Regular n-ary Queries in Trees and Variable Independence

Emmanuel Filiot[1] and Sophie Tison[1]

University of Lille 1, LIFL, UMR 8022 of CNRS, INRIA Lille - Nord Europe
Emmanuel.Filiot@lifl.fr Sophie.Tison@lifl.fr

Summary. Regular n-ary queries in trees are queries which are definable by an MSO formula with n free first-order variables. We investigate the variable independence problem – originally introduced for databases – in the context of trees. In particular, we show how to decide whether a regular query is equivalent to a union of cartesian products, independently of the input tree. As an intermediate step, we reduce this problem to the problem of deciding whether the number of answers to a regular query is bounded by some constant, independently of the input tree. As a (non-trivial) generalization, we introduce variable independence w.r.t. a dependence forest between blocks of variables, which we prove to be decidable.

1 Introduction

Querying a tree consists of selecting nodes of its domain. This task has received a special interest from the XML community as it is fundamental to information extraction and document transformation. Several formalisms have been proposed to express unary queries [13], but less work has been done on n-ary queries, *ie* the selection of n-tuples of nodes [1, 10, 15]. Nevertheless, n-ary queries are of special interest, for instance to select tuples of the form *(name, addr, email, phone, fax)* in an XML document representing a directory. Since the arity of the query can easily get up to 10, efficiency becomes crucial to evaluate (n-ary) queries.

On the other hand, the notion of *variable independence* has been introduced in the context of (infinite) constraint databases [3, 5, 11, 12]. Query evaluation can be improved considerably when variables are independent. In particular, complexities of many evaluation algorithms on constraint databases are related to some independence between the components of the output tuples. For instance if the query can be decomposed into a Cartesian product of queries of lower arities, then all sub-queries of the product can be evaluated independently. *Orthographic dimension* has been proposed as a measure for variable independence [11]. It corresponds to the size of the largest block of dependent variables. It is shown in [5] that this notion is well-defined, since every pair of decompositions can be intersected into a decomposition of lower maximal block size.

However, in the context of constraint databases, the structure is infinite but fixed. A natural question is whether these results carry over into the context of an infinite number of finite tree structures. The notion of variable independence is also closely related to the representation of the set of answers of an n-ary query. In particular, if the variables are independent, this allows to represent the set of answers as a Cartesian product of sets of (sub)answers of lower sizes. More generally, *aggregated answers* have been introduced in [14] as a compact representation of the set of answers. Basically, these are multipartite dags such that each part corresponds to a free variable, and the branching structure of the parts is a tree. This branching structure somehow reflects the structure of the input tree. Consider for instance the tree of Fig. 1 representing a directory. Data are omitted in this picture. Consider the ternary query q which selects all triples (x, y, z) where x is labeled `person`, y is the first name of this person, and z one of its emails. Once x is selected, then y and z are independent. The set of answers to this query can be represented as a 3-partite graph with sets of vertices $\{V_x, V_y, V_z\}$, where there is a directed edge from a node

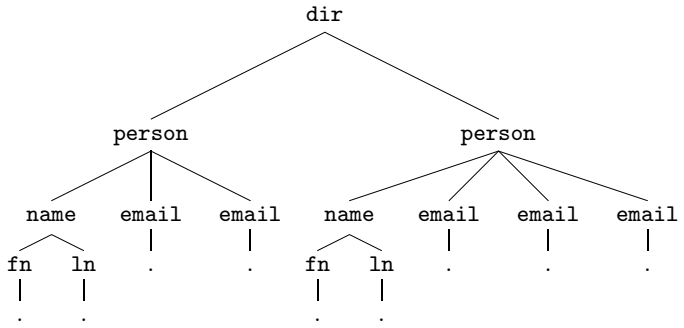

Fig. 1. A tree representing a directory

u of V_x to a node v of V_y, if v is the name of u. Similarly, edge relations between V_y and V_z correspond to the $(person, email)$ relations of the input tree. A person may have several emails, making this representation more compact than the set of all answers. Moreover, as argued in [14], this representation keeps the information on how the components of the output tuples are related in the tree. It is also particularly appropriate for post-processing tasks, such as answer searching, answer browsing, statistical computing, answer enumeration, and cascade-style querying. It raises the fundamental question of how compact this representation is. We answer this question by extending variable independence to *relative independence*. In particular, it allows more complex dependencies between variables, as emphasized by the previous example. We measure compactness in the settings of data-complexity, as we want to be independent of the query formalism.

To achieve the formal study of variable independence, we choose *Monadic Second Order Logic* to express n-ary queries, as it is often used as a yardstick logic in the context of trees [13].

We show that variable independence is decidable for MSO queries in trees, and that a decomposition is computable. We reduce this problem to testing whether a query is bounded, i.e. its number of answers is bounded by some constant independent of the input tree. We prove this problem to be decidable. We show that the notion of orthographic dimension is also well-defined in the context of trees. Finally, we introduce variable independence w.r.t. to a *dependence forest*, which introduces dependencies between blocks of variables.

Note that in the context of trees, a restricted notion of variable independence has been investigated in [15] and proved to be equivalent to non-ambiguity of tree automata.

Acknowledgments

We are very grateful to the anonymous referees for their valuable comments, to Bruno Courcelle who pointed out some references and to Slawomir Staworko for his careful re-reading.

2 Preliminaries

Although XML documents are usually modeled as unranked trees [13], we consider finite binary trees only. All our results can easily be lifted to unranked trees via a binary encoding [13].

Binary trees We consider a finite alphabet Σ consisting of symbols ranged over by a, f, g. A *binary tree* t over Σ is inductively defined by the following grammar: $t ::= a \mid a(t,t) \qquad a \in \Sigma$
The set of binary trees over Σ is denoted by T_Σ. The set of *nodes* N_t of a tree $t \in T_\Sigma$ is a set of words over $\{0,1\}$. We write ϵ for the empty word and $u.u'$ for the concatenation of u and u'. The set N_t is inductively defined by $N_a = \epsilon$ and $N_{a(t_0,t_1)} = \{\epsilon\} \cup \{b.u \mid b \in \{0,1\},\ u \in N_{t_b}\}$. Nodes $u \in N_t$ for which there is some $b \in \{0,1\}$ such that $u.b \in N_t$ are called *inner-nodes*. Other nodes are called *leaves*, and the node ϵ is called the *root* of t.
Let Σ' be another finite alphabet. Let $t \in T_\Sigma$ and $t' \in T_{\Sigma'}$ be two binary trees such that $N_t = N_{t'}$. The *product tree* $t \times t'$ is the tree over $\Sigma \times \Sigma'$ inductively defined by: $a \times b = (a,b)$, for all

$a \in \Sigma, b \in \Sigma'$, and $f(t_0, t_1) \times g(t'_0, t'_1) = (f, g)(t_0 \times t'_0, t_1 \times t'_1)$, for all $f \in \Sigma, g \in \Sigma'$, $t_0, t_1 \in T_\Sigma$ and $t'_0, t'_1 \in T_{\Sigma'}$. More generally, we can define the product of n trees modulo associativity of $\times$. Trees are also viewed as structures over the signature consisting of the binary successor symbols S_0 and S_1 and the unary symbols lab_a, for all $a \in \Sigma$, interpreted by their intuitive meanings.

MSO Monadic Second Order (MSO) logic extends first-order logic with quantification over sets. We consider first-order (resp. second-order) variables ranged over by x, y (resp. X, Y). MSO formulas consists of atomic formulas $lab_a(x)$, $S_0(x, y)$, $S_1(x, y)$ or $x \in X$, and are closed by boolean connectives $\wedge, \neg$ and quantification $\exists x, \exists X$. We let $t, \rho \models \phi$ denotes the satisfaction relation and say that the formula ϕ holds in the tree t under the variable assignment ρ. We refer the reader to [13] for more details about the semantics of MSO.

Definition 1 (regular n-ary queries). *Let $n \geq 0$. An n-ary query q is a mapping from trees $t \in T_\Sigma$ into subsets of N_t^n. It is regular (or MSO-definable) if there is an MSO-formula $\phi(x_1, \ldots, x_n)$ with n free first-order variables such that for all trees t, we have: $q(t) = \{(\rho(x_1), \ldots, \rho(x_n)) \mid t, \rho \models \phi(x_1, \ldots, x_n)\}$*

If $n = 0$ (resp. $n = 1$, $n = 2$), q is called Boolean (resp. unary, binary). We also say that ϕ defines q and denote q by q_ϕ.

Those queries are called regular since there is a close correspondence between MSO-definable queries and tree automata. In particular, it is well-known that every MSO-definable n-ary query $\phi(x_1, \ldots, x_n)$ on trees over Σ can be represented as a tree automaton A over the alphabet $\Sigma \times \{0, 1\}^n$ [15, 19, 4]. Moreover, we can assume that A is *canonical*, i.e. for any tree $t \in L(A)$, and any $i \in \{1, \ldots, n\}$, there is exactly one node of t such that the $(i + 1)$-th component of its label is 1 [19, 4]. The following holds: for all its nodes $u_1, \ldots, u_n$ of t, $t \models \phi(u_1, \ldots, u_n)$ iff $t \times \chi_{u_1} \times \cdots \times \chi_{u_n} \in L(A)$, where for all $i \in \{1, \ldots, n\}$, χ_{u_i} is the tree on $\{0, 1\}$ such that $N_t = N_{\chi_{u_i}}$ and all nodes except u_i are labeled 0.

3 Boundedness Properties of Regular Queries

In this section, we prove intermediate results which are useful for Section 4, but might be of independent interest.
Let $\phi(x_1, \ldots, x_n)$ be an MSO formula whose free variables are first-order. We say that ϕ is *bounded* if there is $K \in \mathbb{N}$ such that for all $t \in T_\Sigma$, the number of assignments ρ of variables x_is into N_t such that $t, \rho \models \phi$ is bounded by K.

Lemma 1. *Given an MSO formula $\phi(x_1, \ldots, x_n)$ with n free first-order variables $x_1, \ldots, x_n$, it is decidable whether ϕ is bounded and, in this case, a bound is computable.*

Proof. As said at the end of Section 2, ϕ can be represented as a canonical tree automaton A over $\Sigma \times \{0, 1\}^n$. Now, we can easily transform A in linear time into a bottom-up transducer T_A which takes trees t_1 over Σ as inputs and outputs trees t_2 over $\{0, 1\}^n$ such that $N_{t_1} = N_{t_2}$ and $t_1 \times t_2 \in L(A)$. Since for all trees t, we have $|T_A(t)| = |q_\phi(t)|$, it suffices to test whether $|T_A(t)|$ is bounded by some constant, which is decidable in polynomial time (in the size of T_A). This is called finite valuedness in [17, 18]. Moreover, for all fixed $k \in \mathbb{N}$, one can decide in non-deterministic polynomial time whether the number of images by T_A is greater than k [17, 18] (with a constant factor which is several exponentials in the size of k). Moreover, the bound is lesser than $2^{2^{p(|T_A|)}}$ for some polynomial p independent of T_A. Hence, based on a dichotomy algorithm, one can compute the smallest upper bound. The time complexity however is several exponentials in the size of T_A.□

Concerning time complexity, it is known that the size of the tree automaton associated with ϕ might be non-elementary in the size of ϕ [19], making the whole procedure possibly non-elementary. However, if the query is given by a canonical tree automaton A, testing boundedness becomes polynomial in the size of A, since testing finite valuedness of a tree transducer can be done in polynomial time [18, 17].

Lemma 1 could also be deduced from a result of [2]. This paper considers an extension of MSO on infinite trees with bounding quantifiers. In particular, for any MSO formula $\psi(X)$, the bounding quantified formula $\mathbb{B}X.\psi(X)$ holds in an infinite tree t, if there is a bound $b \in \mathbb{N}$ such that the size of any subset of the set of nodes of t that satisfies $\psi(X)$ is bounded by b.
Two fragments are proved to be decidable: formulas of the form $\neg\mathbb{B}X.\psi(X)$, where ψ is in MSO, and formulas built from arbitrary MSO formulas and $\mathbb{B}, \exists, \vee$ and $\wedge$.
We can easily reduce our problem to satisfiability of some formula in the first fragment. First, boundedness of an n-ary query reduces to boundedness of all its projections. Hence, we only need to consider unary queries. Now, from a formula $\psi(x)$ in one free variable, we construct a closed formula γ such that $\psi(x)$ is bounded iff γ is unsatisfiable. The formula γ is a conjunction $\gamma = \gamma_1 \wedge \gamma_2$. The first formula γ_1 checks whether the model is a tree (possibly infinite) of the form $\#(t_1, \#(t_2, \#(t_3, \ldots)$, for some fresh symbol $\# \notin \Sigma$ and $t_1, t_2, t_3, \ldots \in T_\Sigma$ are finite trees over Σ. The second formula γ_2 has the form $\neg\mathbb{B}X.\gamma'(X)$, where $\gamma'(X)$ is an MSO formula which holds in $\#(t_1, \#(t_2, \#(t_3, \ldots)$ under some assignment ρ if there is $i \geq 1$ such that $\rho(X)$ corresponds to the set of nodes u of t_i such that $t_i \models \psi(u)$. The formula γ' is defined by first choosing some node x_0 labeled $\#$ and then relating $\psi(x)$ under x_0.
However, we cannot benefit from this reduction if the query is given by a tree automaton (in term of time complexity).

An *equivalence relation on n-tuples* is a $2n$-ary query , often denoted $\equiv$, such that for all trees t, $\equiv(t)$ is an equivalence relation on N_t^n. We let $\equiv_t$ stands for $\equiv(t)$. It is regular if $\equiv$ is regular. We say that $\equiv$ is of bounded index if for all trees t, the number of $\equiv_t$-equivalence classes is bounded by some constant which does not depend on the tree. We can define a regular query which selects the minimal representatives of the equivalence classes, for some MSO-definable order on tuples. Hence, as a corollary of Lemma 1, we get:

Corollary 1 (bounded index property). *Let $\equiv$ be a regular equivalence relation on n-ary tuples. It is decidable whether $\equiv$ is of bounded index.*

Proof. Let t be a tree, we define a total order $\leq_t$ on N_t^n. It suffices to start from a total order on N_t and to extend it to a lexicographic order $\leq_t$ on N_t^n. Take for instance the lexicographic order on words over $\{0,1\}$ which is a total order on N_t. We can easily show that the query $t \mapsto \leq_t$ is regular. Now, we define the n-ary query $q_{min} : t \mapsto \{\overline{u} \mid \forall \overline{u}' \in N_t^n,\ \overline{u} \equiv_t \overline{u}' \implies \overline{u} \leq_t \overline{u}'\}$. The query q_{min} is regular. Finally it suffices to verify boundedness of q_{min}, which is decidable by Lemma 1. □

Beyond Trees

Deciding boundedness of an MSO formula can be done for classes of structures which are images of a regular set of trees by an MSO-transduction. We refer the reader to [6] for a definition of MSO-transductions. Given a set of integers $I \subseteq \mathbb{N}$, we say that I is *linear* if there are integers $\alpha_0, \ldots, \alpha_n \in \mathbb{N}$ such that $I = \{\alpha_0 + \sum_{i=1}^n \alpha_i x_i \mid x_1, \ldots, x_n \in \mathbb{N}\}$. It is *semi-linear* if it is a finite union of linear sets. Let σ be a signature and C be a class of σ-structures, θ an MSO-transduction from binary trees to σ-structures such that C is the image by θ of a set of binary trees. In [6], Courcelle proves[1] that given an MSO formula $\phi(X)$ over the signature σ with one free second-order variable X, the set $\{\#\rho(X) \mid M, \rho \models \phi(X),\ M \in C\}$ is semi-linear, and we can compute the coefficients of the polynomials if the transduction θ is known ($\#\rho(X)$ denotes the cardinality of $\rho(X)$). This is the case for instance for the class of graphs of clique-width less than k, for any fixed k [8]. To decide boundedness of an MSO formula $\phi(x)$ (where x is first-order), it suffices to compute the above coefficients for the formula $\Phi(X) = \forall x,\ x \in X \leftrightarrow \phi(x)$. The formula $\phi(x)$ is bounded iff the coefficients α_0 of the linear sets are the unique (possibly) non-null coefficients. Finally, boundedness of a formula $\phi(x_1, \ldots, x_n)$ reduces to boundedness of every projection of ϕ on a single variable x_i, for all $i \in \{1, \ldots, n\}$. Hence, boundedness is decidable, for instance, for structures of clique-width less than k, for any fixed k, or for unranked trees. However, to decide the bounded index property, we need an MSO-definable total order.

[1] Courcelle proves a more general result where several free variables are allowed

4 Variable Independence

The definition of variable independence was originally defined over a fixed structure [3, 12]. We state it over the class of binary trees. We let ϕ be an MSO formula with free variables $x_1, \ldots, x_n$, and $P = \{B_1, \ldots, B_k\}$ be a partition of $\{x_1, \ldots, x_n\}$. We write $\overline{x}_{B_i}$, $i = 1, \ldots, k$, to denote the tuple formed by variables of B_i given in order. We say that ϕ *conforms to* P, denoted $\phi \sim P$, if ϕ is equivalent to a formula of the form $\bigvee_{j=1}^{N} \phi_{j,1}(\overline{x}_{B_1}) \wedge \cdots \wedge \phi_{j,k}(\overline{x}_{B_k})$, where N is a natural and $\phi_{j,i}$ are MSO formulas with free variables in B_i. Note that if we require N to be equal to 1, the problem becomes easy, since it suffices to test whether ϕ is equivalent to the conjunction of the k projections of ϕ on the variables from each block B_i.

W.l.o.g., we assume that free variables of ϕ are ranked in order given by $B_1, \ldots, B_k$. In other words, we assume $x_1, \ldots, x_n = \overline{x}_{B_1}, \ldots, \overline{x}_{B_k}$ (modulo associativity). Now, for any $i \in \{1, \ldots, k\}$, and any tuples of variables $\overline{x}$, $\overline{y}$ such that $|\overline{x}| = |\overline{y}| = |B_i|$, we let $\psi^i_\phi(\overline{x}, \overline{y})$ be the formula defined by:

$$\psi^i_\phi(\overline{x}, \overline{y}) = \forall \overline{x}_{B_1} \ldots \forall \overline{x}_{B_{i-1}} \forall \overline{x}_{B_{i+1}} \ldots \forall \overline{x}_{B_k} \quad \phi(\overline{x}_{B_1}, \ldots, \overline{x}_{B_{i-1}}, \overline{x}, \overline{x}_{B_{i+1}}, \ldots, \overline{x}_{B_k}) \leftrightarrow \phi(\overline{x}_{B_1}, \ldots, \overline{x}_{B_{i-1}}, \overline{y}, \overline{x}_{B_{i+1}}, \ldots, \overline{x}_{B_k})$$

For any $i \in \{1, \ldots, k\}$, and any tree $t \in T_\Sigma$, we let R^t_i be the binary relation on $N_t^{|B_i|}$ defined by $R^t_i = \{(\overline{u}, \overline{v}) \mid t \models \psi^i_\phi(\overline{u}, \overline{v})\}$. Intuitively, $\overline{u}$ and $\overline{v}$ are equivalent if one can substitute $\overline{u}$ with $\overline{v}$, in any tuple selected by ϕ whose i-th block is $\overline{u}$, and conversely.

Lemma 2. *Let $i \in \{1, \ldots, k\}$ and $t \in T_\Sigma$. The following are true:*

1. *R^t_i is an equivalence relation on $N_t^{|B_i|}$;*
2. *if ϕ is equivalent to some formula of the form $\bigvee_{j=1}^{N} \phi_{j,1}(\overline{x}_{B_1}) \wedge \cdots \wedge \phi_{j,k}(\overline{x}_{B_k})$, then the number of R^t_i-equivalence classes is bounded by 2^N.*

Proof. We only prove the second point, as the first is easy. Given some natural $i \in \{1, \ldots, k\}$, some tree t and some node tuples $\overline{u}, \overline{v}$ of length $|B_i|$, we let $\overline{u} \equiv^t_i \overline{v}$ if there is some set $F \subseteq \{1, \ldots, N\}$ (possibly empty) such that for all $j \in F$, we have $t \models \phi_{j,i}(\overline{u}) \wedge \phi_{j,i}(\overline{v})$, and for all $j \in \{1, \ldots, N\} \backslash F$, we have $t \models \neg\phi_{j,i}(\overline{u}) \wedge \neg\phi_{j,i}(\overline{v})$. We can easily prove that $\equiv^t_i$ is an equivalence relation on $N_t^{|B_i|}$ which has at most 2^N equivalence classes. We now prove that $\equiv^t_i$ is a refinement of R^t_i, which will be sufficient to conclude. Let $\overline{u}, \overline{v}$ be two node tuples of length $|B_i|$ such that $\overline{u} \equiv^t_i \overline{v}$. Let $\overline{w}_1, \ldots, \overline{w}_{i-1}, \overline{w}_{i+1}, \ldots, \overline{w}_k$ be node tuples. We have $t \models \phi(\overline{w}_1, \ldots, \overline{w}_{i-1}, \overline{u}, \overline{w}_{i+1}, \ldots, \overline{w}_k)$ iff $t \models \bigvee_{j=1}^{N} \phi_{j,1}(\overline{w}_1) \wedge \cdots \wedge \phi_{j,i}(\overline{u}) \wedge \cdots \wedge \phi_{j,k}(\overline{w}_k)$ iff (by definition of $\equiv^t_i$) $t \models \bigvee_{j=1}^{N} \phi_{j,1}(\overline{w}_1) \wedge \cdots \wedge \phi_{j,i}(\overline{v}) \wedge \cdots \wedge \phi_{j,k}(\overline{w}_k)$ iff $t \models \phi(\overline{w}_1, \ldots, \overline{w}_{i-1}, \overline{v}, \overline{w}_{i+1}, \ldots, \overline{w}_k)$. Hence we get $(\overline{u}, \overline{v}) \in R^t_i$. □

Lemma 3. *If for all $i \in \{1, \ldots, k\}$, there is some $m_i \in \mathbb{N}$ such that for all trees $t \in T_\Sigma$, the number of equivalence classes of R^t_i is bounded by m_i, then $\phi \sim P$.*

Proof. Let $i \in \{1, \ldots, n\}$. We define a successor relation between equivalence classes, then we introduce formulas $cl^l_i(\overline{x})$,$l = 1, \ldots, m_i$, to define the l-th equivalence class of R^t_i, in any tree $t \in T_\Sigma$.

As already seen in the proof of Corollary 1, there is an MSO-definable total order $\leq$ on node tuples. We now define a successor relation S^i_ϕ between the minimal representatives (for $\leq$) of the equivalence relation defined by ψ^i_ϕ. Now, let the formula $min^i_\phi(\overline{x})$ holds if $\overline{x}$ is the minimal representative of some equivalence class. It can be defined by $\forall \overline{y},\ \psi^i_\phi(\overline{x}, \overline{y}) \rightarrow \overline{x} \leq \overline{y}$. The relation S^i_ϕ is now defined by the following MSO formula:

$$S^i_\phi(\overline{x}, \overline{y}) = \overline{x} < \overline{y} \wedge min^i_\phi(\overline{x}) \wedge min^i_\phi(\overline{y}) \wedge \neg(\exists \overline{z}, \overline{x} < \overline{z} < \overline{y} \wedge min^i_\phi(\overline{z}))$$

We let $s_0(\overline{x})$ stand for $\neg\exists \overline{z}, S^i_\phi(\overline{z}, \overline{x})$ and $s_l(\overline{x})$, $l \in \mathbb{N}$ stands for $\exists \overline{y},\ s_{l-1}(\overline{y}) \wedge S^i_\phi(\overline{y}, \overline{x})$. We now define $cl^l_i(\overline{x})$ by $\exists \overline{y}, s_l(\overline{y}) \wedge \psi^i_\phi(\overline{x}, \overline{y})$, for all $1 \leq l \leq m_i$. Intuitively $s_l(\overline{x})$ holds in t under some

assignment ρ if $\rho(\overline{x})$ is the minimal representative of the l-th equivalence class of R_i^t, while $cl_i^l(\overline{x})$ holds in t under ρ if $\rho(\overline{x})$ belongs to the l-th equivalence class of R_i^t.
Finally, we let L be the set of tuples of naturals $\overline{l} = (l_1, \ldots, l_k)$ such that $1 \leq l_i \leq m_i$, $i = 1, \ldots, k$, and we denote by $\beta^{\overline{l}}(\overline{x}_1)$ the formula $\exists \overline{x}_2 \ldots \overline{x}_k,\ \phi(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_k) \wedge \bigwedge_{j=1}^{k} cl_j^{l_j}(\overline{x}_j)$. We let $\phi^{\overline{l}}$ be the formula $\beta^{\overline{l}}(\overline{x}_1) \wedge cl_2^{l_2}(\overline{x}_2) \cdots \wedge cl_k^{l_k}(\overline{x}_k)$. We now prove that ϕ is equivalent to $\bigvee_{\overline{l} \in L} \phi^{\overline{l}}$.

Let $t \in T_\Sigma$ and $\overline{u}_1, \ldots, \overline{u}_k$ node tuples of t such that $t \models \phi(\overline{u}_1, \ldots, \overline{u}_k)$. For all $i \in \{1, \ldots, k\}$, we necessarily have $t \models \psi_\phi^i(\overline{u}_i, \overline{u}_i)$. Hence, there is some natural $l_i \in \{1, \ldots, m_i\}$ such that $t \models cl_i^{l_i}(\overline{u}_i)$, $i = 1, \ldots, k$. Let $\overline{l} = (l_1, \ldots, l_k)$. It is easy to see that $t \models \phi^{\overline{l}}(\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_k)$.
Conversely, suppose there is some tuple $\overline{l} = (l_1, \ldots, l_k) \in L$ such that $t \models \phi^{\overline{l}}(\overline{u}_1, \ldots, \overline{u}_k)$. In particular, we have $t \models \beta^{\overline{l}}(\overline{u}_1)$, hence there are some node tuples $\overline{u}'_2, \ldots, \overline{u}'_k$, such that $t \models \phi(\overline{u}_1, \overline{u}'_2, \ldots, \overline{u}'_k)$, and for each $i \in \{2, \ldots, k\}$, $t \models cl_i^{l_i}(\overline{u}'_i)$. We now prove by induction that for all $p \in \{2, \ldots, k\}$, we have $t \models \phi(\overline{u}_1, \ldots, \overline{u}_p, \overline{u}'_{p+1}, \ldots, \overline{u}'_k)$. It is true for $p = 1$ by hypothesis. Suppose that it is true at rank $p > 1$. Since $t \models cl_{p+1}^{l_{p+1}}(\overline{u}_{p+1})$, we also have $t \models \psi_\phi^{p+1}(\overline{u}_{p+1}, \overline{u}'_{p+1})$. By induction hypothesis, we have $t \models \phi(\overline{u}_1, \ldots, \overline{u}_p, \overline{u}'_{p+1}, \ldots, \overline{u}'_k)$, and by definition of ψ_ϕ^{p+1}, we easily get $t \models \phi(\overline{u}_1, \ldots, \overline{u}_p, \overline{u}_{p+1}, \overline{u}'_{p+2}, \ldots, \overline{u}'_k)$. □

As a consequence of Lemma 2 and 3, and Corollary 1, we get the main result:

Theorem 1. *Given an MSO formula ϕ with free variables $x_1, \ldots, x_n$ and a partition P of $\{x_1, \ldots, x_n\}$, it is decidable whether $\phi \sim P$ holds or not. If it holds, a decomposition of ϕ is computable.*

Orthographic Dimension The notion of orthographic dimension has been introduced in [11] to measure the degree of independence between variables, over a fixed database. We define it for any tree structure. Given a formula $\phi(x_1, \ldots, x_n)$, the *orthographic dimension* d_ϕ of ϕ is defined by $d_\phi = min_{P\,:\,\phi \sim P}\ max_{B \in P}\ |B|$.
Theorem 1 gives us a naive algorithm to compute d_ϕ: for each partition P of $\{1, \ldots, n\}$, test whether $\phi \sim P$ and compute $max_{B \in P}\ |B|$. But as we next show, we can restrict the tests to 2-partitions.
Given two partitions P, P' of $\{1, \ldots, n\}$, we write $P \sqcap P'$ for the refinement of P and P'. Formally, we have $P \sqcap P' = \{B \cap B' \mid B \in P, B' \in P'\} - \varnothing$. Theorem 1 of [5] states well-definedness of the notion of orthographic dimension, for first-order logic on any vocabulary, over a fixed structure. In particular, it means that there is a unique partition whose largest block is equal to the orthographic dimension, such that the formula conforms to it. However, the proof given in [5] also works if the structure is not fixed. Moreover, it also works as soon as the logic contains first-order quantifiers, negations, and Boolean connectives. Hence, it also proves the following:

Theorem 2. *Let ϕ be an MSO formula in n free first-order variables $x_1, \ldots, x_n$, and P, P' be two partitions of $\{1, \ldots, n\}$. If $\phi \sim P$ and $\phi \sim P'$, then $\phi \sim P \sqcap P'$.*

Now, let $\mathcal{P} = \{P \mid P \text{ is a 2-partition of } \{1, \ldots, n\} \text{ and } \phi \sim P\}$. From Theorem 2, we can deduce that the orthographic dimension of ϕ is the size of the largest block of $\sqcap_{P \in \mathcal{P}} P$. Moreover, by Theorem 1, we can compute a decomposition which corresponds to the orthographic dimension.

Relation to the Answer Set Representation

If $\phi \sim P$, then for any tree $t \in T_\Sigma$, we can represent $q_\phi(t)$ by an aggregated answer of size $O(n|t|^{d_\phi})$, computable in time $O(n|t|^{d_\phi})$ (ϕ is assumed to be fixed). Indeed, ϕ is equivalent to a (computable) formula of the form:$\bigvee_{i=1}^{N} \phi_{i,1}(x_{B_1}) \wedge \ldots \wedge \phi_{i,k}(x_{B_k})$ for some natural N. For every $i \in \{1, \ldots, N\}$, and $j \in \{1, \ldots, k\}$, we compute an automaton $A_{i,j}$ over $\Sigma \times \{0,1\}^{|B_j|}$ such that we can compute in time $O(|t|^{|B_j|}|A_{i,j}|)$ the set $q_{\phi_{i,j}}(t)$ [7]. The answer representation can be identified to the collection of k-tuples $(q_{\phi_{i,1}}(t), \ldots, q_{\phi_{i,k}}(t))$.

5 Relative Variable Independence

In this section, we generalize variable independence w.r.t. a partition to variable independence w.r.t. a dependence forest on free variables. Consider for instance the tree of Fig. 1, and let $\phi(x, y, z)$ be an MSO formula, where x denotes a person, y its first name and z one of its emails. Once the interpretation of x is fixed, then y and z are independent. We call this *relative independence*. Let T be the tree $x(y, z)$, we say that ϕ conforms to T, denoted $\phi \sim T$. We next show that relative independence is decidable as a consequence of Theorem 1.

As a slight generalization, we allow dependence forests to specify dependences between sets of variables instead of single variables. For example, if a formula $\phi(x, y, z, w)$ conforms to some dependence forest $\{x, y\}(\{z\}, \{w\})$, it means that once x and y are selected, then z and w are independent.

Formally, let $V = \{x_1, \ldots, x_n\}$ be a finite set of variables. A *dependence forest* F over V is a forest whose nodes are labeled by subsets of 2^V and such that the set of labels occurring in F form a partition of V. If F has only one root, then it is called a *dependence tree*. We often denote by $\{T_1, \ldots, T_k\}$ a dependence forest consisting of the dependence trees $T_1, \ldots, T_k$, and by $V'(F)$ a dependence tree consisting of a dependence forest F rooted by a set $V' \subseteq V$. Although we take a graph point of view, we use the same notations as for binary trees to denote the set of nodes of some forest F over V and its labeling function, respectively by N_F and $lab_F : N_F \to 2^V$. Finally, we confuse the set and tuple notations for variables, so that we sometimes write $V = \overline{x}$, for some tuple of variables $\overline{x}$.

Let $\phi(x_1, \ldots, x_n)$ be an MSO formula in n free first-order variables $x_1, \ldots, x_n$. Let μ be a mapping from N_F into MSO formulas. We say that μ is *admissible* for F if for all nodes $u, v \in N_F$, if u is the parent of v, then $\mu(v)$ is an MSO formula ψ whose free variables are $lab_F(u) \cup lab_F(v)$; if u is the root of F, then we require that $\mu(u)$ is an MSO formula whose free variables are $lab_F(u)$. If μ is admissible, we naturally extend it to an MSO formula $\mu(F) = \bigwedge_{u \in N_F} \mu(u)$ in free variables $x_1, \ldots, x_n$.

Definition 2. *We say that ϕ conforms to F, denoted $\phi \sim F$, if there is a finite sequence $\mu_1, \ldots, \mu_N$ of admissible mappings for F such that ϕ is equivalent to $\bigvee_{i=1}^{N} \mu_i(F)$.*

We prove decidability of relative independence. We start by a base lemma (Lemma 4), for forests of the form $\overline{x}(\{\overline{y}, \overline{z}\})$. Then we give a recursive algorithm for the general case, that uses Lemma 4 and Theorem 1.

Lemma 4. *Given an MSO formula $\phi(\overline{x}, \overline{y}, \overline{z})$ with free variables $\overline{x}, \overline{y}, \overline{z}$, it is decidable whether ϕ is equivalent to some disjunction of the form $\bigvee_{i=1}^{n} \alpha_i(\overline{x}, \overline{y}) \wedge \beta_i(\overline{x}, \overline{z})$, for some natural n, and MSO formulas α_i, β_i for $i = 1, \ldots, n$. Moreover if it holds, a disjunction is computable.*

Proof. Intuitively, we fix the interpretation of $\overline{x}$ by extending the alphabet with Boolean tuples. This gives a formula $\phi_{\overline{x}}(\overline{y}, \overline{z})$, and we test whether $\phi_{\overline{x}}(\overline{y}, \overline{z}) \sim \{\overline{y}, \overline{z}\}$.

More formally, we first transform $\phi(\overline{x}, \overline{y}, \overline{z})$ into $\phi_{\overline{x}}(\overline{y}, \overline{z})$, interpreted on trees over the alphabet $\Sigma \times \{0, 1\}^m$, where $m = |\overline{x}|$, such that the following property holds (P_1): for all trees t, all nodes $u_1, \ldots, u_m \in N_t$, and all node tuples $\overline{v}, \overline{w}$, we have $t \models \phi(u_1, \ldots, u_m, \overline{v}, \overline{w})$ iff $t \times \chi_{u_1} \times \cdots \times \chi_{u_m} \models \phi_{\overline{x}}(\overline{v}, \overline{w})$, where the trees χ_{u_i} are defined at the end of Section 2.

This can be done by repeating exhaustively the following transformation rule on ϕ: replace each atom of the form $P(\overline{x}_1, x, \overline{x}_2)$, where x is the i-th component of $\overline{x}$ and is a free occurrence in ϕ, by $\exists x,\ \bigvee_{(f,\overline{b}) \in \Sigma \times B_i} lab_{(f,\overline{b})}(x) \wedge P(\overline{x}_1, x, \overline{x}_2)$, where $B_i \subseteq \{0, 1\}^m$ is the set of Boolean tuples whose i-th component is 1 for $i = 1, \ldots, m$. Hence, $\phi(\overline{x}, \overline{y}, \overline{z})$ rewrites to some formula $\phi'_{\overline{x}}(\overline{y}, \overline{z})$. We define $\phi_{\overline{x}}$ by $\phi'_{\overline{x}} \wedge \phi_{can}$, where ϕ_{can} is a sentence which ensures that all models $t \in T_{\Sigma \times \{0,1\}^m}$ of $\phi_{\overline{x}}$ are canonical (*ie* all nodes except one have their i-th component set to 0 $i = 1, \ldots, n$). We call R_1 the transformation from ϕ to $\phi_{\overline{x}}$.

Then it suffices to test whether $\phi_{\overline{x}}(\overline{y}, \overline{z}) \sim \{\overline{y}, \overline{z}\}$, which is decidable by Theorem 1. If it holds, then $\phi_{\overline{x}}(\overline{y}, \overline{z})$ is equivalent to some formula of the form $\psi_{\overline{x}}(\overline{y}, \overline{z}) = \bigvee_{i=1}^{n} \alpha_{i,\overline{x}}(\overline{y}) \wedge \beta_{i,\overline{x}}(\overline{z})$, for some MSO formulas $\alpha_{i,\overline{x}}, \beta_{i,\overline{x}}$. We next consider the following transformation rule R_2: replace each

Algorithm 1 Testing Relative Independence

procedure D(ϕ,F)
case F is a leaf or is of the form $\overline{x}(\overline{y})$:
return ϕ

case F is of the form $\{\overline{x}_1, \ldots, \overline{x}_k\}$:
test whether $\phi \sim \{\overline{x}_1, \ldots, \overline{x}_k\}$ as in the proof of Theorem 1 and return a decomposition. Otherwise breaks.

case F is of the form $\overline{x}(\overline{y}, \overline{z})$:
test whether $\phi \sim \overline{x}(\overline{y}, \overline{z})$ as in the proof of Lemma 4 and return a decomposition. Otherwise breaks.

case F is of the form $\overline{x}(\overline{y}(F'), F'')$:
$\overline{z}', \overline{z}'' \leftarrow$ sets of variables occurring in F', F''
$\bigvee_i \alpha_i(\overline{x}, \overline{y}, \overline{z}') \wedge \beta_i(\overline{x}, \overline{z}'') \leftarrow \mathrm{D}(\phi, \overline{x}(\{\overline{y} \cup \overline{z}', \overline{z}''\}))$
return $\bigvee_i \mathrm{D}(\alpha_i, \overline{y}(F', \overline{x})) \wedge \mathrm{D}(\beta_i, \overline{x}(F''))$

case F is of the form $\{T_1, \ldots, T_k\}$:
for $i \in \{1, \ldots, k\}$ **do**
$\overline{x}_i \leftarrow$ set of variables occurring in T_i
$\bigvee_{i=1}^n \bigwedge_{j=1}^k \alpha_i^j(\overline{x}_j) \leftarrow \mathrm{D}(\phi, \{\overline{x}_1, \ldots, \overline{x}_k\})$
return $\bigvee_{i=1}^n \mathrm{D}(\alpha_i^1, T_1) \wedge \cdots \wedge \mathrm{D}(\alpha_i^k, T_k)$

atom of the form $lab_{f,\overline{b}}(x)$ by $lab_f(x) \wedge \bigwedge_{b_i=1} x_i = x$, where b_i denotes the i-th component of $\overline{b}$, $i = 1, \ldots, m$. Suppose that $\overline{x} = x_1, \ldots, x_m$. Applying exhaustively this transformation rule on $\psi_{\overline{x}}$ leads to a formula $\psi(\overline{x}, \overline{y}, \overline{z})$ of the form $\bigvee_{i=1}^n \alpha_i(\overline{x}, \overline{y}) \wedge \beta_i(\overline{x}, \overline{z})$ interpreted on trees over Σ. We have the following property (P_2): for all trees t, all nodes $u_1, \ldots, u_m \in N_t$, and all node tuples $\overline{v}, \overline{w}$, we have $t \models \psi(u_1, \ldots, u_m, \overline{v}, \overline{w})$ iff $t \times \chi_{u_1} \times \cdots \times \chi_{u_m} \models \psi_{\overline{x}}(\overline{v}, \overline{w})$.

Finally, we prove this algorithm to be correct. Suppose that it returns a decomposition. By combining properties P_1 and P_2, we can prove correctness of this decomposition. Conversely, suppose that ϕ is equivalent to some formula of the form $\bigvee_{i=1}^n \alpha_i(\overline{x}, \overline{y}) \wedge \beta_i(\overline{x}, \overline{z})$. It is easy to see that $\phi_{\overline{x}}$ is equivalent (here we use canonicity of its models) to $\bigvee_{i=1}^n \alpha_{i,\overline{x}}(\overline{y}) \wedge \beta_{i,\overline{x}}(\overline{z})$, where $\phi_{\overline{x}}$, $\alpha_{i,\overline{x}}(\overline{y})$ and $\beta_{i,\overline{x}}(\overline{z})$ are obtained by applying the rewrite rule R_1 on respectively ϕ, $\alpha_i(\overline{x}, \overline{y})$ and $\beta_i(\overline{x}, \overline{z})$. Hence, $\phi_{\overline{x}} \sim \{\overline{y}, \overline{z}\}$, and the proof follows since the algorithm of the proof of Theorem 1 is sound. □

Now, we extend the result of Lemma 4 to full independence forests:

Theorem 3. *Given a formula ϕ in free variables $V = \{x_1, \ldots, x_n\}$ and a dependence forest F over V, it is decidable whether $\phi \sim F$ holds or not.*

Proof. Consider Algorithm 1. The inputs are a formula ϕ with free variables V and a dependence forest F over V. The symbols $T_1, \ldots, T_k$ denote dependence trees while F', F'' denote (possibly empty) dependence forests.
First note that the algorithm terminates. Indeed, the number of nodes of the forest strictly decreases at each recursive call except for the 4th case when F'' is empty, but in this case the height of the forest strictly decreases.
Now, we can prove (inductively and by using Theorem 1 and Lemma 4 for the basic cases) that if this algorithm returns a formula, then it is a decomposition of the input formula ϕ w.r.t. the input forest F. It suffices to push up the disjunctive connectives to get a sequence of admissible mappings for F.
Conversely, let ϕ (resp. F) be an input formula (resp. an input dependence forest), such that we have $\phi \sim F$. We prove by induction that the algorithm outputs a decomposition. We use the fact that the decomposition is not arbitrary, but has a particular form, derived from the algorithms given in the proofs of Theorem 1 and Lemma 4. The first case is obvious, and the two next

cases have already been proved. First remark that we have the following property (*): let F be a dependence forest, γ_1, γ_2 two formulas, α a sentence and $\beta(\overline{x})$ a formula such that $\overline{x}$ is a label of F. If $\gamma_1 \sim F$ and $\gamma_2 \sim F$, we have $\gamma_1 \vee \gamma_2 \sim F$, $\gamma_1 \wedge \gamma_2 \sim F$, $\neg\gamma_1 \sim F$, $\alpha \wedge \gamma_1 \sim F$, and $\beta(\overline{x}) \wedge \gamma_1 \sim F$.

Suppose that F is of the form $\overline{x}(\overline{y}(F'), F'')$. Since $\phi \sim F$, in particular, $\phi \sim \overline{x}(\overline{y} \cup \overline{z}', \overline{z}'')$, where $\overline{z}', \overline{z}''$ are defined as in Algorithm 1. Now, we inspect the proof of Lemma 4. Let $\phi_{\overline{x}}$ be the result of applying the rewriting rule R_1 of this proof. It is clear, by hypothesis, that we have $\phi_{\overline{x}} \sim \{\overline{y} \cup \overline{z}', \overline{z}''\}$. Hence, algorithm of the proof of Theorem 1 outputs a decomposition of the form $\bigvee_{\overline{l} \in L} \beta_1^{\overline{l}}(\overline{y}, \overline{z}') \wedge cl_2^{l_2}(\overline{z}'')$, exactly as defined in the proof of Theorem 1. We let ψ^{-1} be the result of applying exhaustively the rewrite rule R_2 of the proof of Lemma 4 on ψ, for all formulas ψ. Hence, $\mathrm{D}(\phi, \overline{x}(\overline{y} \cup \overline{z}', \overline{z}''))$ returns the formula $\bigvee_{\overline{l} \in L} (\beta_1^{\overline{l}})^{-1}(\overline{x}, \overline{y}, \overline{z}') \wedge (cl_2^{l_2})^{-1}(\overline{x}, \overline{z}''))$. It remains to prove that formulas $(\beta_1^{\overline{l}})^{-1}$ and $(cl_2^{l_2})^{-1}$ satisfy $(\beta_1^{\overline{l}})^{-1} \sim \overline{y}(\overline{x}, F')$ and $(cl_2^{l_2})^{-1} \sim \overline{x}(F'')$. We only prove it for formulas $(cl_2^{l_2})^{-1}$, as the proof for formulas $(\beta_1^{\overline{l}})^{-1}$ is analogous. So let us fix some natural l_2. By going back to the definition of formula $cl_2^{l_2}$, we can prove that $(cl_2^{l_2})^{-1}$ is equivalent to a formula of the form $\Gamma = \exists \overline{z_0},\ \gamma(\overline{x}, \overline{z}_0) \wedge \forall \overline{u}, \phi(\overline{x}, \overline{u}, \overline{z}'') \leftrightarrow \phi(\overline{x}, \overline{u}, \overline{z}_0)$, for some γ. Now, since $\phi \sim F$, it easy to see that ϕ is equivalent to a formula of the form $\Psi = \bigvee_{i=1}^{n} \epsilon_i^1(\overline{x}, \overline{y}, \overline{z}') \wedge \epsilon_i^2(\overline{x}, \overline{z}'')$, such that $\epsilon_i^1 \sim \overline{y}(\overline{x}, F')$ and $\epsilon_i^2 \sim \overline{x}(F'')$ for $i = 1, \ldots, n$. We replace in Γ the formula ϕ by Ψ, and, after a series of rewritings (by pushing up disjunctions and pushing down quantifiers), we can prove that $(cl_2^{l_2})^{-1}$ is equivalent to a formula of the form $\bigvee_{i=1}^{n} \gamma_i(\overline{x}) \epsilon_i^2(\overline{x}, \overline{z}'') \vee \bigvee_{P \subseteq \{1,\ldots,n\}} \gamma'_P(\overline{x}) \wedge \bigwedge_{i \in P} \neg \epsilon_i^2(\overline{x}, \overline{z}'')$, for formulas γ_i, γ'_P depending only on i and P. The conclusion follows by property $(*)$ and the fact that every ϵ_i^2 satisfies $\epsilon_i^2 \sim \overline{x}(F'')$.

Suppose now that F is of the form $\{T_1, \ldots, T_k\}$, let $\overline{x}_i$ be the variables occurring in T_i for $i = 1, \ldots, k$, and let $P = \{\overline{x}_1, \ldots, \overline{x}_k\}$. Since $\phi \sim F$, in particular, $\phi \sim P$. Hence $\mathrm{D}(\phi, P)$ outputs a decomposition of the form $\bigvee_{\overline{l} \in L} \beta_1^{\overline{l}}(\overline{x}_1) \wedge cl_2^{l_2}(\overline{x}_2) \wedge \ldots \wedge cl_k^{l_k}(\overline{x}_k)$, exactly as defined in the proof of Theorem 1. We have to prove that for all $\overline{l} = (l_1, \ldots, l_k) \in L$, we have $\beta_1^{\overline{l}} \sim T_1$, and $cl_i^{l_i} \sim T_i$ for $i = 2, \ldots, k$. This is sufficient, since by induction hypothesis, D will output a decomposition of every $\beta_1^{\overline{l}}$ and $cl_i^{l_i}$. We only prove it for formulas $cl_i^{l_i}(\overline{x}_i)$, as it is similar for formulas $\beta_1^{\overline{l}}$. Let us fix some $\overline{l}$ and i. We come back to the definition of $cl_i^{l_i}$, and we can easily show that it is of the form $\exists \overline{y}, \gamma(\overline{y}) \wedge \psi_\phi^i(\overline{x}_i, \overline{y})$, for some formula γ which selects the minimal representatives of the l_i-th equivalence class of the relation defined by ψ_ϕ^i, where ψ_ϕ^i has been defined in Section 4. Now, since $\phi \sim F$, ϕ is equivalent to a formula of the form $\Psi = \bigvee_{p=1}^{n} \bigwedge_{j=1}^{k} \epsilon_p^j(\overline{x}_j)$ such that every ϵ_p^j satisfy $\epsilon_p^j \sim T_j$. Next, in $\exists \overline{y}, \gamma(\overline{y}) \wedge \psi_\phi^i(\overline{x}_i, \overline{y})$, we replace ϕ by Ψ (ψ_ϕ^i can be viewed as the result of applying the function $\psi_\cdot^i$ on ϕ, and we just replace $\psi_\phi^i(\overline{x}_i, \overline{y})$ by $\psi_\Psi^i(\overline{x}_i, \overline{y})$). We get a formula equivalent to $cl_i^{l_i}$ which, after a series of rewritings preserving equivalence (by moving up disjunctions and pushing down quantifiers), rewrites to a formula (equivalent to $cl_i^{l_i}$) of the form $\bigvee_{p=1}^{n} \phi_p \wedge \epsilon_p^i(\overline{x}_i) \vee \bigvee_{Q \subseteq \{1,\ldots,n\}} \psi_Q \wedge \bigwedge_{p \in Q} \neg \epsilon_p^i(\overline{x}_i)$, for some closed formulas ϕ_p, ψ_Q depending on p and Q. The conclusion follows by using property (*) and the fact that every ϵ_p^i satisfies $\epsilon_p^i \sim T_i$.□

Similarly as the case of variable independence, if $\phi(x_1, \ldots, x_n)$ conforms to F, then for any tree $t \in T_\Sigma$, $q_\phi(t)$ can be represented by an aggregated answer of size $O(n|t|^b)$ (ϕ is assumed to be fixed, and necessarily $|F| \leq n$). The parameter b denotes the maximal sum of the size of a label of F plus the size of the label of its father if it exists.

Note that variable independence w.r.t. a dependence forest subsumes variable independence w.r.t. a partition, since a partition can be viewed as a dependence forest consisting of a set of leaves. Moreover, as stated by the next theorem, there is an MSO formula ϕ such that there is no dependence forest F such that: (i) labels of F are singletons (ii) $\phi \sim F$. Nevertheless, we know that on trees, every MSO formula is equivalent to an existentially quantified Boolean combination of MSO formulas in two free variables [16, 10].

Theorem 4. *There is an MSO formula ϕ such that there is **no** dependence forest F whose labels are singletons and such that $\phi \sim F$.*

Proof. Let $x \preceq y$ be an MSO formula which holds in a tree if y is a descendant of x. It is well-known that it can easily be defined as the reflexive and transitive closure of $S_1 \vee S_2$, this closure being definable in MSO.
Now, let $\phi(x, y, z)$ be an MSO formula defined by:

$$\begin{aligned}\phi(x,y,z) = \exists \alpha \quad & \alpha \preceq x \wedge \alpha \preceq y \wedge \alpha \preceq z \\ & \wedge \forall \alpha' \ \alpha' \preceq x \wedge \alpha' \preceq y \implies \alpha' \preceq \alpha \\ & \wedge \forall \alpha' \ \alpha' \preceq x \wedge \alpha' \preceq z \implies \alpha' \preceq \alpha\end{aligned}$$

For all trees t and all nodes u, v, w of t, we have $t \models \phi(u, v, w)$ iff the least common ancestor of u and v is equal to the least common ancestor of u and w.

We now prove by applying algorithm 1 that there is no forest F whose labels are singletons such that $\phi \sim F$, by proving it for each forest over $\{x, y, z\}$. Let $n \geq 0$ and t_n be the tree over the alphabet $\{a\}$ inductively defined by $t_0 = a$ and $t_n = a(t_{n-1}, a)$. For all n, we denote by $v_0, v_1, \ldots, v_n$ the nodes $1^n, 1^{n-1}, \ldots, \epsilon$ respectively, and, if $n > 0$, by $w_1, \ldots, w_n$ the nodes $1^{n-1}.2, 1^{n-2}.2, \ldots, 2$ respectively.

1. $F = \{x, y, z\}$ or $F = \{x, y(z)\}$ or $F = \{x, z(y)\}$. We apply algorithm 1. The formula $\psi^1_\phi(x, x')$ is defined by $\forall y, z \ \phi(x, y, z) \leftrightarrow \phi(x', y, z)$. We prove that for all n, and all $i, j \leq n$, $v_i \neq v_j$ implies $t_n \not\models \psi^1_\phi(v_i, v_j)$. Indeed, if $v_i \neq v_j$ such that $desc(v_i, v_j)$, then we have $t \models \phi(v_i, v_i, v_j)$ but $t \not\models \phi(v_j, v_i, v_j)$. Hence, the number of classes of the equivalence relation defined by ψ^1_ϕ is at least n, which is unbounded. So Algorithm 1 breaks;
2. $F = \{y, x(z)\}$ or $F = \{y, z(x)\}$. The formula $\psi^2_\phi(y, y')$ is defined by $\forall x, y, z \ \phi(x, y, z) \leftrightarrow \phi(x, y', z)$. We prove that for all n, and all $i, j \leq n$, $v_i \neq v_j$ implies $t_n \not\models \psi^2_\phi(v_i, v_j)$. Indeed, if $v_i \neq v_j$ such that $desc(v_i, v_j)$, then we have $t \models \phi(v_j, v_i, v_i)$ but $t \not\models \phi(v_j, v_j, v_i)$. Hence, the number of classes of the equivalence relation defined by ψ^2_ϕ is at least n, which is unbounded. So Algorithm 1 breaks;
3. $F = \{z, y(x)\}$ or $F = \{z, x(y)\}$. Those cases are symmetric to the previous ones.
4. $F = x(y, z)$. We let $\psi(y, y') = \forall z, \phi_x(y, z) \leftrightarrow \phi_x(y', z)$ where ϕ_x has been defined in the proof of Lemma 4. By definition of Algorithm 1, if the equivalence relation defined by ψ has an unbounded index, then the algorithm fails. This is what we next prove. Let $n \leq 0$. We fix x by a Boolean in the tree t_n: we let t'_n be the tree over $\{a\} \times \{0, 1\}$ such that $N_{t_n} = N_{t'_n}$ and all nodes are labeled $(a, 0)$ except v_0 which is labeled $(a, 1)$. It is easy to see that for all $i \geq 1$, we have $t'_n \models \phi_x(v_i, w_i)$ and for all $j > i$, we have $t'_n \not\models \phi_x(v_j, w_i$. Hence there are at least n equivalence classes for the relation defined by ψ. So Algorithm 1 breaks.
5. $F = y(x, z)$. Similarly to the previous case, we fix a variable. Let $\psi(z, z') = \forall x, \phi_y(x, z) \leftrightarrow \phi_y(x, z')$. Let t'_n be the tree defined in the previous case. Hence y is fixed to denote the node v_0. We can prove that for all $i > 0$, we have $t'_n \models \phi_y(v_i, w_i)$ but for all $j > i$, we have $t'_n \not\models \phi_y(v_i, w_j)$. Hence there are at least n equivalence classes for the relation defined by ψ. So Algorithm 1 breaks.
6. $F = z(x, y)$. This case is symmetric to the previous one. □

Further Extensions

First note that all the results presented in the paper also hold for FO-queries, as we do not use second order variables in decompositions (but in this case we need to add in the tree structure a total order on the nodes).
We would like to investigate independence problems for more general classes of structures C. Indeed, we can give two sufficient conditions for relative independence to be decidable on C: (i) boundedness of an MSO formula with first-order variables is decidable on C, (ii) there is a computable MSO-definable total order on the elements of the structures of C. The first point has already been detailed in Section 3, while the second point is studied in [9]. This is the case for instance for unranked tree structures, over the signature consisting of the first-child and next-sibling predicates, and predicates to test the labels.

Finally, we would like to extend independence w.r.t. a dependence forest to independence w.r.t. a dependence graph. The techniques presented here do not seem to be easily extendable to graphs, even for a clique of size 3 for instance. In particular, we cannot use an inductive proof based on Lemma 4 anymore.

References

1. A. Berlea. On-the-fly tuple selection for XQuery. In *International Workshop on XQuery Implementation, Experience and Perspectives*, June 2007.
2. M. Bojanczyk. A bounding quantifier. In *14th Annual Conference of the EACSL on Computer Science Logic*, 2004.
3. J. Chomicki, D. Goldin, G. Kuper, and D. Toman. Variable independence in constraint databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1422–1436, 2003.
4. H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 2007.
5. S. Cosmadakis, G. Kuper, and L. Libkin. On the orthographic dimension of definable sets. *Inf. Process. Lett.*, 79(3):141–145, 2001.
6. B. Courcelle. Structural properties of context-free sets of graphs generated by vertex replacement. *Inf. Comput.*, 116(2):275–293, 1995.
7. B. Courcelle. Linear delay enumeration and monadic second-order logic. 2007. To appear in Discrete Applied Mathematics.
8. B. Courcelle and S. Oum. Vertex-minors, monadic second-order logic, and a conjecture by seese. *J. Comb. Theory Ser. B*, 97(1):91–126, 2007.
9. Bruno Courcelle. The monadic second-order logic of graphs x: linear orderings. *Theor. Comput. Sci.*, 160(1-2):87–143, 1996.
10. E. Filiot, J. Niehren, J.-M. Talbot, and S. Tison. Polynomial time fragments of xpath with variables. In *ACM Symposium on Principles of Database Systems*, 2007.
11. S. Grumbach, P. Rigaux, and L. Segoufin. On the orthographic dimension of constraint databases. In *7th International Conference on Database Theory*, pages 199–216. Springer-Verlag, 1999.
12. L. Libkin. Variable independence for first-order definable constraints. *TOCL*, 4(4):431–451, 2003.
13. L. Libkin. Logics over unranked trees: an overview. *Logical Methods in Computer Science*, 3(2):1–31, 2006.
14. Holger Meuss, Klaus U. Schulz, and François Bry. Towards aggregated answers for semistructured data. In *ICDT '01: Proceedings of the 8th International Conference on Database Theory*, pages 346–360, 2001.
15. J. Niehren, L. Planque, J.-M. Talbot, and S. Tison. N-ary queries by tree automata. In *10th International Symposium on Database Programming Languages*, volume 3774, pages 217–231, 2005.
16. T. Schwentick. On diving in trees. In *25th International Symposium on Mathematical Foundations of Computer Science*, pages 660–669, 2000.
17. H. Seidl. Ambiguity, valuedness and costs. 1992. Habilitation Thesis.
18. H. Seidl. Equivalence of finite-valued tree transducers is decidable. *Math. Syst. Theory*, 27(4):285–346, 1994.
19. J. W. Thatcher and J. B. Wright. Generalized finite automata with an application to a decision problem of second-order logic. 2:57–82, 1968.