size of the CQs. In other words, a finite chase constitutes a correct model of $\mathcal{I}$ and $\mathcal{D}$ only with respect to CQs of a given size, as we will show in the following.

First, we introduce some auxiliary definitions. We say that a Skolem function ϕ occurs at *depth* 1 in a Skolem term t if ϕ is the outermost Skolem function of t. Then, inductively, if $t = \psi(t_1, \ldots, t_n)$ and ϕ occurs at depth k in some t_i, we say that ϕ occurs at depth $k+1$ in t. Moreover, we say that a Skolem term has depth k if there is a subterm of t occurring within the scope of $k-1$ Skolem functions, and there is no subterm of t occurring within the scope of k Skolem functions.

Then, we define the functions trunc_k, where k is any positive integer, over the domain of all constants and Skolem terms. The functions are inductively defined as follows (c denotes a constant, $\phi(\alpha)$ denotes a Skolem term whose outermost Skolem function is ϕ, and $\langle t_1, \ldots, t_n \rangle$ denotes a tuple of constants and/or Skolem terms):

- if $k = 1$ then $\mathit{trunc}_k(c) = c$ and $\mathit{trunc}_k(\phi(\alpha)) = \phi$;
- if $k > 1$ then $\mathit{trunc}_k(c) = c$ and $\mathit{trunc}_k(\phi(\alpha)) = \phi(\mathit{trunc}_{k-1}(\alpha))$;
- for every k, $\mathit{trunc}_k(\langle t_1, \ldots, t_n \rangle) = \langle \mathit{trunc}_k(t_1), \ldots, \mathit{trunc}_k(t_n) \rangle$.

Informally, $\mathit{trunc}_k(t)$ corresponds to the "truncated" version of t in which all subterms of t occurring at depth $k+1$ have been eliminated.

The following are some simple examples that illustrate the function trunc_k:

$$\begin{array}{l} \mathit{trunc}_2(\phi(\psi(a))) = \phi(\psi) \\ \mathit{trunc}_3(\phi(\psi(\xi(a), b))) = \phi(\psi(\xi, b)) \\ \mathit{trunc}_4(\phi(\psi(\xi(a, \rho(b)), \rho(\psi(\eta(c), d))))) = \phi(\psi(\xi(a, \rho), \rho(\psi))) \end{array}$$

As shown by the above examples, when a Skolem function occurs at depth k in a Skolem term t, it has arity 0 (no arguments) in the term returned by $\mathit{trunc}_k(t)$ (i.e., it plays the role of a constant in $\mathit{trunc}_k(t)$).

From now on, we denote by *maxArityID* the maximum arity of inclusions in $\mathcal{I}$.

Definition 2 (finite chase). *Given a set of IDs $\mathcal{I}$, a database instance $\mathcal{D}$, and an integer $m \geq 1$, we denote by $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ the database obtained starting from $\mathcal{D}$ and closing the database with respect to the following f-chase-rule (that is applied based on a total order on the IDs in $\mathcal{I}$ and on a total order on the facts already generated):*

> ***if*** *$I \in \mathcal{I}$, with $I = r[A] \subseteq s[B]$*
> *and $r(\mathbf{t}) \in \mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$*
> *and there is no fact in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ of the form $s(\mathbf{t}')$*
> *such that $\mathbf{t}'[B] = \mathbf{t}[A]$,*
> ***then*** *add to $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ the fact $s(\mathbf{t}')$ such that $\mathbf{t}'[B] = \mathbf{t}[A]$ and for each attribute p of s and such that $p \notin B$,*
>
> $$\mathbf{t}'[p] = \mathit{trunc}_m(\phi^j_{I,p}(\mathbf{t}'[B]))$$
>
> *where j is the smallest integer such that $1 \leq j \leq m \times (\mathit{maxArityID})^m + 1$ and such that, for every attribute p' of s such that $p' \notin B$, the Skolem function symbol $\phi^j_{I,p'}$ does not occur in $\mathbf{t}'[B]$.*

We say that the fact $s(\mathbf{t}')$ in the above definition is generated by $r(\mathbf{t})$ and I *in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$. Moreover, we call* existential value in $s(\mathbf{t}')$ *every Skolem term in $\mathbf{t}'[B]$.*

Given a fact f belonging to $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$, the *branch for f in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$*, denoted by $B(f)$, is the sequence of facts that generates f in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$, i.e., the sequence $f_0, \ldots, f_h$ with $h \geq 0$ such that: (i) $f_0 \in \mathcal{D}$; (ii) $f_h = f$; (iii) for every i such that $1 \leq i \leq h$, the fact f_i has been generated by the f-chase-rule from the fact f_{i-1} and from some ID $I \in \mathcal{I}$. Notice that, differently from the canonical chase, for every fact $f \in \mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ there is a unique branch for f in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$.[2]

Moreover, since the maximum depth of Skolem functions in terms occurring in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ is m, it immediately follows that the maximum

[2]Observe that, since we are interested in the branches of the finite chase, $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ should be defined as a *set of branches* rather than a database (i.e., a set of facts), to keep track of the chase rules applied to generate the final database. However, to simplify notation, we denote by $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ only the set of facts generated by the finite chase procedure, and when we speak about a "branch in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$" we implicitly refer to the procedure for building the database $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ starting from $\mathcal{I}$ and $\mathcal{D}$ (and the integer m).

number of occurrences of Skolem functions in a term in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ is $\sum_{i=0}^{m-1}(\mathit{maxArityID})^i$, which is less than or equal to $m \times \mathit{maxArityID}^{m-1}$. Therefore, the maximum number of different Skolem functions that can appear in $t'[B]$ is $\mathit{maxArityID} \times m \times (\mathit{maxArityID})^{m-1} = m \times (\mathit{maxArityID})^m$. This proves that Condition 2 of Definition 2 can always be satisfied, i.e., it is always possible to pick a value for j such that, for every attribute p' of s such that $p' \notin B$, the function symbol $\phi^j_{I,p'}$ does not occur in $t'[B]$.

Observe that, in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$, the depth of the functions of the Skolem terms is bound to m, and the number of function and constant symbols used is finite, thus the number of distinct Skolem terms introduced by the f-chase-rule is finite, and therefore the number of terms involved in the construction of $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ is finite. Consequently, $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ is always a finite database.

Finally, notice that $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ is unique, since we assume that the f-chase-rule is applied based on a fixed total order on the IDs in $\mathcal{I}$ and a total order on the facts of $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ already generated.

Example 1. Here is a very simple example of a finite chase. Let $\mathcal{D} = \{r(a, b)\}$, $\mathcal{I} = \{r[2] \subseteq r[1]\}$. The finite chase $\mathit{fchase}(\mathcal{I}, \mathcal{D}, 2)$ is the following (for ease of notation, since there is only one ID in $\mathcal{I}$ we omit subscripts in the Skolem function symbols):

$$\begin{array}{c} r(a,b) \\ \downarrow \\ r(b,\phi^1(b)) \\ \downarrow \\ r(\phi^1(b),\phi^2(\phi^1)) \\ \downarrow \\ r(\phi^2(\phi^1),\phi^3(\phi^2)) \\ \downarrow \\ r(\phi^3(\phi^2),\phi^1(\phi^3)) \\ \downarrow \\ r(\phi^1(\phi^3),\phi^2(\phi^1)) \end{array}$$

Notice that each of the above facts is generated by the f-chase-rule from the previous fact and the ID I. Notice also that the construction stops after the generation of the fact $r(\phi^1(\phi^3), \phi^2(\phi^1))$, because the presence of the fact

$r(\phi^2(\phi^1), \phi^3(\phi^2))$ makes the inclusion dependency not applicable by the f-chase-rule to the fact $r(\phi^1(\phi^3), \phi^2(\phi^1))$.

Notice also that, in this case, the canonical chase $chase(\mathcal{I}, \mathcal{D})$ is infinite.

■

Example 2. Here is a slightly more involved example of finite chase. Let $\mathcal{I}$ and $\mathcal{D}$ be as follows:

$$\mathcal{I} = \{r[2] \subseteq r[1]\ (I_1),\ \ s[1] \subseteq s[2]\ (I_2),\ \ s[1,2] \subseteq t[1,2]\ (I_3),\ \ v[1] \subseteq u[1]\ (I_4)\}$$
$$\mathcal{D} = \{r(a,b),\ r(c,d),\ v(g),\ s(b,c),\ u(e,f,e)\}$$

For ease of notation, instead of writing Skolem functions with complex subscripts, we use three different symbols ϕ, ψ, ξ. More precisely, in the following ϕ stands for $\phi_{I_1,2}$, ψ stands for $\phi_{I_2,1}$, ξ stands for $\phi_{I_3,3}$, ρ stands for $\phi_{I_4,2}$, and σ stands for $\phi_{I_4,3}$.

The finite chase $fchase(\mathcal{I}, \mathcal{D}, 3)$ is the one displayed in Figure 1, in which the facts from $\mathcal{D}$ are underlined. There are 13 branches (B1–B13) in the finite chase: we assume that in this case the finite chase has been generated according to a depth-first strategy, i.e., first all facts of branch B1 have been generated, then all facts of B2, and so on. The reason for the termination of branches B1 and B4 is analogous to what explained in Example 1. As for branch B2, notice that the construction of this branch stops after the generation of the fact $r(\phi^2(\phi^1(d)), \phi^3(\phi^2(\phi^1)))$, since the presence in branch B1 (that has already been generated) of the fact $r(\phi^3(\phi^2(\phi^1)), \phi^4(\phi^3(\phi^2)))$ makes the inclusion I_1 not applicable to the fact $r(\phi^2(\phi^1(d)), \phi^3(\phi^2(\phi^1)))$.

Finally, notice also that the canonical chase $chase(\mathcal{I}, \mathcal{D})$ is infinite. ■

3.3. Completeness of the finite chase

We now prove completeness of the finite chase with respect to the canonical chase $chase(\mathcal{I}, \mathcal{D})$.

Lemma 1. *Let $\mathcal{I}$ be a set of IDs, let $\mathcal{D}$ be a database instance, and let m be an integer such that $m \geq 1$. Then, $fchase(\mathcal{I}, \mathcal{D}, m) \in sem_f(\mathcal{I}, \mathcal{D})$.*

Proof. From Definition 2 it immediately follows that (i) $\mathcal{D} \subseteq fchase(\mathcal{I}, \mathcal{D}, m)$; (ii) $fchase(\mathcal{I}, \mathcal{D}, m)$ satisfies all IDs in $\mathcal{I}$. Therefore, $fchase(\mathcal{I}, \mathcal{D}, m) \in sem_f(\mathcal{I}, \mathcal{D})$. □

$$
\begin{array}{c}
\underline{r(a,b)} \\
\downarrow \\
r(b,\phi^1(b)) \\
\downarrow \\
r(\phi^1(b),\phi^2(\phi^1(b))) \\
\downarrow \\
r(\phi^2(\phi^1(b)),\phi^3(\phi^2(\phi^1))) \\
\downarrow \\
r(\phi^3(\phi^2(\phi^1)),\phi^4(\phi^3(\phi^2))) \\
\downarrow \\
r(\phi^4(\phi^3(\phi^2)),\phi^1(\phi^4(\phi^3))) \\
\downarrow \\
r(\phi^1(\phi^4(\phi^3)),\phi^2(\phi^1(\phi^4))) \\
\downarrow \\
r(\phi^2(\phi^1(\phi^4)),\phi^3(\phi^2(\phi^1))) \\
(B1)
\end{array}
\qquad
\begin{array}{c}
\underline{r(c,d)} \\
\downarrow \\
r(d,\phi^1(d)) \\
\downarrow \\
r(\phi^1(d),\phi^2(\phi^1(d))) \\
\downarrow \\
r(\phi^2(\phi^1(d)),\phi^3(\phi^2(\phi^1))) \\
(B2)
\end{array}
\qquad
\begin{array}{c}
\underline{v(g)} \\
\downarrow \\
u(g,\rho^1(g),\sigma^1(g)) \\
(B3)
\end{array}
$$

$$
\begin{array}{rcl}
\underline{s(b,c)} & \rightarrow & u(b,c,\xi^1(b,c)) \quad (B5) \\
\downarrow & & \\
s(\psi^1(b),b) & \rightarrow & u(\psi^1(b),b,\xi^1(\psi^1(b),b)) \quad (B6) \\
\downarrow & & \\
s(\psi^2(\psi^1(b)),\psi^1(b)) & \rightarrow & u(\psi^2(\psi^1(b)),\psi^1(b),\xi^1(\psi^2(\psi^1),\psi^1(b))) \quad (B7) \\
\downarrow & & \\
s(\psi^3(\psi^2(\psi^1)),\psi^2(\psi^1(b))) & \rightarrow & u(\psi^3(\psi^2(\psi^1)),\psi^2(\psi^1(b)),\xi^1(\psi^3(\psi^2),\psi^2(\psi^1))) \quad (B8) \\
\downarrow & & \\
s(\psi^4(\psi^3(\psi^2)),\psi^3(\psi^2(\psi^1))) & \rightarrow & u(\psi^4(\psi^3(\psi^2)),\psi^3(\psi^2(\psi^1)),\xi^1(\psi^4(\psi^3),\psi^3(\psi^2))) \quad (B9) \\
\downarrow & & \\
s(\psi^1(\psi^4(\psi^3)),\psi^4(\psi^3(\psi^2))) & \rightarrow & u(\psi^1(\psi^4(\psi^3)),\psi^4(\psi^3(\psi^2)),\xi^1(\psi^1(\psi^4),\psi^4(\psi^3))) \quad (B10) \\
\downarrow & & \\
s(\psi^2(\psi^1(\psi^4)),\psi^1(\psi^4(\psi^3))) & \rightarrow & u(\psi^2(\psi^1(\psi^4)),\psi^1(\psi^4(\psi^3)),\xi^1(\psi^2(\psi^1),\psi^1(\psi^4))) \quad (B11) \\
\downarrow & & \\
s(\psi^3(\psi^2(\psi^1)),\psi^2(\psi^1(\psi^4))) & \rightarrow & u(\psi^3(\psi^2(\psi^1)),\psi^2(\psi^1(\psi^4)),\xi^1(\psi^3(\psi^2),\psi^2(\psi^1))) \quad (B12) \\
(B4) & &
\end{array}
\qquad
\begin{array}{c}
\underline{u(e,f,e)} \\
(B13)
\end{array}
$$

Figure 1: Finite chase $fchase(\mathcal{I},\mathcal{D},3)$ of Example 2.

Lemma 2. *Let $\mathcal{I}$ be a set of IDs, let $\mathcal{D}$ be a database instance, and let m be an integer such that $m \geq 1$. For every CQ q and for every tuple of constants* $\mathbf{c}$*, if* $\mathbf{c} \in q^{chase(\mathcal{I},\mathcal{D})}$ *then* $\mathbf{c} \in q^{fchase(\mathcal{I},\mathcal{D},m)}$.

Proof. From Lemma 1 and from the fact that $sem_f(\mathcal{I},\mathcal{D}) \subseteq sem(\mathcal{I},\mathcal{D})$, it follows that $fchase(\mathcal{I},\mathcal{D},m) \in sem(\mathcal{I},\mathcal{D})$. Therefore, since by hypothesis $\mathbf{c} \in q^{chase(\mathcal{I},\mathcal{D})}$, from Proposition 2 it follows that $q(\mathbf{c})$ is true in every database in $sem(\mathcal{I},\mathcal{D})$, and hence $\mathbf{c} \in q^{fchase(\mathcal{I},\mathcal{D},m)}$. □

Unfortunately, proving soundness of the evaluation of conjunctive queries over the finite chase with respect to the canonical chase is much harder and more involved than the above proof of completeness. Thus, in order to make the material in the following more readable, we first present an overview of the proof.

3.4. Overview of the proof of soundness of the finite chase

Let F be a set of facts of the finite chase $fchase(\mathcal{I},\mathcal{D},m)$ such that the cardinality of F is sufficiently smaller than m (as explained in detail below), and let F constitute an image of a conjunctive query q (i.e., such that there exists a query homomorphism from q to F). Our goal is to map the subset F of facts of $fchase(\mathcal{I},\mathcal{D},m)$ to a set of facts IM of $chase(\mathcal{I},D)$ which constitutes an image of F (Lemma 12): this in turn implies soundness of the evaluation of conjunctive queries whose size is sufficiently smaller than m over the finite chase $fchase(\mathcal{I},\mathcal{D},m)$ with respect to the evaluation over the canonical chase (Lemma 13).

To identify such a set IM, an intuitive idea would be to simply map the branches of the finite chase relative to every fact $f \in F$ into the "naturally corresponding" branch of the canonical chase. In other words, for every $f \in F$, let B_{fin} be the branch for f in the finite chase and let f_0 be the root of B_{fin} (i.e., f_0 is a fact from $\mathcal{D}$); then, the definition of the canonical chase implies that, in $chase(\mathcal{I},\mathcal{D})$, there exists a branch B_{can} starting from f_0 and obtained by applying the same sequence of IDs that have been used in the branch B_{fin}. Such a branch ends with a fact f' that is equal to f on the first m nesting levels of the Skolem functions, in the following sense: the relation symbol of f coincides with the relation symbol of f' and, for every pair t, t' of corresponding arguments in f and f', $trunc_m(t)$ and $trunc_m(t')$ are equal, up to superscripts of the Skolem functions. This mapping identifies a set of facts $IM = \{f'_1, \ldots, f'_n\}$ of $chase(\mathcal{I},\mathcal{D})$: however, a homomorphism from F

to such a set IM is guaranteed to exist *only if there are no join Skolem terms in F having depth m*, where a join Skolem term in F is a Skolem term that occurs in at least two distinct facts of F. Indeed, due to the reuse of Skolem terms done in the finite chase, if a Skolem term t having depth m occurs in two facts f_i, f_j of the finite chase, the occurrence of t in f_i may be mapped by the above strategy to a Skolem term of the canonical chase that is different from the Skolem term used to map the occurrence of t in f_j (notice that, if instead t has depth less than m, then it can immediately be shown that the two occurrences of t are necessarily mapped to the same Skolem term in the canonical chase).

In order to correctly handle join Skolem terms of depth m in F, we define a mapping of the branches of the finite chase relevant for F to the canonical chase (Definition 7) that identifies a set of facts IM of the canonical chase by suitably "linking" different branches of the finite chase and then mapping such composed branches to corresponding branches of the canonical chase. This mapping guarantees that, for every Skolem term t of depth m, every occurrence of t in F is mapped in IM to the same Skolem term t' (for a more detailed explanation, we refer the reader to Definition 7 and to the proof of Lemma 12).

A crucial property to correctly implement this idea is to find the right "linking points" in the branches of both the finite chase and the canonical chase. The identification of such linking points is obtained by resorting to the relations of *predecessor* and *sibling* holding between Skolem terms in a branch of the finite chase (Definition 3). Moreover, after showing some auxiliary properties of the predecessor and sibling relations (Lemma 4, Lemma 5, Lemma 6), we define a way for mapping, through the function τ (Definition 5), all the join Skolem terms of depth m that are "relevant" for of a fact $f \in F$ (the notion of relevant join Skolem term for f is formally stated by the function $RJST$). The function τ is the key tool in order to prove correctness of the above mapping of the portion of the finite chase relative to the facts in F to a portion of the canonical chase (Lemma 12).

Some of the above results (and in particular Lemma 12) hold under the condition that the finite chase $fchase(\mathcal{I}, \mathcal{D}, m)$ and the set $F \subseteq fchase(\mathcal{I}, \mathcal{D}, m)$ are such that $m \geq 2(|F| \times |JST(F)|) + 2$, where $|F|$ is the number of facts in F and $|JST(F)|$ represents the number of join Skolem terms of depth m of F. We remark that these conditions on m and F do not constitute any restriction towards the proof of the general result, i.e., Theorem 1 (as we will explain right after Lemma 13).

3.5. Proof of soundness of the finite chase

In this section we provide a detailed proof of soundness of the evaluation of conjunctive queries over the finite chase. Throughout the section, we assume that we are given a set of IDs $\mathcal{I}$, a database $\mathcal{D}$ and an integer $m \geq 2$.

We start with some preliminary lemmas and definitions.

Lemma 3. *Let f be a fact of fchase$(\mathcal{I}, \mathcal{D}, m)$. For every Skolem function ϕ, there exists at most one argument of f having ϕ as its most external Skolem function.*

Proof. The proof is by induction on the structure of *fchase*$(\mathcal{I}, \mathcal{D}, m)$. The base case is immediate since there are no occurrences of Skolem functions in $\mathcal{D}$. As for the inductive case, when a new fact f is generated by applying the f-chase-rule to a fact f' that has no pair of arguments sharing the same external function, then also f has no pair of arguments sharing the same external function, because the f-chase-rule uses Skolem functions that do not occur in f' for the existential values in f, and there are no repetitions of attributes in the inclusion dependencies, which implies that every argument in f' is repeated at most once in f. □

Let $f = s(t_1, \ldots, t_h)$ be a fact of *fchase*$(\mathcal{I}, \mathcal{D}, m)$, let $B(f) = f_0, \ldots, f_n$ be the branch for f in *fchase*$(\mathcal{I}, \mathcal{D}, m)$, and let t_i (with $1 \leq i \leq h$) be a Skolem term. An *introduction point* for t_i in $B(f)$ is a fact f_j (with $1 \leq j \leq n$) such that t_i is generated as an existential value in f_j (i.e., t_i is not propagated by the f-chase-rule from the previous fact f_{j-1} of $B(f)$). Observe that (due to the f-chase-rule) there may be multiple introduction points for the same term in a branch. We say that the *last introduction point* of t in $B(f)$ is the introduction point of t that is the closest to the end of the branch $B(f)$ (i.e., the closest to f in $B(f)$).

Definition 3. *Let f be a fact of fchase$(\mathcal{I}, \mathcal{D}, m)$, let $B(f)$ be the branch for f in fchase$(\mathcal{I}, \mathcal{D}, m)$, and let t_1, t_2 be two Skolem terms that occur as arguments of f. Then:*

- *if the last introduction point of t_1 in $B(f)$ coincides with the last introduction point of t_2 in $B(f)$, then we say that t_1 and t_2 are* siblings *in $B(f)$. More precisely, if in such a fact t_1 is the argument at position p and t_2 is the argument at position q, then we say that t_1 and t_2 are* p-q-siblings *in $B(f)$ (we also say that t_1 is the p-sibling of t_2 in $B(f)$, and t_2 is the q-sibling of t_1 in $B(f)$);*

- *we say that* t_1 *is the* k-predecessor of t_2 in $B(f)$ *if: (i) the last introduction point of* t_1 *in* $B(f)$ *precedes the last introduction point of* t_2 *in* $B(f)$*; and (ii)* t_1 *occurs at position* k *in the last introduction point of* t_2 *in* $B(f)$*. We also say that* t_1 *is a* predecessor of t_2 in $B(f)$ *if* t_1 *is the* k*-predecessor of* t_2 *in* $B(f)$ *for some* k*.*

Example 3. In Example 1 there is only one branch (let us call it B) in *fchase*$(\mathcal{I}, \mathcal{D}, 2)$. The Skolem term $\phi^2(\phi^1)$ has two introduction points in B, i.e., the facts $r(\phi^1(b), \phi^2(\phi^1))$ and $r(\phi^1(\phi^3), \phi^2(\phi^1))$ (and the last introduction point of $\phi^2(\phi^1)$ is $r(\phi^1(\phi^3), \phi^2(\phi^1)))$ while all other Skolem terms occurring in B have only one introduction point. Moreover, the term $\phi^2(\phi^1)$ is the 1-predecessor of $\phi^3(\phi^2)$ in B, $\phi^3(\phi^2)$ is the 1-predecessor of $\phi^1(\phi^3)$ in B and $\phi^1(\phi^3)$ is the 1-predecessor of $\phi^2(\phi^1)$ in B. Notice that the Skolem term $\phi^1(b)$ is *not* the 1-predecessor of $\phi^2(\phi^1)$ in B, because $r(\phi^1(b), \phi^2(\phi^1))$ is not the last introduction point of $\phi^2(\phi^1)$ in B. Finally, in Example 2, the Skolem terms $\rho^1(g)$, $\sigma^1(g)$ are 2-3-siblings in branch B3. ∎

Based on the above notion of predecessor, we now define the notion of ancestor in a branch of the finite chase.

Definition 4. *The notion of* j*-ancestor (for every integer* $j \geq 0$*) is inductively defined as follows:*

- *for every term* t_1 *occurring in* $B(f)$*,* t_1 *is* 0*-ancestor of itself in* $B(f)$*;*
- *for every pair of terms* t_1, t_2*,* t_1 *is* 1*-ancestor of* t_2 *in* $B(f)$ *if* t_1 *is a predecessor of* t_2 *in* $B(f)$*;*
- *for every pair of terms* t_1, t_2*,* t_1 *is* j*-ancestor of* t_2 *in* $B(f)$ *if there exists* t' *such that* t_1 *is a predecessor of* t' *in* $B(f)$ *and* t' *is (*j*-1)-ancestor of* t_2 *in* $B(f)$*.*

Moreover, we say that t_1 *is an* ancestor of t_2 in $B(f)$ *if there exists an integer* j *such that* t_1 *is a* j*-ancestor of* t_2 *in* $B(f)$*.*

Let F be a set of facts such that $F \subseteq$ *fchase*$(\mathcal{I}, \mathcal{D}, m)$. We denote by $JST(F)$ the set of Skolem terms of depth m that occur in at least two distinct facts of F (we point out that, as an immediate consequence of Lemma 3, there exists no Skolem term that occurs twice in the same fact of *fchase*$(\mathcal{I}, \mathcal{D}, m)$).[3]

Notice that, in the above definition, we do not consider join Skolem terms of depth less than m, because (as we have already briefly explained in Section 3.4) such terms are not problematic when mapping the branches $B(f_1), \ldots, B(f_n)$ of *fchase*$(\mathcal{I}, \mathcal{D}, m)$ to "corresponding" branches of *chase*$(\mathcal{I}, \mathcal{D})$: this is due to the fact that every Skolem term t having depth less than m has a unique introduction point in *fchase*$(\mathcal{I}, \mathcal{D}, m)$.

Example 4. Consider the finite chase of Example 2, and let F be the following set of facts:

$$r(\phi^4(\phi^3(\phi^2)), \phi^1(\phi^4(\phi^3)))$$
$$r(\phi^3(\phi^2(\phi^1)), \phi^4(\phi^3(\phi^2)))$$
$$r(\phi^2(\phi^1(d)), \phi^3(\phi^2(\phi^1)))$$
$$s(\psi^2(\psi^1(b)), \psi^1(b))$$
$$u(\psi^2(\psi^1(b)), \psi^1(b), \xi^1(\psi^2(\psi^1), \psi^1(b)))$$

Then,

$$JST(F) = \{(\phi^4(\phi^3(\phi^2)),\ \phi^3(\phi^2(\phi^1)),\ \psi^2(\psi^1(b))\}$$

Notice that the Skolem term $\psi^1(b)$ does not belong to $JST(F)$, because $\psi^1(b)$ has depth 2 and in Example 2 we are considering $m = 3$ (i.e., F is a subset of *fchase*$(\mathcal{I}, \mathcal{D}, 3)$). ■

Let t_1, t_2 be two terms and let $k \geq 1$. We write $t_1 \overset{k}{=} t_2$ if $trunc_k(t_1) = trunc_k(t_2)$. That is, $t_1 \overset{k}{=} t_2$ if t_1 and t_2 coincide in the k most external levels of Skolem functions. Moreover, given two facts f_1, f_2, we write $f_1 \overset{k}{=} f_2$ if the relation symbol of f_1 coincides with the relation symbol of f_2 and for every pair t, t' of corresponding arguments of f_1 and f_2, $t \overset{k}{=} t'$.

The following three lemmas are obvious consequences of the definition of *fchase* and the definition of $trunc_k$.

[3] Of course, there exists a distinct function JST for every value of m: to simplify notation, however, when denoting such function we omit the parameter m (the value of m will always be clear from the context). We will do analogous simplifications of the notation for some other auxiliary functions defined later on in this section.

Lemma 4. *If there exist branches B_1 and B_2 in fchase$(\mathcal{I}, \mathcal{D}, m)$ and an integer k such that t_1 is the k-predecessor of t_2 in B_1, t'_1 is the k-predecessor of t'_2 in B_2, and $t_2 \stackrel{h}{=} t'_2$ with $h \geq 1$, then $t_1 \stackrel{h-1}{=} t'_1$. Moreover, if there exist branches B_1 and B_2 in fchase$(\mathcal{I}, \mathcal{D}, m)$ such that t_1 and t_2 are p-q-siblings in B_1, t'_1 and t'_2 are p-q-siblings in B_2, and $t_2 \stackrel{h}{=} t'_2$ with $h \geq 2$, then $t_1 \stackrel{h}{=} t'_1$.*

Lemma 5. *Let $m \geq 2$, let t and t' be terms occurring in a branch B of fchase$(\mathcal{I}, \mathcal{D}, m)$, and let g and g' be introduction points of t and t' respectively. If $t \stackrel{h}{=} t'$ for some h such that $2 \leq h \leq m$, then $g \stackrel{h-1}{=} g'$.*

Lemma 6. *Let f be a fact of fchase$(\mathcal{I}, \mathcal{D}, m)$ and let t_1 and t_2 be distinct Skolem terms such that t_1 occurs at position k_1 in f (i.e., as the k_1-th argument of f) and t_2 occurs at position k_2 in f. Then, one of the following cases holds: (i) t_1 is the k_1-predecessor of t_2 in $B(f)$; (ii) t_2 is the k_2-predecessor of t_1 in $B(f)$; (iii) t_1 and t_2 are k_1-k_2-siblings in $B(f)$.*

We now define the ternary relation τ, which is a central notion in our proof of soundness of the finite chase.

Definition 5. *Let F be a set of facts such that $F \subseteq$ fchase$(\mathcal{I}, \mathcal{D}, m)$ and let T be the set of terms occurring in fchase$(\mathcal{I}, \mathcal{D}, m)$. We define τ as the minimal subset of $F \times JST(F) \times T$ satisfying the following equation:*

$$\begin{aligned}
\tau = \{\, \langle f, t, t \rangle \mid\ & f \in F \textit{ and } t \in JST(F) \textit{ and } t \textit{ occurs in } f \,\} \cup \\
\{\, \langle f, t_1, t''_1 \rangle \mid\ & f \in F \textit{ and there exist } f', t'_1, t_2, t'_2, t''_2, k \textit{ s.t. } \langle f', t_1, t'_1 \rangle \in \tau \\
& \textit{and } \langle f', t_2, t'_2 \rangle \in \tau \textit{ and } \langle f, t_2, t''_2 \rangle \in \tau \\
& \textit{and } t'_1 \textit{ is the } k\textit{-predecessor of } t'_2 \textit{ in } B(f') \\
& \textit{and } t''_1 \textit{ is the } k\textit{-predecessor of } t''_2 \textit{ in } B(f) \,\} \cup \\
\{\, \langle f, t_1, t''_1 \rangle \mid\ & f \in F \textit{ and there exist } f', t'_1, t_2, t'_2, t''_2, p, q \textit{ s.t. } \langle f', t_1, t'_1 \rangle \in \tau \\
& \textit{and } \langle f', t_2, t'_2 \rangle \in \tau \textit{ and } \langle f, t_2, t''_2 \rangle \in \tau \\
& \textit{and } t'_1 \textit{ and } t'_2 \textit{ are } p\textit{-}q\textit{-siblings in } B(f') \\
& \textit{and } t''_1 \textit{ and } t''_2 \textit{ are } p\textit{-}q\textit{-siblings in } B(f) \,\}
\end{aligned}$$

In the following, we will also make use of the following, equivalent, bottom-up inductive definition of τ: τ is the relation iteratively obtained starting from $\tau_0 = \{ \langle f, t, t \rangle \mid f \in F$ and $t \in JST(F)$ and t occurs in $f \}$ and in which τ_{j+1} is defined as the relation obtained from τ_j by adding one triple (if such a triple exists) arbitrarily chosen among the triples $\langle f, t_1, t''_1 \rangle$ such that $f \in F$ and $\langle f, t_1, t''_1\rangle \notin \tau_j$ and: (i) either there exist $f', t'_1, t_2, t'_2, t''_2, k$ such that $\langle f', t_1, t'_1\rangle \in \tau_j$ and $\langle f', t_2, t'_2\rangle \in \tau_j$ and $\langle f, t_2, t''_2\rangle \in \tau$ and t'_1 is the k-predecessor of t'_2 in $B(f')$ and t''_1 is the k-predecessor of t''_2 in $B(f)$; or (ii) there exist $f', t'_1, t_2, t'_2, t''_2, p, q$ such that $\langle f', t_1, t'_1\rangle \in \tau_j$ and $\langle f', t_2, t'_2\rangle \in \tau_j$ and $\langle f, t_2, t''_2\rangle \in \tau$ and t'_1 and t'_2 are p-q-siblings in $B(f')$ and t''_1 and t''_2 are p-q-siblings in $B(f)$. If such a triple does not exist, then we define $\tau_{j+1} = \tau_j$.

Before giving an intuitive explanation of τ, we first need to restrict our attention to cases in which τ is a function. Specifically, we will show in Lemma 10 that, if m is sufficiently greater than the cardinality of F and $JST(F)$, then τ is actually a binary function, i.e., given $f \in F$ and $t \in JST(F)$, there is at most one triple of the form $\langle f, t, t'\rangle$ in τ. To prove this property, we need some prelimary lemmas (Lemma 7, Lemma 8, and Lemma 9).

The following lemma shows that the triple $\langle f, t, t'\rangle$ added at step j of the above bottom-up construction of τ is such that $t \overset{m-j}{=} t'$.

Lemma 7. *Let j be an integer such that $0 \leq j < m$ and let $\langle f, t, t'\rangle \in \tau_j$. Then, $t \overset{m-j}{=} t'$.*

Proof. The proof is by induction. Base case: if $\langle f, t, t'\rangle \in \tau_0$ then $t = t'$, so the thesis follows. Inductive case: suppose the thesis holds for τ_i with i such that $1 \leq i < m - 1$. Now let $\langle f, t, t'\rangle$ be the triple added to τ_{i+1} at step $i + 1$. Now, by Definition 5 there are two possible cases: (i) there exist triples $\langle f'', t, t''\rangle$, $\langle f'', t_1, t''_1\rangle$, $\langle f, t_1, t'_1\rangle$ in τ_i and an integer k such that t'' is the k-predecessor of t''_1 in $B(f'')$ and t' is the k-predecessor of t'_1 in $B(f)$; (ii) there exist triples $\langle f'', t, t''\rangle$, $\langle f'', t_1, t''_1\rangle$, $\langle f, t_1, t'_1\rangle$ in τ_i and integers p, q such that t'' and t''_1 are p-q-siblings in $B(f'')$ and t' and t'_1 are p-q-siblings in $B(f)$. In both cases, by the inductive hypothesis, we have that $t_1 \overset{m-i}{=} t'_1$ and $t_1 \overset{m-i}{=} t''_1$; consequently, by Lemma 4 it follows that, in both cases, $t' \overset{m-i-1}{=} t''$, Moreover, since $t \overset{m-i}{=} t''$ by the inductive hypothesis, it follows that $t \overset{m-i-1}{=} t'$ in case (i), and $t \overset{m-i}{=} t'$ in case (ii), which proves the claim. □

The following property follows immediately from Definition 2 and Definition 3.

Lemma 8. *Let $f \in fchase(\mathcal{I}, \mathcal{D}, m)$, let j be an integer such that $0 \leq j < m$, let t_1, t_2 be Skolem terms, and let ϕ be the outermost Skolem function of t_1. If t_1 is j-ancestor of t_2 in $B(f)$, then ϕ occurs at depth $j + 1$ in t_2.*

We now prove that, for every pair of terms t''_1 and t''_2 occurring in $B(f)$ and involved in the relation τ, t''_1 and t''_2 are connected through the ancestor and sibling relations in $B(f)$.

Lemma 9. *Let F be a set of facts such that $F \subseteq fchase(\mathcal{I}, \mathcal{D}, m)$, let $j < m$, let $f \in F$ and let $\langle f, t_1, t''_1\rangle \in \tau_j$, $\langle f, t_2, t''_2\rangle \in \tau_j$. Then, one of the following conditions holds:*

1. *t''_1 is a h-ancestor of t''_2 in $B(f)$ for some h such that $0 \le h \le j+1$;*
2. *t''_2 is a h-ancestor of t''_1 in $B(f)$ for some h such that $0 \le h \le j+1$;*
3. *there exist t, t' such that $\langle f, t, t'\rangle \in \tau$ and t''_1 and t' are siblings in $B(f)$ and t' is a h-ancestor of t''_2 in $B(f)$ for some h such that $0 \le h \le j+1$;*
4. *there exist t, t' such that $\langle f, t, t'\rangle \in \tau$ and t''_2 and t' are siblings in $B(f)$ and t' is a h-ancestor of t''_1 in $B(f)$ for some h such that $0 \le h \le j+1$.*

Proof. First, if $t''_1 = t''_2$, then t''_1 is 0-ancestor of itself in $B(f)$, hence condition 1 of the claim holds. So suppose $t''_1 \neq t''_2$. We prove the claim by induction on the structure of τ.

Base case: if $\langle f, t_1, t''_1\rangle \in \tau_0$ and $\langle f, t_2, t''_2\rangle \in \tau_0$, then $t''_1 = t_1$, $t''_2 = t_2$ and t_1 and t_2 occur in f. Thus by Lemma 6, one of the following cases holds: (i) t_1 is the k-predecessor of t_2 in $B(f)$ for some k, which implies that condition 1 of the claim holds (with $h = 1$); (ii) t_2 is the k-predecessor of t_1 in $B(f)$ for some k, which implies that condition 2 of the claim holds (with $h = 1$); (iii) t_1 and t_2 are siblings in $B(f)$, which implies that condition 3 of the claim holds (with $t' = t_2$ and since t_2 is 0-ancestor of itself in $B(f)$). Thus, the claim follows.

Inductive case: suppose the claim holds for τ_i with $1 \le i < j$, and let $\tau_{i+1} = \tau_i \cup \{\langle f, t_1, t''_1\rangle\}$. Then, according to Definition 5 there are two possible cases:

Case A. There exist $f', t'_1, t_3, t'_3, t''_3, k$ such that $\langle f', t_1, t'_1\rangle \in \tau_i$, $\langle f', t_3, t'_3\rangle \in \tau_i$, t'_1 is the k-predecessor of t'_3 in $B(f')$, $\langle f, t_3, t''_3\rangle \in \tau_i$, and t''_1 is the k-predecessor of t''_3 in $B(f)$. In this case, let $\langle f, t_2, t''_2\rangle \in \tau_i$, let g_1 be the last introduction point of t''_1 in $B(f)$, let g_2 be the last introduction point of t''_2 in $B(f)$, and let g_3 be the last introduction point of t''_3 in $B(f)$. By the inductive hypothesis, t''_2 and t''_3 are such that one of the following four cases holds:

A1. t''_2 is a h-ancestor of t''_3 in $B(f)$ for some h such that $0 \le h \le i+1$. In this case, there exists a sequence of terms $s_0, s_1, \ldots, s_h$ such that

$s_0 = t''_2$, $s_h = t''_3$ and for every ℓ such that $0 \leq \ell \leq h-1$, s_ℓ is the k_ℓ-predecessor of $s_{\ell+1}$ in $B(f)$ for some k_ℓ. Now, there are three possible cases:

A1.1. g_1 precedes g_2 in $B(f)$. In this case, since t''_1 is a predecessor of t''_3 in $B(f)$, t''_1 occurs in all facts that lie between g_1 and g_3 in $B(f)$, therefore t''_1 occurs in g_2, i.e., t''_1 is a predecessor of t''_2 in $B(f)$. Hence, condition 1 of the claim holds;

A1.2. $g_1 = g_2$. In this case, t''_1 and t''_2 are siblings in $B(f)$, which implies that condition 3 of the claim holds;

A1.3. g_1 follows g_2 in $B(f)$. In this case, since in every fact g that lies between g_2 and g_3 in $B(f)$ there exists an s_ℓ with $0 \leq \ell \leq h-1$ such that s_ℓ occurs in g in a predecessor position (i.e., s_ℓ is not introduced in g), it follows that such an s_ℓ occurs in g_1. Therefore, such an s_ℓ is a predecessor of t''_1 in $B(f)$, which implies that t''_2 is a h'-ancestor of t''_1 in $B(f)$ with $h' = h - \ell$. Hence, condition 2 of the claim holds;

A2. t''_3 is a h-ancestor of t''_2 in $B(f)$ for some h such that $0 \leq h \leq i+1$. In this case, since t''_1 is a predecessor of t''_3 in $B(f)$, it follows that t''_1 is h+1-ancestor of t''_2 in $B(f)$, thus condition 1 of the claim holds (because $h+1 \leq (i+1)+1$);

A3. there exist t, t' such that $\langle f, t, t' \rangle \in \tau_i$ and t''_2 and t' are siblings in $B(f)$ and t' is a h-ancestor of t''_3 in $B(f)$ for some h such that $0 \leq h \leq i+1$. Again, there are three possible cases:

A3.1. g_1 precedes g_2 in $B(f)$. In this case, since t''_1 is a predecessor of t''_3 in $B(f)$, t''_1 occurs in all facts that lie between g_1 and g_3 in $B(f)$, therefore t''_1 occurs in g_2, i.e., t''_1 is a predecessor of t''_2 in $B(f)$. Hence, condition 1 of the claim holds;

A3.2. $g_1 = g_2$. In this case, t''_1 and t''_2 are siblings in $B(f)$, which implies that condition 3 of the claim holds;

A3.3. g_1 follows g_2 in $B(f)$. In this case, since t' is a h-ancestor of t''_3 in $B(f)$, there exists a sequence of terms $s'_0, s'_1, \ldots, s'_h$ such that $s'_0 = t'$, $s'_h = t''_3$ and for every ℓ such that $0 \leq \ell \leq h-1$, s'_ℓ is the k_ℓ-predecessor of $s'_{\ell+1}$ in $B(f)$ for some k_ℓ. Now, since in every fact g that lies between g_2 and g_3 in $B(f)$ there exists an s'_ℓ with

$0 \leq \ell \leq h-1$ such that s'_ℓ occurs in g in a predecessor position (i.e., s'_ℓ is not introduced in g), it follows that such an s'_ℓ occurs in g_1. Therefore, such an s'_ℓ is a predecessor of t''_1 in $B(f)$, which implies that t' is a h'-ancestor of t''_1 in $B(f)$ with $h' = h-\ell$. Hence, condition 4 of the claim holds;

A4. there exist t, t' such that $\langle f, t, t' \rangle \in \tau_i$ and t''_3 and t' are siblings in $B(f)$ and t' is a h-ancestor of t''_2 in $B(f)$ for some h such that $0 \leq h \leq i+1$. In this case, since t''_1 is the k-predecessor of t''_3 in $B(f)$, it immediately follows that t''_1 is the k-predecessor of t' in $B(f)$, hence t''_1 is a $h+1$-ancestor of t''_2 in $B(f)$. Therefore, condition 1 of the claim holds (since $h+1 \leq (i+1)+1$).

Case B. there exist $f', t'_1, t_3, t'_3, t''_3, p, q$ such that $\langle f', t_1, t'_1 \rangle \in \tau_i$, $\langle f', t_3, t'_3 \rangle \in \tau_i$, t'_1 and t'_3 are p-q-siblings in $B(f')$, $\langle f, t_3, t''_3 \rangle \in \tau_i$, and t''_1 and t''_3 are p-q-siblings in $B(f)$. In this case, let $\langle f, t_2, t''_2 \rangle \in \tau_i$, let g_1 be the last introduction point of t''_1 in $B(f)$, let g_2 be the last introduction point of t''_2 in $B(f)$, and let g_3 be the last introduction point of t''_3 in $B(f)$. By the inductive hypothesis, t''_2 and t''_3 are such that one of the following four cases holds:

B1. t''_2 is a h-ancestor of t''_3 in $B(f)$ for some h such that $0 \leq h \leq i+1$. In this case, we have that t''_2 is also h-ancestor of t''_1 in $B(f)$, thus condition 2 of the claim holds;

B2. t''_3 is a h-ancestor of t''_2 in $B(f)$ for some h such that $0 \leq h \leq i+1$. In this case, it immediately follows that condition 3 of the claim holds;

B3. there exist t, t' such that $\langle f, t, t' \rangle \in \tau$ and t''_2 and t' are siblings in $B(f)$ and t' is a h-ancestor of t''_3 in $B(f)$ for some h such that $0 \leq h \leq i+1$. Again, in this case we have that t' is also a h-ancestor of t''_1 in $B(f)$, therefore condition 4 of the claim holds;

B4. there exist t, t' such that $\langle f, t, t' \rangle \in \tau$ and t''_3 and t' are siblings in $B(f)$ and t' is a h-ancestor of t''_2 in $B(f)$ for some h such that $0 \leq h \leq i+1$. In this case, it immediately follows that t''_1 and t' are siblings in $B(f)$, therefore condition 3 of the claim holds.

□

Using the above lemmas, we are now able to prove that τ is actually a binary function, provided that m is sufficiently greater than the cardinality of F and $JST(F)$.

Lemma 10. *Let F be a set of facts such that $F \subseteq \mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ and $m \geq 2(|F| \times |JST(F)|) + 2$. Let $\langle f, t, t' \rangle \in \tau$, $\langle f, t, t'' \rangle \in \tau$. Then, $t' = t''$.*

Proof. We prove the lemma by showing that, at every step $j + 1$ of the bottom-up construction of τ, if $\langle f, t, t' \rangle \in \tau_j$ then it is not possible to add a triple of the form $\langle f, t, t'' \rangle$ with $t' \neq t''$. So, suppose conversely that τ contains such a pair of triples and let the $(j+1)$-th iteration in the bottom-up inductive definition of τ be the first iteration in which the triple inserted $\langle f, t, t'' \rangle$ is such that there exists a triple of the form $\langle f, t, t' \rangle$ (for some term t') in τ_j. Observe that the bottom-up definition of τ implies that $t' \neq t''$. Suppose that $j \leq |F| \times |JST(F)|$: thus, $j \leq m - \ell$, where we define $\ell = |F| \times |JST(F)| + 2$. Therefore, by Lemma 7 we have that $t \overset{\ell}{=} t'$ and $t \overset{\ell}{=} t''$, which implies $t' \overset{\ell}{=} t''$. So let ϕ be the outermost Skolem function of both t' and t''. Now, from Lemma 9, one of the following cases holds:

1. t' and t'' are siblings in $B(f)$. But in this case, by definition of *fchase* the outermost Skolem functions of t' and t'' should be different, thus contradicting the above conclusion $t' \overset{2}{=} t''$. Consequently, this case cannot occur;
2. t' is a h-ancestor of t'' in $B(f)$ for some h such that $1 \leq h \leq j + 1$ (notice that the case $h = 0$ is impossible since we have assumed $t' \neq t''$). In this case, since $j \leq m - \ell$, we have that $h + 1 \leq m - \ell + 2$, and since $\ell \geq 2$, it follows that $h + 1 \leq m$, thus by Lemma 8 ϕ (i.e., the outermost Skolem function of t') occurs at depth $h+1$ in t''. But this is impossible, since ϕ is also the outermost Skolem function of t'' and, by Definition 2, the outermost Skolem function of a Skolem term cannot occur elsewhere in the term. Consequently, this case cannot occur;
3. t'' is a h-ancestor of t' in $B(f)$ for some h such that $1 \leq h \leq j + 1$. By an argument identical to the previous case, it follows that this case cannot occur;
4. there exist t_1, t'_1 such that $\langle f, t_1, t'_1 \rangle \in \tau$ and t' and t'_1 are siblings in $B(f)$ and t'_1 is a h-ancestor of t'' in $B(f)$ for some h such that $0 \leq h \leq j+1$. In this case, let t' be of the form $\phi(\alpha_1, \ldots, \alpha_k)$. Then, since t' and t'_1 are siblings in $B(f)$, t'_1 is of the form $\psi(\alpha_1, \ldots, \alpha_k)$. Moreover, since $t' \overset{\ell}{=} t''$, t'' has the form $\phi(\beta_1, \ldots, \beta_k)$ with $\alpha_i \overset{\ell-1}{=} \beta_i$ for every i such that $1 \leq i \leq k$ (recall that $\ell \geq 2$). Now, as shown above, $h + 1 \leq m$, so by Lemma 8 we have that ψ occurs at depth $h + 1$ in t''. Let β_h be the subterm of t'' in which ψ occurs. Then, from $\alpha_i \overset{\ell-1}{=} \beta_i$ and since

$h+1 \leq \ell-1$, it follows that ψ also occurs in α_h. But this contradicts Definition 2 which implies that the outermost Skolem function of a Skolem term cannot occur elsewhere in the term. Consequently, this case cannot occur;

5. there exist t_1, t_1' such that $\langle f, t_1, t_1' \rangle \in \tau$ and t'' and t_1' are siblings in $B(f)$ and t_1' is a h-ancestor of t' in $B(f)$ for some h such that $1 \leq h \leq j+1$. By an argument identical to the previous case, it follows that this case cannot occur.

Thus, we have proved that, for j such that $j \leq |F| \times |JST(F)|$, if $\langle f, t, t' \rangle \in \tau_j$ then, it is impossible that τ_{j+1} contains a triple $\langle f, t, t'' \rangle$ with $t' \neq t''$. But this immediately implies that, for $j = |F| \times |JST(F)|$, $\tau_j = \tau_{j+1}$, i.e., $\tau = \tau_j$, which in turn implies the thesis. □

We now introduce a notion of completeness for the relation τ. We say that τ is *complete* if the following conditions hold: (i) if $\langle f', t_1, t_1' \rangle \in \tau$ and $\langle f', t_2, t_2' \rangle \in \tau$ and $\langle f, t_2, t_2'' \rangle \in \tau$ and t_1' is the k-predecessor of t_2' in $B(f')$, then there exists t_1'' such that t_1'' is the k-predecessor of t_2'' in $B(f)$ (and thus $\langle f', t_1, t_1'' \rangle \in \tau$); (ii) if $\langle f', t_1, t_1' \rangle \in \tau$ and $\langle f', t_2, t_2' \rangle \in \tau$ and $\langle f, t_2, t_2'' \rangle \in \tau$ and t_1' and t_2' are p-q-siblings in $B(f')$, then there exists t_1'' such that t_1'' and t_2'' are p-q-siblings in $B(f)$ (and thus $\langle f', t_1, t_1'' \rangle \in \tau$).

Notice that τ may not be complete: e.g., it may be the case that t_1' is the k-predecessor of t_2' in $B(f')$ but t_2'' has no k-predecessor in $B(f)$ (for instance, if t_2'' is a constant, it has neither predecessors nor siblings in $B(f)$).

It is easy to show that, under the same conditions of the previous lemma, τ is complete.

Lemma 11. *Let F be a set of facts such that $F \subseteq fchase(\mathcal{I}, \mathcal{D}, m)$ and $m \geq 2(|F| \times |JST(F)|) + 2$. Then, τ is complete.*

Proof. By Lemma 10, for $j = |F| \times |JST(F)|$, $\tau_j = \tau_{j+1}$, thus from the hypothesis and Lemma 7 it follows that, for every $\langle f, t, t' \rangle \in \tau$, $t \stackrel{\ell}{=} t'$ with $\ell = |F| \times |JST(F)| + 2$. Since $\ell \geq 2$, this in turn implies that, if $\langle f', t_1, t_1' \rangle \in \tau$ and $\langle f', t_2, t_2' \rangle \in \tau$ and $\langle f, t_2, t_2'' \rangle \in \tau$ and t_1' is the k-predecessor of t_2' in $B(f')$, then we have $t_2 \stackrel{2}{=} t_2'$ and $t_2 \stackrel{2}{=} t_2''$, consequently $t_2' \stackrel{2}{=} t_2''$, which implies by Definition 2 that there exists t_1'' such that t_1'' is the k-predecessor of t_2'' in $B(f)$ (the same proof holds in the case when t_1' and t_2' are p-q-siblings in $B(f')$). □

From now on, we will always assume the conditions under which τ is a binary function and is complete, i.e., the integer m and the set of facts F of $fchase(\mathcal{I}, \mathcal{D}, m)$ are such that $m \geq 2(|F| \times |JST(F)|) + 2$. For every $f \in F$, we denote by τ_f the following function:

$$\tau_f(t) = \{\langle t, t' \rangle \mid \langle f, t, t' \rangle \in \tau\}$$

Moreover, we denote by $RJST(f)$ the subset of $JST(F)$ defined as follows:

$$RJST(f) = \{t \mid \langle t, t' \rangle \in \tau_f\}$$

Finally, given a set of facts F' such that $F' \subseteq F$, we define $RJST(F') = \bigcup_{f \in F'} RJST(f)$.

Therefore, the function τ_f provides a mapping of a subset of $JST(F)$ to a subset of the Skolem terms occurring in $B(f)$. The domain of τ_f, i.e., the set $RJST(f)$, is called the set of *relevant join Skolem terms of* f.

Example 5. Let the following be sub-branches of a finite chase $fchase(\mathcal{I}, \mathcal{D}, m)$:

$$\begin{array}{ccc} \vdots & \qquad & \vdots \\ f_1 = r(t_o, t_1) & & f_4 = r(t'_o, t'_1) \\ \downarrow & & \downarrow \\ f_2 = s(t_1, t_2) & & f_5 = s(t'_1, t'_2) \\ \downarrow & & \downarrow \\ f_3 = u(t_2, t_3) & & f_6 = u(t'_2, t_3) \end{array}$$

where $t_1, t_2, t_3, t'_1, t'_2, t'_3$ are Skolem terms of depth m. Now let $F = \{f_1, \ldots, f_6\}$. Then, we have

$$JST(F) = \{t_1, t_2, t_3, t'_1, t'_2\}$$

Now, following the bottom-up definition of τ, we have:

$$\begin{array}{l} \tau_0 = \{\langle f_1, t_1, t_1 \rangle, \langle f_2, t_1, t_1 \rangle, \langle f_2, t_2, t_2 \rangle, \langle f_3, t_2, t_2 \rangle, \langle f_3, t_3, t_3 \rangle, \\ \qquad \langle f_4, t'_1, t'_1 \rangle, \langle f_5, t'_1, t'_1 \rangle, \langle f_5, t'_2, t'_2 \rangle, \langle f_6, t'_2, t'_2 \rangle, \langle f_6, t_3, t_3 \rangle\} \end{array}$$

Then, one possible bottom-up construction of τ is the following:

- $\tau_1 = \tau_0 \cup \{\langle f_3, t_1, t_1 \rangle\}$, since $\langle f_2, t_1, t_1 \rangle \in \tau_0$, $\langle f_2, t_2, t_2 \rangle \in \tau_0$, $\langle f_3, t_2, t_2 \rangle \in \tau_0$, t_1 is the 1-predecessor of t_2 in $B(f_2)$ and t_1 is the 1-predecessor of t_2 in $B(f_3)$;

- $\tau_2 = \tau_1 \cup \{\langle f_6, t'_1, t'_1 \rangle\}$, since $\langle f_5, t'_1, t'_1 \rangle \in \tau_1$, $\langle f_5, t'_2, t'_2 \rangle \in \tau_1$, $\langle f_6, t'_2, t'_2 \rangle \in \tau_1$, t'_1 is the 1-predecessor of t'_2 in $B(f_5)$ and t'_1 is the 1-predecessor of t'_2 in $B(f_6)$;

- $\tau_3 = \tau_2 \cup \{\langle f_3, t'_2, t_2 \rangle\}$, since $\langle f_6, t'_2, t'_2 \rangle \in \tau_2$, $\langle f_6, t_3, t_3 \rangle \in \tau_2$, $\langle f_3, t_3, t_3 \rangle \in \tau_2$, t'_2 is the 1-predecessor of t_3 in $B(f_6)$ and t_2 is the 1-predecessor of t_3 in $B(f_3)$;

- $\tau_4 = \tau_3 \cup \{\langle f_6, t_2, t'_2 \rangle\}$, since $\langle f_3, t_2, t_2 \rangle \in \tau_3$, $\langle f_3, t_3, t_3 \rangle \in \tau_3$, $\langle f_6, t_3, t_3 \rangle \in \tau_3$, t_2 is the 1-predecessor of t_3 in $B(f_3)$ and t'_2 is the 1-predecessor of t_3 in $B(f_6)$;

- $\tau_5 = \tau_4 \cup \{\langle f_3, t'_1, t_1 \rangle\}$, since $\langle f_6, t'_1, t'_1 \rangle \in \tau_4$, $\langle f_6, t_2, t'_2 \rangle \in \tau_4$, $\langle f_3, t_2, t_2 \rangle \in \tau_4$, t'_1 is the 1-predecessor of t'_2 in $B(f_6)$ and t_1 is the 1-predecessor of t_2 in $B(f_3)$;

- $\tau_6 = \tau_5 \cup \{\langle f_6, t_1, t'_1 \rangle\}$, since $\langle f_3, t'_1, t_1 \rangle \in \tau_5$, $\langle f_3, t'_2, t_2 \rangle \in \tau_5$, $\langle f_6, t'_2, t'_2 \rangle \in \tau_5$, t_1 is the 1-predecessor of t_2 in $B(f_3)$ and t'_1 is the 1-predecessor of t'_2 in $B(f_6)$;

- $\tau_7 = \tau_6 \cup \{\langle f_2, t'_1, t_1 \rangle\}$, since $\langle f_3, t'_1, t_1 \rangle \in \tau_6$, $\langle f_3, t_2, t_2 \rangle \in \tau_6$, $\langle f_2, t_2, t_2 \rangle \in \tau_6$, t_1 is the 1-predecessor of t_2 in $B(f_3)$ and t_1 is the 1-predecessor of t_2 in $B(f_2)$;

- $\tau_8 = \tau_7 \cup \{\langle f_5, t_1, t'_1 \rangle\}$, since $\langle f_6, t_1, t'_1 \rangle \in \tau_7$, $\langle f_6, t'_2, t'_2 \rangle \in \tau_7$, $\langle f_5, t'_2, t'_2 \rangle \in \tau_7$, t'_1 is the 1-predecessor of t'_2 in $B(f_6)$ and t'_1 is the 1-predecessor of t'_2 in $B(f_5)$;

- $\tau = \tau_8$.

Therefore,

$$\begin{aligned}
RJST(f_1) &= \{t_1\} \\
RJST(f_2) &= \{t_1, t_2, t'_1\} \\
RJST(f_3) &= \{t_1, t_2, t_3, t'_1, t'_2\} \\
RJST(f_4) &= \{t'_1\} \\
RJST(f_5) &= \{t'_1, t'_2, t_1\} \\
RJST(f_6) &= \{t'_1, t'_2, t_3, t_1, t_2\}
\end{aligned}$$

It is interesting to observe that the set $RJST(f)$ not only comprises join Skolem terms occurring in the branch $B(f)$, but also collects join Skolem terms from other branches of the finite chase: for example, $RJST(f_3)$ contains the terms t'_1 and t'_2 which do not occur in $B(f_3)$, while $RJST(f_6)$ contains the terms t_1 and t_2 which do not occur in $B(f_6)$. Then, every function τ_f maps every term of $RJST(f)$ into a Skolem term (not necessarily from $JST(F)$) occurring in $B(f)$: for instance, τ_{f_3} maps the join Skolem term t'_1 into t_1 and t'_2 into t_2, while τ_{f_6} maps the join Skolem term t_1 into t'_1 and t_2 into t'_2. ∎

We now try to provide an intuitive explanation of the usage of the function τ that we will make in the proof of soundness of the finite chase. As already mentioned, for each fact $f \in F$, the set $RJST(f)$ constitutes the set of all join Skolem terms from $JST(F)$ that are relevant for the fact f. The reason why we call the terms in $RJST(f)$ "relevant join Skolem term for f" will be completely clear only in the light of Lemma 12: the intuition, however, can be explained through a simple example.

Example 6. Consider again Example 5. Then, suppose we want to identify a portion of the canonical chase *chase*$(\mathcal{I}, \mathcal{D})$ with facts $f'_1, \ldots, f'_6$ that are homomorphic to $f_1, \ldots, f_6$ (this is of course crucial for proving soundness of the finite chase). Now, it can be shown that the fact that f_3 and f_6 have t_3 as their second argument implies that f'_3 *and* f'_6 *must have the same first argument*: this is reflected in the definition of τ_{f_3}, which states that t'_2 must be mapped to t_2 in $B(f_3)$, and in the definition of τ_{f_6}, which states that t_2 must be mapped to t'_2 in $B(f_6)$. This in turn implies that f'_2 and f'_5 must have the same first argument, and this is also reflected in the definition of τ_{f_2}, which states that t'_1 must be mapped to t_1 in $B(f_2)$, and in the definition of τ_{f_5}, which states that t_1 must be mapped to t'_1 in $B(f_5)$. Therefore, t'_1 must be considered as a relevant join Skolem term for f_3, because t_1 occurs in f_1 and t'_1 and t_1 are forced to have an identical mapping on the canonical chase. For the same reason, for instance, t_1 must be considered as a relevant join Skolem term for f_6. ∎

Informally, Lemma 7, Lemma 10 and Lemma 11 guarantee that, for every fact $f \in F$ and for every join Skolem term t of F that is relevant for f, the branch $B(f)$ of *fchase*$(\mathcal{I}, \mathcal{D}, m)$ contains a Skolem term $\tau_f(t)$ that "correctly" represents t, in the sense that $t \stackrel{\ell}{=} \tau_f(t)$ where $\ell = |F| \times |JST(F)| + 2$. This property is actually crucial for the subsequent Lemma 12, since it allows us to map all the branches of *fchase*$(\mathcal{I}, \mathcal{D}, m)$ relative to the facts in F in the canonical chase *chase*$(\mathcal{I}, \mathcal{D})$, in a way such that it can be proved that the set of facts corresponding to the leaves of such branches of $\mathit{chase}(\mathcal{I}, \mathcal{D})$ has a homomorphism to F.

To formally state such a property, we need the following crucial notion of *last relevant join Skolem term* of a fact f with respect to a set of facts F'.

Definition 6. *Let F be a set of facts such that $F \subseteq \mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ and $m \geq 2(|F| \times |JST(F)|) + 2$. Given a fact $f \in F$ and a set of facts $F' \subseteq F$, we define the function $\mathit{LastRJST}(f, F')$ as follows. If $RJST(f) \cap RJST(F') \neq \emptyset$, then we define $\mathit{LastRJST}(f, F')$ as the Skolem term t such that:*

i. $t \in RJST(f) \cap RJST(F')$;

ii. there exists no $t' \in RJST(f) \cap RJST(F')$ such that the last introduction point of $\tau_f(t')$ in $B(f)$ is subsequent to the last introduction point of $\tau_f(t)$ in $B(f)$;

iii. if there exists $t' \in RJST(f) \cap RJST(F')$ such that $\tau_f(t)$ and $\tau_f(t')$ are p-q-siblings in $B(f)$ (which implies that the last introduction point of $\tau_f(t')$ in $B(f)$ is equal to the last introduction point of $\tau_f(t)$ in $B(f)$), then $p < q$.

If otherwise $RJST(f) \cap RJST(F') = \emptyset$, then we define $\mathit{LastRJST}(f, F') = \bot$.

Example 7. Consider a pair $\mathcal{I}$, $\mathcal{D}$ such that $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ contains the following branches:

$$
\begin{array}{ccccc}
 & & \vdots & & \\
 & & r(t') & & \\
 & & \downarrow & & \\
 & & s(t', t_1) & & \\
 & & \downarrow & & \\
 & & u(t_1, t_2) & & \\
 & & \downarrow & & \\
(B2) & q(t_2, t_3) \leftarrow & v(t_1, t_2, t_3) & \rightarrow z(t_1, t_2, t'') & (B3) \\
 & & \downarrow & & \\
 & & w(t_1, t_3) & & \\
 & & (B1) & &
\end{array}
$$

where t_1, t_2, t_3, t', t'' are Skolem terms of depth m. Now let $g_1 = w(t_1, t_3)$ $g_2 = q(t_2, t_3)$, and $g_3 = z(t_1, t_2, t'')$, and let $F = \{g_1, g_2, g_3\}$. Then, we have the following:

$$\begin{array}{rcl} JST(F) & = & \{t_1, t_2, t_3\} \\ RJST(g_1) & = & \{t_1, t_2, t_3\} \\ RJST(g_3) & = & \{t_1, t_2\} \\ RJST(g_1) \cap RJST(g_3) & = & \{t_1, t_2\} \end{array}$$

Now, $LastRJST(g_3, \{g_1\})$ is the Skolem term t_2, since the last introduction point of t_2 in B3 (i.e., the fact $u(t_1, t_2)$) follows the last introduction point of t_1 in B3 (i.e., the fact $s(t', t_1)$). ■

Informally, the intersection of $RJST(f)$ and $RJST(F')$ represents the set of relevant join Skolem terms that f "shares" with F', in the sense that such terms are relevant both for f and for some $f' \in F'$, while $LastRJST(f, F')$ represents the join Skolem term t belonging to $RJST(f) \cap RJST(F')$ that is the "last" or "most recent" one in $B(f)$, in the sense that the last introduction point of $\tau_f(t)$ in $B(f)$ does not precede the last introduction point of $\tau_f(t')$ for every other t' belonging to $RJST(f) \cap RJST(F')$.

In the following, we denote by $Branches(chase(\mathcal{I}, \mathcal{D}))$ the set of branches of $chase(\mathcal{I}, \mathcal{D})$, we denote by $Branches(fchase(\mathcal{I}, \mathcal{D}, m))$ the set of branches of $fchase(\mathcal{I}, \mathcal{D}, m)$, and we call $Terms(chase(\mathcal{I}, \mathcal{D}))$ the set of all terms occurring in $chase(\mathcal{I}, \mathcal{D})$.

Definition 7. *Let $S = \langle f_1, \ldots, f_n \rangle$ be a sequence of distinct facts from $fchase(\mathcal{I}, \mathcal{D}, m)$ let F be the set $\{f_1, \ldots, f_n\}$, and let S be such that $m \geq 2(n \times |JST(F)|) + 2$. We define:*

- *the function $Bfin^S : JST(F) \rightarrow Branches(fchase(\mathcal{I}, \mathcal{D}, m))$,*
- *the function $Bcan^S : JST(F) \rightarrow Branches(chase(\mathcal{I}, \mathcal{D}))$,*
- *the function $IP^S : JST(F) \rightarrow chase(\mathcal{I}, D)$,*
- *the function $MapCan^S : fchase(\mathcal{I}, \mathcal{D}, m) \rightarrow chase(\mathcal{I}, \mathcal{D})$,*
- *and the function $h^S : JST(F) \rightarrow Terms(chase(\mathcal{I}, \mathcal{D}))$*

as follows. For each i such that $1 \leq i \leq n$:

1. *if $LastRJST(f_i, \{f_1, \ldots, f_{i-1}\}) = \bot$ then:*

- *$Bfin^S(f_i)$ is defined as $B(f_i)$;*
- *$Bcan^S(f_i)$ is defined as the branch of $chase(I, D)$ obtained by executing the sequence of IDs corresponding to the branch $Bfin^S(f_i)$ in $fchase(\mathcal{I}, \mathcal{D}, m)$ starting from the fact of $\mathcal{D}$ that is the root of $B(f_i)$;*

otherwise, if $LastRJST(f_i, \{f_1, \ldots, f_{i-1}\}) = t'$, then:

- *$Bfin^S(f_i)$ is defined as the (sub-)branch of $B(f_i)$ starting from the last introduction point of $\tau_{f_i}(t')$ in $B(f_i)$ (and ending in f);*
- *$Bcan^S(f_i)$ is defined as the (sub-)branch of $chase(I, D)$ corresponding to $Bfin^S(f_i)$ and starting from $IP^S(t')$;*

2. *for every fact $g \in Bfin^S(f_i)$, $MapCan^S(g)$ is defined as the fact corresponding to g in $Bcan^S(f_i)$;*
3. *for each $t \in RJST(f_i) - RJST(\{f_1, \ldots, f_{i-1}\})$:*
   - *$IP^S(t)$ is defined as $MapCan^S(g)$, where g denotes the last introduction point of $\tau_{f_i}(t)$ in $B(f_i)$;*
   - *$h^S(t)$ is defined as the term occurring in $IP^S(t)$ in the same argument position as $\tau_{f_i}(t)$ in g, where g denotes the last introduction point of $\tau_{f_i}(t)$ in $B(f_i)$.*

Finally, we define $IM^S = \bigcup_{1 \leq i \leq n} MapCan^S(f_i)$.

The goal of the above definition is to identify a set IM^S of facts of $chase(\mathcal{I}, \mathcal{D})$ and a function h^S such that h^S is a homomorphism from F to IM^S. As explained in Section 3.4, the set IM^S is identified by mapping (through the function $MapCan^S$) the portion of the finite chase relative to F (i.e., the branches of the finite chase whose leaves are the facts in F) to a portion of the canonical chase: in practice, for every $f \in F$, the branch $B(f)$ of the finite chase (more precisely, the branch $Bfin^S(f)$) is mapped to a "corresponding" branch of the canonical chase (the branch $Bcan^S(f)$). The problem here is due to the presence of the join Skolem terms of depth m, i.e., the Skolem terms in $JST(F)$: such terms are problematic because two occurrences of the same term t in different branches of the finite chase might refer to two different introduction points of t (in other words, the terms in $JST(F)$ might be introduced multiple times in the finite chase). This makes it generally incorrect to directly map every branch of the finite chase into the "naturally corresponding" branch of the canonical chase (as defined in Section 3.4), because two occurrences of such a term t in two different branches of the finite chase (or even in a single branch) may correspond to two different Skolem terms in the corresponding branches of the canonical chase.

To satisfy all joins among Skolem terms of depth m in F also in IM^S and map every join Skolem term t of $JST(F)$ to a single Skolem term of the canonical chase, the functions $Bfin^S$, $Bcan^S$ and $MapCan^S$ are defined as follows.

When considering the first fact f_1 of the sequence S, $Bfin^S(f_1)$ is defined as the whole branch $B(f_1)$ and $Bcan^S(f_1)$ is obtained as explained above, by considering the branch that starts from the fact of D that is the root of $B(f_1)$ and has been generated by applying the same sequence of IDs used in the generation of $B(f_1)$. Let f_1' be the leaf of $Bcan^S(f_1)$. Then, $MapCan^S(f_1) = f_1'$ (more generally, $MapCan^S$ maps every fact of $B(f_1)$ to the corresponding fact of $Bcan^S(f_1)$). Then, the function τ_{f_1} is taken into account, which *identifies, in the branch* $B(f_1)$ *of the finite chase, a term* $\tau_{f_1}(t)$ *of* $B(f_1)$ *for every Skolem term* t *such that* $t \in RJST(f_1)$. In turn, $\tau_{f_1}(t)$ is used to identify a corresponding Skolem term $h^S(t)$ in the branch of the canonical chase corresponding to $B(f_1)$. From now on, the definition will enforce the use of $h^S(t)$ when mapping the other facts of F in which t occurs.

When mapping the fact f_i, with $i > 1$, there are two possibilities:

i. f_i is such that $LastRJST(f_i, \{f_1, \ldots, f_{i-1}\}) = \bot$. This condition implies that the mapping of f_i does not involve join Skolem terms that have been already mapped to terms of $chase(\mathcal{I}, \mathcal{D})$, hence the term f_i can be mapped exactly like before, by mapping the whole branch of the finite chase $B(f_i)$ on the "naturally corresponding" branch of $chase(\mathcal{I}, \mathcal{D})$, i.e., the branch that starts from the root of $B(f_i)$ and is generated by applying the same sequence of IDs used in the generation of $B(f_i)$;

ii. f_i is such that $LastRJST(f_i, \{f_1, \ldots, f_{i-1}\}) = t'$ for some Skolem term t'. In this case, f_i must be mapped to a corresponding fact f_i' of $chase(\mathcal{I}, \mathcal{D})$ in such a way that f_i' makes use of the images $h^S(t)$ of all the terms t of $JST(F)$ that have already been mapped on the portion of $chase(\mathcal{I}, \mathcal{D})$ identified in the mapping of $f_1, \ldots, f_{i-1}$. This is realized by considering the relevant join Skolem terms of f_i, and selecting among such join Skolem terms the "most recent" (or "last") term that has already been mapped through the function h^S: this is formally defined by the function *LastRJST* of Definition 6. So let $t = \mathit{LastRJST}(f_i, \{f_1, \ldots, f_{i-1}\})$: then, let g be the fact that is the last introduction point of $\tau_{f_i}(t)$ in $B(f_i)$ and let g' be the fact that is the introduction point of the term $h^S(t)$ in the canonical chase. The fact g' constitutes the "linking point" of $B(f_i)$ in the canonical chase, in the sense that, to generate f'_i, the sequence of IDs of the portion of the branch $B(f_i)$ of the finite chase that starts from the fact g is applied, starting from the fact g'. In this way, the image $h^S(t)$ of every join Skolem term of $JST(F)$ that has been already mapped is correctly reused, and thus all joins between f_i and the facts already considered in the construction are correctly satisfied in the new fact f'_i thus generated.

Obviously, in the construction of f'_i, further join Skolem terms from $JST(F)$ may be mapped by the function h^S. By iterating the above procedure, we end up with a set of facts $IM^S = \{f'_1, \ldots, f'_n\}$ and a function h^S such that h^S is (more precisely, can be extended to) a homomorphism from F to IM^S, as we will show in the next lemma.

Example 8. We briefly illustrate Definition 7 referring to Example 7 and to the sequence of facts $S = \langle g_1, g_2, g_3 \rangle$ with $g_1 = w(t_1, t_3)$, $g_2 = z(t_1, t_2, t'')$ and $g_3 = q(t_2, t_3)$. Of course, $\mathit{LastRJST}(g_1, \emptyset) = \bot$, so $\mathit{Bfin}^S(g_1) = B(g_1) =$ B1 and $\mathit{Bcan}^S(g_1)$ is the branch of the canonical chase that has the same root as the root of branch B1 and whose sequence of IDs coincides with the sequence of IDs used to generate branch B1 of the finite chase. Then, $RJST(\{g_1\}) = \{t_1, t_2, t_3\}$, thus $IP^S(t_1)$ is the fact in $\mathit{Bcan}^S(g_1)$ that corresponds to the fact $s(t', t_1)$ in B1 (let p_1 be such a fact of the canonical chase), $IP^S(t_2)$ is the fact in $\mathit{Bcan}^S(g_1)$ that corresponds to the fact $u(t_1, t_2)$ in B1 (let p_2 be such a fact of the canonical chase), and $IP^S(t_3)$ is the fact in $\mathit{Bcan}^S(g_1)$ that corresponds to the fact $v(t_1, t_2, t_3)$ in B1 (let p_3 be such a fact of the canonical chase). Let g'_1 be the leaf of $\mathit{Bcan}^S(g_1)$: then, $\mathit{MapCan}^S(g_1) = g'_1$. Moreover, since t_1 and t_3 occur in g_1, we have $\tau_{g_1}(t_1) = t_1$ and $\tau_{g_1}(t_3) = t_3$, and in branch B1 we have also $\tau_{g_1}(t_2) = t_2$ (since t_2 is actually the 2-predecessor of t_3 in B1), thus $h^S(t_1)$ is the term corresponding to t_1 in p_1, in the sense that $h^S(t_1)$ is the second argument of p_1, t_1 is the second argument of $s(t', t_1)$, and p_1 is the representation on the canonical chase of $s(t', t_1)$. In the same way, $h^S(t_2)$ is the term corresponding to t_2 in p_2, and $h^S(t_3)$ is the term corresponding to t_3 in p_3.

Then, consider the fact g_2. As explained in Example 7, $LastRJST(g_2, \{g_1\})$ is the Skolem term t_2, so $Bfin^S(g_2)$ is the sub-branch of branch B3 of the finite chase starting from $u(t_1, t_2)$ and ending in g_2 (such a sub-branch contains three facts). Therefore, $Bcan^S(g_2)$ is the branch of the canonical chase that starts from $IP^S(t_2) = p_2$ and whose sequence of IDs coincides with the sequence of IDs of the sub-branch $Bfin^S(g_2)$ of the finite chase. Let g'_2 be the leaf of $Bcan^S(g_2)$: then, $MapCan^S(g_2) = g'_2$.

Now, consider the fact g_3. It can be verified that $LastRJST(g_3, \{g_1, g_2\})$ is the Skolem term t_3, so $Bfin^S(g_3)$ is the sub-branch of branch B2 of the finite chase starting from $v(t_1, t_2, t_3)$ and ending in f (such a branch contains two facts). Therefore, $Bcan^S(g_3)$ is the branch of the canonical chase that starts from $IP^S(t_3) = p_3$ and whose sequence of IDs coincides with the sequence of IDs of the sub-branch $Bfin^S(g_3)$ of the finite chase. Let g'_3 be the leaf of $Bcan^S(g_3)$: then, $MapCan^S(g_3) = g'_3$.

Finally, we have $IM^S = \{g'_1, g'_2, g'_3\}$. ∎

Let t be a term. We denote by $\delta(t)$ the term obtained from t by replacing each occurrence of a Skolem function of the form $\phi^j_{I,p}$ with the Skolem function $\phi_{I,p}$. In other words, in $\delta(t)$ we eliminate all superscripts from the Skolem functions occurring in t. Moreover, given a fact $f = r(t_1, \ldots, t_n)$, we denote by $\delta(f)$ the fact $r(\delta(t_1), \ldots, \delta(t_n))$. The function δ will be necessary in the proof of the next lemma in order to properly compare Skolem terms of the finite chase with Skolem terms of the canonical chase.

We are finally ready to show the following crucial property.

Lemma 12. *Let F be a set of facts such that $F \subseteq fchase(\mathcal{I}, \mathcal{D}, m)$ and $m \geq 2(|F| \times |JST(F)|) + 2$. There exists a set of facts $IM \subseteq chase(\mathcal{I}, \mathcal{D})$ such that there exists a homomorphism h from F to IM.*

Proof. We first state the following Property (*), whose proof follows immediately from the definition of the f-chase-rule in Definition 2 and of the ID-chase-rule in Definition 1.

Property (*): Let g, g' be two facts such that g belongs to a branch $B(f)$ of $fchase(\mathcal{I}, \mathcal{D}, m)$, $g' \in chase(\mathcal{I}, \mathcal{D})$, and $\delta(g) \stackrel{k}{=} g'$ for some k such that $1 \leq k \leq m$. Let S be the sequence of IDs from $\mathcal{I}$ that are applied in $B(f)$ from g to obtain f, and let f' be the fact obtained starting from g' and applying the chase-rule of the canonical chase using the sequence of IDs S (let us denote by $B(f')$ this branch of $\mathit{chase}(\mathcal{I}, \mathcal{D})$). Then, $\delta(f) \stackrel{k}{=} f'$. Moreover, if t is a Skolem term introduced in $B(f)$ in a fact g_1 that follows (or is the same as) g in $B(f)$, and t' is the corresponding Skolem term introduced in $B(f')$, then $\delta(t) \stackrel{k+1}{=} t'$.

Let $f_1, \ldots, f_n$ be any enumeration of F, i.e., $F = \{f_1, \ldots, f_n\}$, and let $S = \langle f_1, \ldots, f_n \rangle$. With Property (*) in place, we now show that the following properties hold for every $f_i \in F$:

(1) for every $t \in \mathit{RJST}(\{f_1, \ldots, f_i\})$, $\delta(t) \stackrel{\ell}{=} h^S(t)$, where $\ell = |F| \times |\mathit{JST}(F)| + 2$;

(2) $\delta(f_i) \stackrel{\ell-1}{=} \mathit{MapCan}^S(f_i)$;

(3) for every $t \in \mathit{RJST}(f_i)$, if $\tau_{f_i}(t)$ occurs as the argument at position p in the last introduction point of $\tau_{f_i}(t')$ in $B(f_i)$, where $t' = \mathit{LastRJST}(f_i, \{f_1, \ldots, f_n\})$, then $h^S(t)$ occurs as the argument at position p in $\mathit{IP}^S(t')$.

We first prove property (1), by induction on the structure of sequence S.

Base case (fact f_1). If $\mathit{RJST}(f_1) = \emptyset$, then property (1) holds trivially. If $\mathit{RJST}(f_1) \neq \emptyset$, then suppose $t \in \mathit{RJST}(f_1)$. From Lemma 7 and from the fact that $\tau_{j+1} = \tau_j$ for every $j \geq |F| \times |\mathit{JST}(F)|$, it follows that $\tau_{f_1}(t) \stackrel{\ell}{=} t$ (since by hypothesis $m \geq |F| \times |\mathit{JST}(F)| + \ell$). Moreover, we have that $h^S(t)$ is the Skolem term occurring in $\mathit{Bcan}^S(f_1)$ that corresponds to $\tau_{f_1}(t)$ in $B(f_1)$, hence by construction we have $\delta(\tau_{f_1}(t)) \stackrel{m}{=} h^S(t)$; this, together with $\tau_{f_1}(t) \stackrel{\ell}{=} t$ immediately implies $\delta(t) \stackrel{\ell}{=} h^S(t)$.

Inductive case (fact f_i with $2 \leq i \leq n$). Let $t \in \mathit{RJST}(\{f_1, \ldots, f_i\})$. If $t \in \mathit{RJST}(\{f_1, \ldots, f_{i-1}\})$ then $\delta(t) \stackrel{\ell}{=} h^S(t)$ follows from the inductive hypothesis. If otherwise $t \in \mathit{RJST}(f_i) - \mathit{RJST}(\{f_1, \ldots, f_{i-1}\})$, there are two possible cases:

(a) $\mathit{RJST}(f_i) \cap \mathit{RJST}(\{f_1, \ldots, f_{i-1}\}) = \emptyset$. In this case, the branch $\mathit{Bcan}^S(f_i)$ starts from the root of $B(f_i)$ and the proof of property (1) is the same as in the base case;

(b) $\mathit{RJST}(f_i) \cap \mathit{RJST}(\{f_1, \ldots, f_{i-1}\}) \neq \emptyset$. In this case, let

$$t' = \mathit{LastRJST}(f_i, \{f_1, \ldots, f_{i-1}\})$$

By the inductive hypothesis, $\delta(t') \stackrel{\ell}{=} h^S(t')$. Now there are three possibilities:

(b1) $t = t'$. This case is impossible since by hypothesis $t \in RJST(f_i) - RJST(\{f_1, \ldots, f_{i-1}\})$ and $t' = LastRJST(f_i, \{f_1, \ldots, f_{i-1}\})$;

(b2) the last introduction point of $\tau_{f_i}(t)$ is subsequent to (or is the same as) the last introduction point of $\tau_{f_i}(t')$ in $B(f_i)$. In this case, from the fact that $\delta(t') \stackrel{\ell}{=} h^S(t')$ and from the second part of Property (*) it follows that $\delta(t) \stackrel{\ell}{=} h^S(t)$;

(b3) the last introduction point of $\tau_{f_i}(t)$ precedes the last introduction point of $\tau_{f_i}(t')$ in $B(f_i)$. In this case, let f' be a fact from $\{f_1, \ldots, f_{i-1}\}$ such that $t' \in RJST(f')$. From Lemma 9 we have that one of the following cases holds: (1) $\tau_{f_i}(t)$ is a h-ancestor of $\tau_{f_i}(t')$ in $B(f_i)$ for some h such that $h \leq \ell - 1$; (2) there exist t_0, t'_0 such that $\tau_{f_i}(t_0) = t'_0$ and $\tau_{f_i}(t)$ and t'_0 are siblings in $B(f_i)$ and t'_0 is a h-ancestor of $\tau_{f_i}(t')$ in $B(f_i)$ for some h such that $h \leq \ell - 1$; (notice that cases 2 and 4 in the claim of Lemma 9 cannot occur because the last introduction point of $\tau_{f_i}(t)$ precedes the last introduction point of $\tau_{f_i}(t')$ in $B(f_i)$). It is immediate to see that, in both cases, Definition 5 implies that $t \in RJST(f')$, which contradicts the hypothesis that $t \in RJST(f_i) - RJST(\{f_1, \ldots, f_{i-1}\})$. Thus, this case cannot occur.

Then, we prove property (2), again by induction on the structure of S. In the base case, property (2) follows immediately from the first part of Property (*) and from the fact that $B(f_1)$ and $Bcan^S(f_1)$ have the same root. As for the inductive case, if $LastRJST(f_i, \{f_1, \ldots, f_{i-1}\}) = \bot$, then since the root of $Bcan^S(f_i)$ is the same as the root of $B(f_i)$, from the first part of Property (*) it follows that $\delta(f_i) \stackrel{m}{=} MapCan^S(f_i)$. If $LastRJST(f_i, \{f_1, \ldots, f_{i-1}\}) = t'$, then let g be the last introduction point of $\tau_{f_i}(t')$ in $B(f_i)$: by property (1) we have $\delta(t') \stackrel{\ell}{=} h^S(t')$, hence by Lemma 5 it follows that $\delta(g) \stackrel{\ell-1}{=} IP^S(t')$, therefore from the first part of Property (*) the thesis follows.

We now prove property (3), again by induction on the structure of S. The base case is straightforward. As for the inductive case, assume that $\tau_{f_i}(t)$ occurs as the argument at position p in the last introduction point of $\tau_{f_i}(t')$ in $B(f_i)$. Let f_j be the first fact of S such that $t' \in RJST(f_j)$ (i.e., there is no $j' < j$ such that $t' \in RJST(f_{j'})$). Of course, $1 \leq j < i$. Now, since $\tau_{f_i}(t)$ and $\tau_{f_i}(t')$ occur in the same fact of $B(f_i)$ (which is an introduction point of $\tau_{f_i}(t')$ in $B(f_i)$), by Lemma 6 there are two possible cases:

i. $\tau_{f_i}(t)$ and $\tau_{f_i}(t')$ are p-q siblings in $B(f_i)$ for some q. Then, from Lemma 11 it follows that $\tau_{f_j}(t)$ is defined (i.e., $t \in RJST(f_j)$) and that $\tau_{f_j}(t)$ and $\tau_{f_j}(t')$ are p-q-siblings in $B(f_j)$. This in turn immediately implies that f_j is also the first fact of S such that $t \in RJST(f_j)$. Therefore, by Definition 7, we have that $h^S(t)$ occurs as the argument at position p in $IP^S(t')$, which proves the thesis;

ii. $\tau_{f_i}(t)$ is the p-predecessor of $\tau_{f_i}(t')$ in $B(f_i)$. In this case, from Lemma 11 it follows that $\tau_{f_j}(t)$ is defined (i.e., $t \in RJST(f_j)$) and that $\tau_{f_j}(t)$ is the p-predecessor of $\tau_{f_j}(t')$ in $B(f_j)$. Now, there are two possibilities: (a) f_j is also the first fact of S such that $t \in RJST(f_j)$. In this case, the thesis follows by an argument analogous to the proof of the above point (i); (b) there exists j' such that $j' < j$ and $t \in RJST(f_{j'})$. In this case, w.l.o.g. assume that j' is the minimum integer such that $t \in RJST(f'_j)$, let $t'' = \mathit{LastRJST}(f_j, F'_j)$, where $F'_j = \{f_1, \ldots, f_{j'-1}\}$, let g be the last introduction point of $\tau_{f_j}(t)$ in $B(f_j)$, let g' be the last introduction point of $\tau_{f_j}(t')$ in $B(f_j)$, and let g'' be the last introduction point of $\tau_{f_j}(t'')$ in $B(f_j)$. Now, we have that g'' must follow g in $B(f_j)$ (otherwise the hypothesis $t'' = \mathit{LastRJST}(f_j, F'_j)$ would be contradicted), and that g'' must precede g' in $B(f_j)$ (otherwise by Lemma 11 t' would belong to $RJST(f_{j'})$, thus contradicting the hypothesis that f_j is the first fact of S such that $t' \in RJST(f_j)$). Now, since $\tau_{f_j}(t)$ is the p-predecessor of $\tau_{f_j}(t')$ in $B(f_j)$, it follows that $\tau_{f_j}(t)$ occurs in all facts that lie between g and g' in $B(f_j)$, therefore $\tau_{f_j}(t)$ must occur at some position p' in g''. Consequently, by the inductive hypothesis, $h^S(t)$ occurs at position p' in $IP^S(t'')$. Then, by definition of $\mathit{Bfin}^S(f_j)$ and $\mathit{Bcan}^S(f_j)$ (and MapCan^S and IP^S), and since $\tau_{f_j}(t)$ is the p-predecessor of $\tau_{f_j}(t')$ in $B(f_j)$, we conclude that $h^S(t)$ occurs at position p in $IP^S(t')$.

Finally, let us define the function h which extends the above function h^S as follows: (i) for every constant c, $h(c) = c$; (ii) for every Skolem term of depth less than m, $h(t) = \delta(t)$. We now prove that, if t occurs in the i-th position of $f \in F$ and $h(t)$ has been defined (i.e., t is a constant or a Skolem term of depth less than m or a term of $JST(F)$), then $h(t)$ is the i-th argument of $\mathit{MapCan}^S(f)$. The case when t is a constant is immediately implied by the above property (2). In the case when t is a Skolem term of depth less than m, the property follows from the fact that t has a unique introduction point in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$. In the case when $t \in JST(F)$, the property immediately follows from the above property (3).

Thus, the function h defined so far maps all constants and Skolem terms occurring in F, with the exception of the Skolem terms of depth m which occur only once in F: it is now trivial to extend h to these terms: if such a term t occurs as the i-th argument of $f \in F$, then $h(t)$ is defined as the i-th argument of $MapCan^S(f)$. The above properties (2) and (3) immediately imply that the function h thus defined constitutes a homomorphism from F to IM^S. □

Based on the above lemma, we are finally to prove soundness of the evaluation of conjunctive queries over the finite chase with respect to the canonical chase.

Lemma 13. *Let $\mathcal{I}$ be a set of IDs, let $\mathcal{D}$ be a database instance, and let q be a conjunctive query with k occurrences of existential variable symbols. Let n be the size of q and let m be any integer such that $m \geq 2(n \times k) + 2$. Then, for every tuple of constants $\mathbf{c}$, if $\mathbf{c} \in q^{fchase(\mathcal{I},\mathcal{D},m)}$ then $\mathbf{c} \in q^{chase(\mathcal{I},\mathcal{D})}$.*

Proof. Assume $\mathbf{c} \in q^{fchase(\mathcal{I},\mathcal{D},m)}$: thus, there exists a set of facts F of $fchase(\mathcal{I}, \mathcal{D}, m)$ such that there exists a query homomorphism from $q(\mathbf{c})$ to F. Then, the proof follows immediately from the observation that F is a set of facts that satisfies the conditions of Lemma 12, since at most k occurrences of Skolem terms may appear in F, which implies that $|JST(F)| \leq k$, therefore by such lemma it follows that there exists a set of facts IM of $chase(\mathcal{I}, \mathcal{D})$ such that there exists a homomorphism h from F to IM. Thus, by composing the query homomorphism from $q(\mathbf{c})$ to F with h we obtain a query homomorphism from $q(\mathbf{c})$ to IM, which implies that $\mathbf{c} \in q^{chase(\mathcal{I},\mathcal{D})}$.
□

Now, since for every conjunctive query q there exists a value of m that satisfies the hypothesis of Lemma 13, it follows that for every conjunctive query q there exists a finite database in $sem_f(\mathcal{I}, D)$ (i.e., $fchase(\mathcal{I}, \mathcal{D}, m)$ for a suitable m) such that the evaluation of q over such a finite database coincides with the evaluation of q over the canonical chase $chase(\mathcal{I}, \mathcal{D})$. From the above property and Lemma 2 we are finally able to conclude that Theorem 1 holds.

4. Results for (unions of) conjunctive queries

Based on Theorem 1, in this section we complete our analysis of OWA-answering of CQs and UCQs.

First, we extend Theorem 1 to unions of conjunctive queries.

Theorem 2. *OWA-answering UCQs under IDs is finitely controllable.*

Proof. Let Q be the UCQ $q_1 \vee \ldots \vee q_k$, let n be the maximum size of a CQ in Q, and let h be the maximum number of occurrences of existential variables in a conjunct q_i of Q. Let $m = 2(n \times h) + 2$, and let $\mathbf{c}$ be a tuple of constants. We prove that $\mathbf{c} \in \mathit{ans}_f(Q, \mathcal{I}, \mathcal{D})$ iff $\mathbf{c} \in Q^{\mathit{fchase}(\mathcal{I},\mathcal{D},m)}$. First, suppose that $\mathbf{c} \notin Q^{\mathit{fchase}(\mathcal{I},\mathcal{D},m)}$. Then, since by Lemma 1 $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m) \in \mathit{sem}_f(\mathcal{I}, \mathcal{D})$, it immediately follows that $\mathbf{c} \notin \mathit{ans}_f(\mathcal{I}, \mathcal{D})$. Now suppose that $\mathbf{c} \in Q^{\mathit{fchase}(\mathcal{I},\mathcal{D},m)}$. Then, there exists q_i (with $1 \leq i \leq k$) such that $\mathbf{c} \in q_i^{\mathit{fchase}(\mathcal{I},\mathcal{D},m)}$. Therefore, from Lemma 13 it follows that $\mathbf{c} \in q_i^{\mathit{chase}(\mathcal{I},\mathcal{D})}$, and from Proposition 2 it follows that $\mathbf{c} \in \mathit{ans}(q_i, \mathcal{I}, \mathcal{D})$, which immediately implies $\mathbf{c} \in \mathit{ans}(Q, \mathcal{I}, \mathcal{D})$, and since $\mathit{sem}_f(\mathcal{I}, \mathcal{D}) \subseteq \mathit{sem}(\mathcal{I}, \mathcal{D})$, it follows that $\mathbf{c} \in \mathit{ans}_f(Q, \mathcal{I}, \mathcal{D})$. □

It is then possible to prove the analogous of Theorem 2 for the case of KDs and FKs.

Theorem 3. *OWA-answering UCQs under single KDs and FKs is finitely controllable.*

Proof. Let $\mathcal{I}$ be a set of IDs, let $\mathcal{K}$ be a set of single KDs such that $\mathcal{I}$ are foreign keys for $\mathcal{K}$, let $\mathcal{D}$ be a database instance, and let q be a UCQ. The proof follows immediately from the fact that, if $\mathcal{D}$ satisfies $\mathcal{K}$, then, by construction, the database $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ also satisfies the KDs in $\mathcal{K}$: indeed, it is immediate to verify that, since every ID in $\mathcal{I}$ is a foreign key for $\mathcal{K}$, every fact f that is added by the f-chase-rule in the construction of $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$ is such that there is no other fact with the same key as f in $\mathit{fchase}(\mathcal{I}, \mathcal{D}, m)$. □

Then, we prove that, as soon as we extend the ICs beyond single KDs and FKs, finite controllability of OWA-answering of CQs does not hold anymore.

Theorem 4. *OWA-answering CQs under non-conflicting KDs and IDs is not finitely controllable.*

Proof. Let $\mathcal{I}$ be the set of non-conflicting KDs and IDs constituted by the ID $r[2] \subseteq r[1]$ and the KD $\mathit{key}(r) = 2$. It is immediate to verify that $\mathcal{I}$ implies the ID $I = r[1] \subseteq r[2]$ over finite databases, while $\mathcal{I}$ does not imply I over unrestricted databases. Consequently, given an instance $\mathcal{D} = \{r(a,b)\}$, the query $\exists x.r(x,a)$ is true over finite databases while it is false over unrestricted databases. □

Then, we recall a result presented in [16] for OWA-answering CQs under non-conflicting KDs and IDs over unrestricted databases.

Proposition 3. *[16, Theorem 3.9] OWA-answering CQs under non-conflicting KDs and IDs is decidable, in particular it is in PTIME in data complexity and in PSPACE in combined complexity.*

Finally, we prove that the above property cannot be extended to the case of finite databases.

Theorem 5. *Finite OWA-answering CQs under non-conflicting KDs and IDs is undecidable.*

Proof. We prove the theorem by reducing implication of IDs from FDs and IDs (which is not finitely controllable [17], and is undecidable both for finite databases and for unrestricted databases [18, 19]) to OWA-answering of CQs under non-conflicting KDs and IDs. Given a set of FDs $\mathcal{F}$ which contains m FDs, a set of IDs $\mathcal{I}$, and an ID I, we define a set of KDs $\mathcal{K}'$ and a set of IDs $\mathcal{I}'$ as follows: we start from $\mathcal{K} = \emptyset$ and $\mathcal{I}' = \mathcal{I}$. Then, for each FD in $\mathcal{F}$: if the i-th FD in $\mathcal{F}$ is of the form $r : i_1, \ldots, i_k \rightarrow b$ (such a FD is denoted in the following by F_i), we use an auxiliary relation r_i (i.e., a new relation symbol that does not already occur in $\mathcal{F} \cup \mathcal{I}' \cup \{I\}$) of arity $2k+1$, add to $\mathcal{K}'$ the KD $key(r_i) = k+1, \ldots, 2k$, and add to $\mathcal{I}'$ the IDs

$$r_i[k+1, \ldots, 2k] \subseteq r_i[1, \ldots, k]$$
$$r[i_1, \ldots, i_k, b] \subseteq r[1, \ldots, k, 2k+1]$$

Finally, if the ID I has the form $I = r[l_1, \ldots, l_h] \subseteq s[j_1, \ldots, j_h]$ (where r has arity n and s has arity p), we define $\mathcal{D}(I)$ as the database $\mathcal{D} = \{r(\mathbf{c})\}$ with $\mathbf{c} = \langle c_1, \ldots, c_n \rangle$, and define $q(I)$ as the Boolean CQ $\exists x_1, \ldots, x_p.s(v_1, \ldots, v_p)$ where each v_i is such that $v_i = c_{l_k}$ if $i = j_k$ for some k s.t. $1 \leq k \leq h$, while $v_i = x_i$ otherwise. Notice that the set $\mathcal{K}' \cup \mathcal{I}'$ thus constructed is a set of non-conflicting KDs and IDs.

We now prove that $\mathcal{F} \cup \mathcal{I} \models_f I$ iff the CQ $q(I)$ is true in all databases of $sem_f(\mathcal{K}' \cup \mathcal{I}', \mathcal{D}(I))$. The proof is based on the fact that, for each of the above auxiliary relations r_i, the KD $K = key(r_i) = 1, \ldots, k$ is finitely implied by $\mathcal{K}' \cup \mathcal{I}'$, i.e., $\mathcal{K}' \cup \mathcal{I}' \models_f K$. This in turn implies that, due to the presence of the ID $r[i_1, \ldots, i_k, b] \subseteq r[1, \ldots, k, 2k+1]$, the KD K is "pulled back" to r, thus the original FD F_i is also implied, i.e., $\mathcal{K}' \cup \mathcal{I}' \models_f F_i$. Hence, all the initial FDs and IDs are finitely implied by $\mathcal{K}' \cup \mathcal{I}'$. Moreover, it is possible to prove that, for each IC φ where φ is either a FD or an ID over the initial relations (i.e., the relations occurring in $\mathcal{F} \cup \mathcal{I}$), if $\mathcal{K}' \cup \mathcal{I}' \models_f \varphi$ then $\mathcal{F} \cup \mathcal{I} \models_f \varphi$. Consequently: (i) if $\mathcal{F} \cup \mathcal{I} \models_f I$, then for each database $\mathcal{B}$ in $sem_f(\mathcal{K}' \cup \mathcal{I}', \mathcal{D}(I))$, there is a fact $s(\mathbf{t}')$ such that $\mathbf{t}'[B] = \mathbf{t}[A]$, which implies that the query $q(I)$ is true in $\mathcal{B}$; (ii) if $\mathcal{F} \cup \mathcal{I} \not\models_f I$, then there exists a database $\mathcal{B}$ in $sem_f(\mathcal{K}' \cup \mathcal{I}', \mathcal{D}(I))$ such that there is no fact $s(\mathbf{t}')$ such that $\mathbf{t}'[B] = \mathbf{t}[A]$, which implies that the query $q(I)$ is false in $\mathcal{B}$. □

Observe that the above results identify the first combination of ICs and query language (CQs under non-conflicting KDs and IDs) in which OWA-answering is decidable for unrestricted databases and is undecidable over finite databases.

Finally, we recall a result of [16], which answers the question whether OWA-answering of CQs in the presence of *conflicting* (i.e., arbitrary) single KDs and IDs is still decidable.

Proposition 4. *[16, Theorem 3.4] OWA-answering CQs under single KDs and IDs is undecidable.*

5. From OWA-answering to query containment

In this section we introduce query containment under ICs and briefly relate the results for OWA-answering presented above to query containment.

Given two queries q_1 and q_2 and a set of ICs $\mathcal{C}$, we say that q_1 *is contained in* q_2 *under* $\mathcal{C}$ (denoted by $q_1 \subseteq^{\mathcal{C}} q_2$) if, for each database $\mathcal{B} \in sem(\mathcal{C}, \emptyset)$, $q_1^{\mathcal{B}} \subseteq q_2^{\mathcal{B}}$. Analogously, we say that q_1 *is finitely contained in* q_2 *under* $\mathcal{C}$ (denoted by $q_1 \subseteq_f^{\mathcal{C}} q_2$) if, for each database $\mathcal{B} \in sem_f(\mathcal{C}, \emptyset)$, $q_1^{\mathcal{B}} \subseteq q_2^{\mathcal{B}}$.

When the query q_1 is a CQ, the relationship between OWA-answering and query containment can be informally explained as follows (for more details see e.g. [5]). In the absence of ICs, we "freeze" q_1 by replacing each distinct variable with a distinct constant in q_1 through a substitution σ, thus obtaining a set of facts, i.e., a database $\mathcal{D}(q_1)$. Then, it can be shown that $q_1 \subseteq^{\mathcal{C}} q_2$ iff $\mathbf{c} \in ans(q_2, \mathcal{C}, \mathcal{D}(q_1))$, where $\mathbf{c} = \sigma(\mathbf{x})$ and $\mathbf{x}$ is the tuple of all variables occurring in q_1. In the presence of a set of ICs $\mathcal{C}$, we must add a unification phase to the above procedure, since the ICs may imply equalities on the constants used for freezing the query q_1 (so the terms used for freezing q_2 are now "soft", i.e., unifiable, constants): if $\mathcal{C}$ is such that implication of ICs under $\mathcal{C}$ is decidable, then also this unification is computable in a finite amount of time.

As a consequence of the above reduction, all the decidability and finite controllability results for OWA-answering presented in this paper immediately extend to the corresponding query containment problems.

We only point out the two following results, which are corollaries of Theorem 1 and Theorem 3 respectively.

Given a class of queries $\mathcal{Q}$ and a class of ICs $\mathcal{C}$, we say that containment between queries in $\mathcal{Q}$ under $\mathcal{C}$ is finitely controllable if, for every set of ICs $\mathcal{I} \subseteq \mathcal{C}$, and for every pair of queries $q_1, q_2 \in \mathcal{Q}$, $q_1 \subseteq^{\mathcal{I}} q_2$ iff $q_1 \subseteq^{\mathcal{I}}_f q_2$.

Corollary 1. *Containment between CQs under arbitrary IDs is finitely controllable.*

Corollary 2. *Containment between CQs under single KDs and FKs is finitely controllable.*

The above two properties close two problems left open in [5], which established finite controllability of containment between CQs under *unary* IDs (i.e., IDs with arity 1) and under the so-called *key-based* dependencies, which constitute a combination of KDs and IDs much more restricted than single KDs and FKs, and left open the problem of finite controllability under arbitrary IDs and under more expressive combinations of KDs and IDs.

6. Related work

In this section we briefly describe some of the studies that are most closely related to the present work.

Query answering and query containment under ICs. With respect to query containment, the most closely related work is certainly [5], which shows decidability of containment of CQs under IDs (which immediately implies decidability of OWA-answering of CQs under IDs) and under the class of key-based dependencies (that has already been introduced in Section 5). These results have been extended in [16] to containment (and OWA-answering) of CQs under non-conflicting KDs and IDs for unrestricted databases.

The work in [20, 21] present results on undecidability of first-order query answering using unary conjunctive views. This setting is quite different from the one studied in the present paper, which actually cannot be reduced to the framework of unary conjunctive views (and vice versa). However, although in different settings, some of the results (in particular with respect to the use of negation and inequality) are similar.

View-based query processing is also closely related to OWA query answering. We only mention the approach presented in [22, 23], which studies query answering using views. In particular, [23] analyzes the presence of ICs, in particular functional dependencies, in this setting.

Many decidability results have been established for classes of ICs which admite a finite *chase*, i.e., a finite "canonical model" for the database and the ICs (see [24, 14]). For instance, [25] studies containment of conjunctive queries under (a generalized form of) acyclic IDs and FDs (whose chase is finite). Moreover, the approach presented in [26] studies containment of conjunctive queries under Datalog ICs, i.e., ICs that can be expressed in terms of a Disjunctive Datalog program. Again, Disjunctive Datalog programs cannot express arbitrary IDs, so the kinds of ICs analyzed in the present paper are not covered by the results in [26]. A similar setting is studied in [27, 28] (although under a least-fixpoint-based semantics that differs from the one presented in this paper), which also present results about conjunctive queries with inequality predicates which extend the one in [29]. Also, [30, 7] present results about query answering in a combination of dependencies for which the chase is finite, although in the different setting of *data exchange*. In particular, conjunctive queries and conjunctive queries with inequalities are studied. We point out that our results on finite controllability have very important implications for data exchange: for instance, Theorem 1 immediately implies that it is possible to compute certain answers in data exchange settings where target constraints are expressed by arbitrary (non-weakly-acyclic) inclusion dependencies. This is an interesting result, since it contrasts with the fact that for this class of constraints no finite *universal solution* exists for conjunctive queries.

Differently from the above mentioned work, in the present paper we have studied classes of ICs for which the chase is in general infinite, since we admit IDs with arbitrary cycles. This is the main technical difficulty of our work, and one of the main differences with respect to the above mentioned studies.

Finally, a recent interesting work that deals with the infinite chase is [31], which presents new general decidability and complexity results for OWA-answering under tuple-generating dependencies and equality-generating dependencies, for unrestricted databases. Also, [32] presents new results on the decidability of termination of the standard chase, and new sufficient conditions for the termination of the chase.

Implication of ICs. Many studies have dealt with the implication problem for FDs and IDs. Besides the "classical" results already cited in the previous sections, below we briefly describe some works which have a close relation to the present paper.

In [33] the authors identify one of the first combinations of ICs (namely, unary FDs and unary IDs) for which implication is not finitely controllable, although decidable both for the finite and for the unrestricted case. In this respect, our results about CQs under non-conflicting KDs and IDs (Theorem 5) identify the first (to our knowledge) class of FDs and IDs under which finite model reasoning is undecidable while unrestricted model reasoning is decidable.

The work presented in [34] defines a notion of non-conflicting FDs and IDs and proves decidability of implication from such ICs. Our notion of non-conflicting KDs and IDs is significantly different, because we take into account cyclic IDs, which cause the chase to be infinite, while in [34] only proper-circular IDs are considered (i.e., a class of IDs that has a finite chase).

Finally, [35, 36] have studied integrity constraints for XML. To this aim, they have shown that the implication problem for KDs and FKs is undecidable, which apparently contradicts our decidability results for KDs and FKs. However, we point out that the notion of foreign key in [35, 36] is different from ours: actually, since in [35, 36] a FK may involve a superset of a key, it follows that a set of keys and foreign keys according to [35, 36] is a set of conflicting KDs and IDs according to our classification, and hence OWA-answering under such ICs is undecidable, which agrees with the results in [35, 36].

7. Conclusions

In this paper we have studied query answering in databases with integrity constraints under open-world assumption. The main results of this paper concern the finite controllability of answering unions of conjunctive queries in the presence of either (i) arbitrary inclusion dependencies, or (ii) keys and foreign keys. We have also shown a class of integrity constraints (non-conflicting inclusion and key dependencies) in which answering unions of conjunctive queries is not finitely controllable, decidable over unrestricted databases, and undecidable over finite databases.

As for further development of the present work, we believe that one of the most promising aspects to investigate is the extension of the analysis presented in this paper towards different kinds of IC/schema languages (data models, ontology languages, etc.) and query languages.[4]. In particular, we conjecture that our results for the relational setting may imply interesting results for other schema languages (and data models) that have the ability of expressing analogous forms of key dependencies and inclusion dependencies: e.g., these results might drive the design of decidable query languages for such settings. Also, it would be interesting to analyze whether the finite chase, defined for proving the above mentioned finite controllability results, may imply interesting consequences for the design of practical query answering algorithms in the settings considered by this paper.

Acknowledgments

This research has been partially supported by the project TONES (FP6-7603) funded by the EU, by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by MIUR FIRB 2005 project "Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet" (TOCAI.IT). The author is extremely grateful to an anonymous reviewer of this paper for her/his invaluable comments and suggestions. The author also wishes to thank Georg Gottlob and Maurizio Lenzerini for useful discussions on the subject of the paper.

References

[1] M. Y. Vardi, The implication and finite implication problems for typed template dependencies, in: Proceedings of the First ACM SIGACT SIGMOD Symposium on Principles of Database Systems (PODS'82), 1982, pp. 230–238.

[2] M. H. Graham, A. O. Meldelzon, M. Y. Vardi, Notions of dependency satisfaction, Journal of the ACM 33 (1) (1986) 105–129.

[4] A study on the decidability and complexity of OWA query answering for classes of queries more expressive than CQs and UCQs has been presented in [13, 37].

[3] R. van der Meyden, Logical approaches to incomplete information, in: J. Chomicki, G. Saake (Eds.), Logics for Databases and Information Systems, Kluwer Academic Publishers, 1998, pp. 307–356.

[4] C. Beeri, M. Y. Vardi, Formal systems for tuple and equality generating dependencies, SIAM Journal on Computing 13 (1) (1984) 76–98.

[5] D. S. Johnson, A. C. Klug, Testing containment of conjunctive queries under functional and inclusion dependencies, Journal of Computer and System Sciences 28 (1) (1984) 167–189.

[6] M. Lenzerini, Data integration: A theoretical perspective., in: Proceedings of the Twentyfirst ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2002), 2002, pp. 233–246.

[7] R. Fagin, P. G. Kolaitis, L. Popa, Data exchange: Getting to the core, in: Proceedings of the Twentysecond ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2003), 2003, pp. 90–101.

[8] A. Y. Halevy, Answering queries using views: A survey, Very Large Database Journal 10 (4) (2001) 270–294.

[9] U. Hustadt, B. Motik, U. Sattler, Data complexity of reasoning in very expressive description logics, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI 2005), 2005.

[10] R. Fagin, P. G. Kolaitis, L. Popa, W.-C. Tan, Composing schema mappings: Second-order dependencies to the rescue, in: Proceedings of the Twentythird ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2004), 2004.

[11] M. Arenas, L. E. Bertossi, J. Chomicki, Consistent query answers in inconsistent databases, in: Proceedings of the Eighteenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS'99), 1999, pp. 68–79.

[12] A. Halevy, Z. Ives, D. Suciu, I. Tatarinov, Schema mediation in peer data management systems, in: Proceedings of the Nineteenth IEEE International Conference on Data Engineering (ICDE 2003), 2003, pp. 505–516.

[13] R. Rosati, On the decidability and finite controllability of query processing in databases with incomplete information, in: Proc. of the Twentyfifth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2006), 2006, pp. 356–365.

[14] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison Wesley Publ. Co., 1995.

[15] M. Y. Vardi, On the integrity of databases with incomplete information, in: Proceedings of the First ACM SIGACT SIGMOD Symposium on Principles of Database Systems (PODS'82), 1982, pp. 252–266.

[16] A. Calì, D. Lembo, R. Rosati, On the decidability and complexity of query answering over inconsistent and incomplete databases, in: Proceedings of the Twentysecond ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2003), 2003, pp. 260–271.

[17] M. A. Casanova, R. Fagin, C. H. Papadimitriou, Inclusion dependencies and their interaction with functional dependencies, Journal of Computer and System Sciences 28 (1) (1984) 29–59.

[18] J. C. Mitchell, The implication problem for functional and inclusion dependencies, Information and Control 56 (1983) 154–173.

[19] A. K. Chandra, M. Y. Vardi, The implication problem for functional and inclusion dependencies is undecidable, SIAM Journal on Computing 14 (3) (1985) 671–677.

[20] J. Bailey, G. Dong, K. Ramamohanarao, Decidability and undecidability results for the termination problem of active database rules, in: Proceedings of the Seventeenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS'98), 1998, pp. 264–273.

[21] J. Bailey, G. Dong, Decidability of first-order logic queries over views, in: Proceedings of the Seventh International Conference on Database Theory (ICDT'99), 1999, pp. 83–99.

[22] S. Abiteboul, O. Duschka, Complexity of answering queries using materialized views, in: Proceedings of the Seventeenth ACM SIGACT

SIGMOD SIGART Symposium on Principles of Database Systems (PODS'98), 1998, pp. 254–265.

[23] O. M. Duschka, M. R. Genesereth, A. Y. Levy, Recursive query plans for data integration, Journal of Logic Programming 43 (1) (2000) 49–73.

[24] C. Beeri, M. Y. Vardi, A proof procedure for data dependencies, Journal of the ACM 31 (4) (1984) 718–741.

[25] X. Zhang, Z. M. Ozsoyoglu, Implication and referential constraints: A new formal reasoning, IEEE Transactions on Knowledge and Data Engineering 9 (6) (1997) 894–910.

[26] G. Dong, J. Su, Conjunctive query containment with respect to views and constraints, Information Processing Letters 57 (2) (1996) 95–102.

[27] R. van der Meyden, The complexity of querying indefinite information, Ph.D. thesis, Rutgers University (1992).

[28] R. van der Meyden, The complexity of querying indefinite data about linearly ordered domains, Journal of Computer and System Sciences 54 (1) (1997) 113–135.

[29] A. C. Klug, On conjunctive queries containing inequalities, Journal of the ACM 35 (1) (1988) 146–160.

[30] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa, Data exchange: Semantics and query answering, in: Proceedings of the Ninth International Conference on Database Theory (ICDT 2003), 2003, pp. 207–224.

[31] A. Calì, G. Gottlob, M. Kifer, Taming the infinite chase: Query answering under expressive relational constraints, in: Proc. of the 11th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2008), 2008.

[32] A. Deutsch, A. Nash, J. B. Remmel, : The chase revisited, in: Proc. of the Twentyseventh ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2008), 2008, pp. 149–158.

[33] S. S. Cosmadakis, P. C. Kanellakis, M. Vardi, Polynomial-time implication problems for unary inclusion dependencies, Journal of the ACM 37 (1) (1990) 15–46.

[34] M. Levene, G. Loizou, How to prevent interaction of functional and inclusion dependencies, Information Processing Letters 71 (4) (1999) 115–125.

[35] W. Fan, J. Siméon, Integrity constraints for XML, in: Proceedings of the Nineteenth ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2000), 2000, pp. 23–34.

[36] W. Fan, L. Libkin, On XML integrity constraints in the presence of DTDs, Journal of the ACM 49 (3) (2002) 368–406.

[37] R. Rosati, The limits of querying ontologies, in: Proceedings of the Eleventh International Conference on Database Theory (ICDT 2007), Vol. 4353 of Lecture Notes in Computer Science, Springer, 2007, pp. 164–178.