On the Relationship of Joint Acyclicity and Super-Weak Acyclicity

Technical Report 3037
Institute AIFB, Karlsruhe Institute of Technology, DE
10 January 2013

Markus Krötzsch[1] and Sebastian Rudolph[2]

[1] Department of Computer Science, University of Oxford, UK
markus.kroetzsch@cs.ox.ac.uk
[2] Institute AIFB, Karlsruhe Institute of Technology, DE
sebastian.rudolph@kit.edu

1 Introduction

Rules that allow existential quantifiers in the head (conclusion) have been studied extensively in knowledge representation, databases, and logic programming (in the related form of logic programs with function symbols). In general, it is not decidable if a set of such rules admits a finite model, but a number of sufficient conditions have been proposed to detect situations where this is the case.

Two such *acyclicity* conditions are *joint acycliclity* [8] and *super-weak acyclicity* [9]. This short technical note investigates the relationship between the two conditions. The main result is that, while super-weak acyclcity is a more general condition, the two conditions coincide on all sets of rules that do not contain multiple occurrences of the same variable in any body atom (Theorem 1). We also observe that this restriction does not reduce the expressivity of jointly acyclic rules (Theorem 2).

We first introduce our basic notation (Section 2) and recall the idea of acyclicity using the case of *weak acyclicity* (Section 3). Thereafter, we introduce joint acyclicity (Section 4) and conmpare it with super-weak acyclicity (Section 5).

2 Preliminaries

Our syntax is based on a standard first-order logic signature that is based on finite sets $\mathbf{C}$ of *constants*, $\mathbf{F}$ of *function symbols*, and $\mathbf{P}$ of *predicates*, and on an infinite set of *variables* $\mathbf{V}$. These sets are mutually disjoint. We will usually keep the signature implicit. The assumption on finiteness is useful when defining notions such as *grounding*. A function $\mathsf{ar} : \mathbf{F} \cup \mathbf{P} \to \mathbb{N}$ associates a natural number $\mathsf{ar}(\sigma)$ with each function symbol or predicate $\sigma \in \mathbf{F} \cup \mathbf{P}$ that defines the *arity* of σ. The set of *positions* of $\sigma \in \mathbf{F} \cup \mathbf{R}$ is the set $\Pi_\sigma = \{\langle \sigma, 1\rangle, \ldots, \langle \sigma, \mathsf{ar}(\sigma)\rangle\}$. A *term* is a standard first-order term constructed from variables, constants, and function symbols.

We generally use letters x, y, z, v, w, u for variables, a, b, c, d for constants, f, g for function symbols, R, S, T for predicates, and s, t for terms, possibly with sub-

or superscripts. Boldfaced expressions, such as $\mathbf{x}$ or $\mathbf{c}$, are used to denote lists of the respective elements.

Definition 1. *An* atom *is a formula of the form $R(t_1, \ldots, t_n)$ where* $\mathsf{ar}(R) = n$*. An* existential rule *(or simply* rule *in the context of this paper) is a formula of the form*

$$\forall \mathbf{x}. \forall \mathbf{y}. \varphi[\mathbf{x}, \mathbf{y}] \to \exists \mathbf{z}. \psi[\mathbf{x}, \mathbf{z}]$$

where $\mathbf{x}, \mathbf{y}$, and $\mathbf{z}$ are mutually disjoint lists of variables, and $\varphi[\mathbf{x}, \mathbf{y}]$ and $\psi[\mathbf{x}, \mathbf{z}]$ are conjunctions of function-free atoms that contain exactly *the variables in $\mathbf{x}, \mathbf{y}$ and $\mathbf{x}, \mathbf{z}$, respectively. In particular, all variables in $\mathbf{x}$ must occur in φ, a condition known as* safety.

The conjunction φ is called the premise *or* body*, ψ is called the* conclusion *or* head*, and $\mathbf{x}$ is called the* frontier*. A* Datalog rule *is a rule without existential quantifiers, i.e., one where $\mathbf{z}$ is empty. A* fact *is a rule with empty body (a conclusion that is unconditionally true). As opposed to the body, we require the head of a rule to contain at least one atom.*

Safety of rules simplifies the definition of some syntactic conditions in the paper. However, all of our results remain valid if one assumes the existence of a special atom of the form $\top(x)$ that describes a tautological condition, which can be used to make unsafe rules safe without changing their semantics. We usually omit the preceding universal quantifiers when writing rules, and we use sets of atoms as a convenient notation for conjunctions of atoms.

In the literature on databases, existential rules are known as *Tuple Generating Dependencies*, constants are known as *values*, and predicates are known as *relations*. Rules are often assumed to contain no constant symbols and are thus strictly separated from facts. A set of facts forms a *database* (*instance*). We do not use this terminology in this paper.

Definition 2. *A rule set $\mathcal{R}$ is* renamed apart *if each variable name is bound in at most one quantifier in $\mathcal{R}$. A* universal (existential) variable *in such a renamed rule set is one that occurs in a universal (existential) quantifier. A* frontier variable *is a universal variable that occurs in the head of its rule.*

The previous condition allows us to uniquely identify variables by their name, which simplifies many definitions.

Existential rules are a syntactic fragment of first-order predicate logic, and we consider it under the according semantics.

Definition 3. *An* interpretation *$\mathcal{I}$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$, defined as usual. The notion of a* model *of a set S of logical formulae is defined according to the standard semantics of first-order logic. Two sets S and T of logical formulae are* equivalent *if they have the same set of models. In particular, this terminology applies to sets of rules.*

This also means that every rule set is equivalent to one that is renamed apart. We do not make the *Unique Name Assumption* (UNA) where different constants are required to be interpreted differently. This becomes relevant only when considering rules that include equality, which we do not do herein.

The primary reasoning problem that we consider is the following kind of query containment.

Definition 4. *A* boolean conjunctive query *(BCQ) is a formula* $\exists \mathbf{v}.Q$ *where* Q *is a conjunction of atoms and* $\mathbf{v}$ *contains all variables in* Q*. A BCQ* $\exists \mathbf{v}.Q$ *is* entailed *by* $\mathcal{R}$ *if it is entailed under standard first-order logic semantics.*

This is closely related to the relevant task of finding *certain answers* to conjunctive queries with free variables (that is, for CQs that are not BCQs). Checking satisfiability and BCQ entailment for unrestricted existential rules is undecidable [4,3] even with very strong restrictions on the vocabulary or the number of rules [2].

3 Weak Acyclicity and the Skolem Chase

As a first introduction to the idea of acyclicity, we recapitulate the notion of *weak acyclicity* [6,7] and its relationship to a simple inferencing procedure known as the *Skolem chase* [9].

Definition 5. *Given a set of rules* $\mathcal{R}$*, the* dependency graph *is a directed graph that has the positions of predicates in* $\mathcal{R}$ *as its nodes. For every rule* $\rho \in \mathcal{R}$*, and every variable* x *at position* $\langle R, p\rangle$ *in the head of* ρ*, the graph contains edges as follows:*

- *If* x *is universally quantified, and* x *occurs in a body atom at position* $\langle S, q\rangle$*, there is an edge from* $\langle S, q\rangle$ *to* $\langle R, p\rangle$*.*
- *If* x *is existentially quantified, and the body of* ρ *contains a (necessarily universally quantified) variable* y *at* $\langle S, q\rangle$ *such that* y *also occurs in the head of* ρ*, then there is a* special edge *from* $\langle S, q\rangle$ *to* $\langle R, p\rangle$*.*

$\mathcal{R}$ *is* weakly acyclic *if its dependency graph has no cycle going through a special edge. The class of weakly acyclic rule sets is denoted* wa*.*

Intuitively, non-special edges encode the possible passing of values in bottom-up reasoning, whereas special edges encode the dependency between the premise that a rule was applied to and the new individuals that the application of this rule entails. A cycle over special edges may indicate that newly invented values can recursively be used in premises which require the invention of further values *ad infinitum*. For instance, the dependency graph of the rule

$$R(x, y) \rightarrow \exists z.R(y, z) \tag{1}$$

has a special edge from $\langle R, 2\rangle$ to itself. Indeed this rule may lead to the construction of an infinite R-chain of new elements. To formalise this, it is convenient to think of a concrete reasoning procedure being applied to rules. To this end, it is useful to express existential quantifiers using Skolem function symbols:

Definition 6. *Consider a rule ρ of the form $\forall \mathbf{x}, \mathbf{y}.\varphi[\mathbf{x}, \mathbf{y}] \rightarrow \exists \mathbf{z}.\psi[\mathbf{x}, \mathbf{z}]$ where $\mathbf{x}$ are exactly those variables that occur in head and body. Let l be the size of $\mathbf{x}$. The* Skolemisation *of ρ is the rule $\forall \mathbf{x}, \mathbf{y}.\varphi[\mathbf{x}, \mathbf{y}] \rightarrow \psi'[\mathbf{x}]$ where ψ' is obtained from ψ by replacing every variable $z \in \mathbf{z}$ with a function term $f_z(\mathbf{x})$ where f_z is a fresh function symbol of arity l. The Skolemisation of a set of rules is the union of the Skolemisations of each of its elements.*

The set of body variables $\mathbf{x}$ of a rule ρ that also occur in the head of ρ is known as the *frontier* of ρ. In Definition 6 we only use frontier variables in Skolem terms since one can rewrite $\forall \mathbf{x}, \mathbf{y}.\varphi[\mathbf{x}, \mathbf{y}] \rightarrow \exists \mathbf{z}.\psi[\mathbf{x}, \mathbf{z}]$ as $\forall \mathbf{x}.(\exists \mathbf{y}.\varphi[\mathbf{x}, \mathbf{y}]) \rightarrow (\exists \mathbf{z}.\psi[\mathbf{x}, \mathbf{z}])$. This also explains why Definition 5 considers only frontier variables as possible sources of special edges.

Fact 1 *Given a rule set $\mathcal{R}$ with Skolemization $\mathcal{R}'$, and a BCQ $\exists \mathbf{v}.Q$ over the signature of $\mathcal{R}$ (without Skolem function symbols), we have $\mathcal{R} \models \exists \mathbf{v}.Q$ iff $\mathcal{R}' \models \exists \mathbf{v}.Q$.*

A possible approach to BCQ answering is to recursively construct consequences bottom-up, similar to the consequence operator in Logic Programming. In database theory where existential rules have often been studied, this procedure (and its possibly infinite result) is known as the *chase*.

Definition 7. *Consider a set of rules $\mathcal{R}$ and let $\mathcal{R}'$ be a Skolemisation of $\mathcal{R}$. A* ground instance *of a rule is a formula that is obtained by removing all quantifiers and by uniformly replacing each variable with a variable-free (i.e.* ground*) term (possibly including Skolem functions). The* Skolem chase *is defined to be the least set SkC of ground facts that is closed under the following rule:*

> *if the rule ρ is the ground instance of a rule in $\mathcal{R}'$ and if $B \in SkC$ for every body atom B of ρ, then $H \in SkC$ for every head atom H of ρ.*

The Skolem chase can be computed by applying rules bottom-up but this construction may not terminate. For example, using rule (2) together with a fact $R(a, b)$ the Skolem chase has the form $\{R(a, b), R(b, f(b)), R(b, f(f(b))), \ldots\}$. In essence, all acyclicity conditions studied in this paper can be viewed as syntactic checks whether the Skolem chase terminates for a given set of rules. Deciding this in the general case is not possible:

Fact 2 *The problem of determining whether the Skolem chase terminates for a given set of rules is undecidable.*

This is an easy consequence of standard Turing machine simulations in logic programs [5], where the halting of the Turing machine is equivalent to the finiteness of the facts derived in forward-chaining. Finite or not, the Skolem chase is a correct reasoning procedure, and we can strengthen Fact 1 as follows:

Fact 3 *Given a rule set $\mathcal{R}$ with (possibly infinite) Skolem chase SkC, and a BCQ $\exists \mathbf{v}.Q$ over the signature of $\mathcal{R}$ (without Skolem function symbols), we have $\mathcal{R} \models \exists \mathbf{v}.Q$ iff $SkC \models \exists \mathbf{v}.Q$.*

Fact 4 *The Skolem chase of a weakly-acyclic set of rules $\mathcal{R}$ is finite.*

The previous statement is easy to verify. Weak acyclicity ensures that the Skolem chase does not lead to any Skolem term of the form $f(\mathbf{t})$ where the arguments $\mathbf{t}$ contain another term with the Skolem function f. Since there are only a finite number of terms without this form of nesting, the overall number of derivable ground facts over the (fixed) predicates is bounded. This type of reasoning can also be used to obtain a good intuition about the advanced notions of acyclicity considered below.

Summing up, an acyclicity condition is a criterion that allows us to decide whether the Skolem chase will terminate on a given set of rules.

4 Joint Acyclicity

This section introduces *joint acyclicity*, which is a proper generalisation of the notion of weak acyclicity introduced in Section 3. Considering again the example (1), we note that the potential for creating an infinite Skolem chase is lost when extending the rule as follows:

$$R(x, y) \land C(y) \to \exists z.R(y, z). \tag{2}$$

This rule cannot be applied recursively since invented values are not required to belong to C. Yet, the dependency graph contains the same cycle as before and the rule is not weakly acyclic. This highlights an important deficiency of weak acyclicity that was also noted in [9]: weakening a rule by adding additional requirements to its body may destroy weak acyclicity. Moreover, note that even the occurrence of existentially quantified variables on all premise positions would not necessarily lead to an infinite Skolem chase:

$$A(x) \land B(x) \to \exists y, z.S(x, y, z) \land A(y) \land B(z). \tag{3}$$

Again, the example fails to be weakly acyclic even though no infinite Skolem chase can occur. We capture this formally by shifting our focus from positions to variables (which can occur in multiple positions):

Definition 8. *Consider a renamed apart set of rules $\mathcal{R}$. For a variable x, let $\mathsf{Pos}_B(x)$ ($\mathsf{Pos}_H(x)$) be the set of all positions where x occurs in the body (head) of a – necessarily unique – rule. Now for any existentially quantified variable v, let $\mathsf{Move}(v)$ be the smallest set of positions such that (1) $\mathsf{Pos}_H(v) \subseteq \mathsf{Move}(v)$, and (2) $\mathsf{Pos}_H(y) \subseteq \mathsf{Move}(v)$ for every universally quantified variable y with $\mathsf{Pos}_B(y) \subseteq \mathsf{Move}(v)$.*

The existential dependency graph *of $\mathcal{R}$ has the existentially quantified variables of $\mathcal{R}$ as its nodes. There is an edge from v to w if the rule where w occurs contains a universally quantified (body) variable z that also occurs in the head and with $\mathsf{Pos}_B(z) \subseteq \mathsf{Move}(v)$.*

$\mathcal{R}$ is jointly acyclic *if its existential dependency graph is acyclic. The class of jointly acyclic rule sets is denoted* ja.

Thus $\mathsf{Move}(x)$ contains the positions in which values invented for x may appear. This captures the effect of non-special edges in Definition 5, whereas special edges correspond to edges in the existential dependency graph. Definition 5 is obtained by modifying condition (2) in Definition 8 to require $\mathsf{Pos}_B(y) \cap \mathsf{Move}(x) \neq \emptyset$ instead of $\mathsf{Pos}_B(y) \subseteq \mathsf{Move}(x)$. This states that a value is propagated by a rule if it satisfies *some* – instead of *all* – of the rule's premises. Joint acyclicity therefore appears to be the more natural condition.

The following rule is jointly acyclic (as a singleton set) but not weakly acyclic: its existential dependency graph does not have any edges whereas its dependency graph is a clique of special edges.

$$R(x, y) \wedge S(x, y) \rightarrow \exists v, w.R(x,v) \wedge R(w, y) \wedge S(x, w) \wedge S(v, y) \tag{4}$$

In spite of this generalisation, joint acyclicity is easy to recognise.

Proposition 1. *Checking whether a set of rules is jointly-acyclic is* P*-complete w.r.t. the size of the rule set.*

Proof. Detecting cycles in a directed graph and checking inclusion of a position in $\mathsf{Move}(x)$ is clearly possible in polynomial time. The latter problem is also hard for P since propositional Horn logic entailment can be expressed using unary predicates with a single variable to encode propositions. □

5 Super-Weak Acyclicity

Another generalisation of weak acyclicity, called *super-weak acyclicity* (*swa*), has been proposed in [9]. Super-weak acyclicity is more general than joint acyclicity as it uses function symbols and unification to exclude some additional cases of value propagation. However, as we show in this section, joint acyclicity and super-weak acyclicity coincide on the major class of *duplicate-free* rule sets. Since every rule set can be expressed by one that is duplicate-free, we argue that *ja* already captures the main improvement that *swa* provides over weak acyclicity, so that the additional machinery needed to define *swa* might not be desirable.

To provide for a more accurate estimation of acyclicity, super-weak acyclicity considers not just the predicate positions where an invented value may occur, but also the syntactic form of the atoms where values occurred in the Skolemised rule set. A position in the context of a rule atom is called a *place*, formally given by a pair $\langle a, i\rangle$ where a is an atom with n arguments and $i \in \{1, \ldots, n\}$ is the index of a parameter in a. For example, consider the rule

$$R(x, x) \rightarrow \exists y.R(x, y) \wedge R(y, x). \tag{5}$$

This rule is not jointly acyclic, since $\langle R, 1\rangle, \langle R, 2\rangle \in \mathsf{Move}(y)$. Yet, the Skolem chase of the according Skolemised rule

$$R(x, x) \rightarrow R(x, f(x)) \wedge R(f(x), x). \tag{6}$$

is finite, since the rule is not applicable to any of the derived facts. Namely, $f(x)$ occurs in the places $\langle R(x, f(x))\rangle$ and $\langle x, f(x)\rangle$, and neither of these occurrences can be *unified* with $R(x', x')$ (a renamed apart variant of the premise of (6)). By using places and unification instead of positions and set containment, Definition 8 can be generalised to the definition of super-weak acyclicity:

Definition 9. *Consider a renamed apart set of rules $\mathcal{R}$ with Skolemisation $\mathcal{R}'$. For a term t, let $\mathsf{Plc}_B(t)$ ($\mathsf{Plc}_H(t)$) be the set of all places where t occurs in the body (head) of a – necessarily unique – rule of $\mathcal{R}'$. Given two sets P_1, P_2 of places of $\mathcal{R}'$, we write $P_1 \rightsquigarrow P_2$ if, for every $\langle a, i\rangle \in P_2$, there is some $\langle b, i\rangle \subseteq P_1$ and two substitutions θ, θ' such that $a\theta = b\theta'$.*

Now for any Skolem term $f(\mathbf{x})$ in $\mathcal{R}'$, let $\mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x}))$ be the smallest set of places such that (1) $\mathsf{Plc}_H(f(\mathbf{x})) \subseteq \mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x}))$, and (2) $\mathsf{Plc}_H(y) \subseteq \mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x}))$ for every universally quantified variable y with $\mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x})) \rightsquigarrow \mathsf{Plc}_B(y)$.

The unification dependency graph *of $\mathcal{R}$ has the Skolem terms of $\mathcal{R}'$ as its nodes. There is an edge from $f(\mathbf{x})$ to $g(\mathbf{y})$ if $\mathbf{y}$ contains a variable z with $\mathsf{Plc}_B(z) \rightsquigarrow \mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x}))$.*

$\mathcal{R}$ is super-weakly acyclic *if its unification dependency graph is acyclic. The class of weakly acyclic rule sets is denoted swa.*

We have slightly changed the formulation as compared to [9] to emphasize the similarity to Definition 8. It is easy to verify that the resulting notion is the same.

Though one could expect that the use of substitution provides a much better estimate of value propagation during the chase, this effect is very much limited since only individual atoms are considered. Indeed, joint acyclicity and super-weak acyclicity coincide in many cases. More precisely, we say that a rule set is *duplicate-free* if it contains no body atom in which a variable occurs more than once.

Theorem 1. *A duplicate-free rule set $\mathcal{R}$ is jointly acyclic if and only if it is super-weakly acyclic.*

Proof. Given a set P of places, define $P{\downarrow}$ to be the set of positions $\{\langle p, i\rangle \mid \langle p(\mathbf{t}, i)\rangle \in P\}$. Consider an existentially quantified variable v in $\mathcal{R}$ that was replaced by the Skolem term $f(\mathbf{x})$ in the Skolemisation $\mathcal{R}'$ of $\mathcal{R}$. We first show that $\mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x})){\downarrow} = \mathsf{Move}(v)$ by induction over their definitions in Definition 8 and 9. The base case (1) is obvious.

For the inductive step (2), consider a universally quantified variable y as in (2) of the respective defintions. We show that $\mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x})) \rightsquigarrow \mathsf{Plc}_B(y)$ iff $\mathsf{Move}(v) \subseteq \mathsf{Pos}_B(y)$. The "only if" direction is part of the definition of $\rightsquigarrow$. For the "if" direction, we observe that every place $\langle a, i\rangle \in \mathsf{Pos}_B(y)$ is of the form $R(\mathbf{z})$ where $\mathbf{z}$ is a vector of mutually distinct variables. Therefore, the substitutions required in the definition of $\rightsquigarrow$ do always exist as long as there is an atom $\langle b, i\rangle \in \mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x}))$ where b is of the form $R(\mathbf{t})$. By the induction hypothesis, this is the case exactly if $\langle R, i\rangle \in \mathsf{Move}(v)$, which shows the claim.

We thus find that the existential dependency graph and the unification dependency graph are equal up to renaming of existential variables into Skolem terms. Indeed, the edges of both graphs agree since $\mathsf{Plc}_B(z) \rightsquigarrow \mathsf{Move}_{\mathsf{swa}}(f(\mathbf{x}))$ iff $\mathsf{Pos}_B(z) \subseteq \mathsf{Move}(v)$ which follows by a similar argument as in the induction step above. □

Thus the difference between swa and ja is limited to rule sets that are not duplicate-free. The following result that is based on a construction by Afrati et al. [1] shows that such rule sets do not add real expressivity over duplicate-free rule sets:

Theorem 2. *For every rule set $\mathcal{R}$ that contains a body atom in which a variable occurs more than once, there is a rule set $\mathcal{R}'$ where this is not the case and such that a BCQ* *over $\mathcal{R}$ follows from $\mathcal{R}$ iff it follows from $\mathcal{R}'$. Moreover, $\mathcal{R}'$ can be constructed in time exponential in the maximal arity of predicates in $\mathcal{R}$ and polynomial in the size of $\mathcal{R}$ if the maximal arity is fixed.*

This statement has been shown in [1, Proposition 2.10] for the case of Datalog programs, where duplicate-free Datalog programs have been called *normal*. The construction for sets of existential rules is essentially the same. We do not recapitulate the formal construction of the proof from [1] and rather give an illustrating example. Consider the rules

$$R(x,x,y) \to C(y) \tag{7}$$
$$R(x,y,z) \to \exists v.R(y,z,v). \tag{8}$$

To eliminate the duplicate variable in rule (7), the atom $R(x,x,y)$ is replaced by an auxiliary atom $R_{113}(x,y)$ where R_{113} is a new predicate (the vector 113 encodes the sequence of parameter positions where each variable in $R(x,x,y)$ first occurs). Instead of adding a rule $R(x,x,y) \to R_{113}(x,y)$, we consider all rules that could possibly lead to the derivation of a fact matching $R(x,x,y)$, and we create special instances of these rules for deriving $R_{113}(x,y)$. This is only the case for rule (8) if y and z represent the same value. Thus we obtain a new rule set:

$$R_{113}(x,y) \to C(y) \tag{9}$$
$$R(x,y,z) \to \exists v.R(y,z,v) \tag{10}$$
$$R(x,y,y) \to \exists v.R_{113}(y,v). \tag{11}$$

where rule (11) has been introduced as a specialization of (8). This leads to a new atom $R(x,y,y)$ with duplicate variables that is again replaced by an auxiliary atom $R_{122}(x,y)$. However, none of the original rules (especially not rule (8)) can actually lead to $R(x,y,y)$ (or $R_{122}(x,y)$) being derived. So the final rule set is:

$$R_{113}(x,y) \to C(y) \tag{12}$$
$$R(x,y,z) \to \exists v.R(y,z,v) \tag{13}$$
$$R_{122}(x,y) \to \exists v.R_{113}(y,v). \tag{14}$$

If facts were given, they would be translated like rules. It is not difficult but a bit tedious to formulate the complete transformation in general. It is easy to see that only the original rule set needs to be considered when looking for ways to derive an auxiliary atom, and that the number of auxiliary rules obtained in each replacement step is thus bounded by the (linear) number of head atoms in the input rule set. Moreover, up to renaming of variables the number of possible duplicate variable patterns is exponentially bounded by the arity of the respective predicate. These observations lead to the claimed upper bounds in Theorem 2.

References

1. Afrati, F., Cosmadakis, S., Foustoucos, E.: Datalog programs and their persistency numbers. ACM Transactions on Computational Logic 6(3), 481–518 (2005)

2. Baget, J.F., Leclère, M., Mugnier, M.L., Salvat, E.: On rules with existential variables: Walking the decidability line. Artificial Intelligence 175(9–10), 1620–1654 (2011)
3. Beeri, C., Vardi, M.Y.: The implication problem for data dependencies. In: Even, S., Kariv, O. (eds.) Proc. 8th Colloquium on Automata, Languages and Programming (ICALP'81). LNCS, vol. 115, pp. 73–85. Springer (1981)
4. Chandra, A.K., Lewis, H.R., Makowsky, J.A.: Embedded implicational dependencies and their inference problem. In: Proc. 13th Annual ACM Symposium on Theory of Computation (STOC'81). pp. 342–354. ACM (1981)
5. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. ACM Computing Surveys 33(3), 374–425 (2001)
6. Deutsch, A., Tannen, V.: Reformulation of XML queries and constraints. In: Calvanese, D., Lenzerini, M., Motwani, R. (eds.) Proc. 9th Int. Conf. on Database Theory (ICDT'03). LNCS, vol. 2572, pp. 225–241. Springer (2003)
7. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data exchange: semantics and query answering. Theoretical Computer Science 336(1), 89–124 (2005)
8. Krötzsch, M., Rudolph, S.: Extending decidable existential rules by joining acyclicity and guardedness. In: Walsh, T. (ed.) Proc. 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI'11). pp. 963–968. AAAI Press/IJCAI (2011)
9. Marnette, B.: Generalized schema-mappings: from termination to tractability. In: Paredaens, J., Su, J. (eds.) Proc. 28th Symposium on Principles of Database Systems (PODS'09). pp. 13–22. ACM (2009)