FO Queries Strongly Distributing over Components in Arbitrary Cardinality

Francesco Di Cosmo

dicosmo.francesco@gmail.com

Abstract. In previous works a coordination-free strategy to compute Datalog with negation queries over databases distributed over many computational nodes has been studied, providing a syntactic characterization and proving the undecidability of the problem of deciding whether a query distributes. In view of a recast for FO queries, we report about a work in progress, namely the study of FO queries strongly distributing over components. We prove that the decision problem of establishing whether a FO query strongly distributes over components is undecidable and highlight how some syntactic bonds typical of Datalog with negation, namely safeness, play a crucial role in specifying strongly distributive FO queries.

Keywords: Strong distribution · Undecidability · Safeness

1 Introduction

In [1] the problem of establishing which Datalog¬ queries distribute over components has been studied, i.e. to determine those queries q such that, for any database D, the following holds:

$$q(D) = \bigcup_{C \in cc(D)} q(C)$$

Here by $cc(D)$ we denote the set of connected components of D (formally introduced in sec. 2.1). Informally, for example, if the database is a set of family trees, then each tree is a *component*, where its memebers are *connected* by family relationships.

The aforementioned problem arises in the context of looking for a parallelized coordination-free query computation strategy over databases distributed across many computational nodes [2]. Indeed, we can interpret the right-hand term as the result of the following strategy:

1. Facts of the whole database are stored in a scattered fashion over many computational nodes. Hence, for each node there is a local database.
2. During a preliminary phase, nodes can communcate with each other to update local databases asking and getting all and only facts about individuals from the local database. In the end, each node will host some connected components of the global database, say one.

3. Each node computes the query over its local database, neglecting coordination with other nodes, getting a set of local answers.
4. All the sets of local answers are merged through the union set operator. This is the result of the computation.

By Datalog¬ we refer to a variant of standard Datalog¬ [3] whose queries are expressible through a program P and a goal g, such that:

- P is a set of rules like:
$$H \leftarrow B$$
where H, the head, is an asserted literal, over an intentional vocabulary, satisfying safeness, in the sense that its variables occurs also in B, the body, and B is a conjunction of literals, over the same vocabulary extended with an extensional one, also satisfying safeness, in the sense that every variable occurring in a negated literal in a rule occurs also in an asserted literal in the body of the same rule.
- g is a rule without head, like:
$$\bot \leftarrow B$$

In both program and goal, neither constants nor equality are allowed. We say that a specification (P, g) is connected if, for each body B in (P, g), the graph (V, E) is connected, where V is the set of variables occurring in B and $\{x, y\} \in E$ iff x and y occurs in the same asserted literal. For example the rule:

$$E'(x, w) \leftarrow E(x, y) \wedge E(z, w)$$

is not connected, but the following one is it:

$$E'(x, w) \leftarrow E(x, y) \wedge E(y, w)$$

Two results have been established, namely that:

1. A Datalog¬ query distributes over components iff it is specifiable by a connected specification;
2. The problem of deciding whether, given a Datalog¬ query q, q distributes over components is undecidable.

These results are proved exploiting recursive specifications. With a view to recasting these results in absence of recursion, in this paper we report on a work in progress, namely the study of first order (FO) queries distributing over components.[1] Moreover, in this preliminary discussion, we abstract databases with purely relational structures of arbitrary cardinality and consider a stronger version of distribution over components, named strong distribution over components, i.e. those queries such that, for any structure S:

$$q(S) = \bigcup_{S' \subset S} q(S') = \bigcup_{C \in cc(S)} q(C)$$

The questions we want to answer are:

[1] We consider FO queries because, except for some details, they have the same expressive power as Datalog¬ without recursion (see Codd's theorems [3]).

1. Is it possible to characterize FO queries strongly distributing over components by syntactics means as in [1]?
2. Is the problem of deciding whether a FO query strongly distributes over components decidable?

2 Preliminaries

2.1 Structures and connected components

Let $\mathcal{L}$ be a relational vocabulary, i.e. a finite non empty set of relation symbols R/n, with positive arity $n > 0$, and constant symbols c.[2] A structure S over $\mathcal{L}$ is a not empty set $Dom(S)$, called the domain of S, enriched by interpretations $R^S \subset Dom(S)^n$, for each relation symbol $R/n \in \mathcal{L}$, and $c^S \in Dom(S)$, for each constant symbol $c \in \mathcal{L}$.

We say that a structure S' is a substructure of S (S is a superstructure of S'), $S' \subset S$, iff:

1. $Dom(S') \subset Dom(S)$;
2. for each relation symbol $R/n \in \mathcal{L}$, $R^{S'} = R^S \cap Dom(S')^n$;
3. for each constant symbol $c \in \mathcal{L}$, $c^{S'} = c^S$.[3]

Given a subset $\mathscr{A} \subset Dom(S)$ such that, for each constant symbol $c \in \mathcal{L}$, $c^S \in \mathscr{A}$, $\mathscr{A}$ generates a substructure A of S, the only one such that $Dom(A) = \mathscr{A}$.

The underlying graph of a structure S over L is the graph (V, E), where $V = Dom(S)$ and $\{a, b\} \in E$ iff there is a relation symbol $R/n \in \mathcal{L}$ and a n-tuple τ such that $a, b \in \tau \in R^S$. A connected component of S is a substructure generated by a connected component of its underlying graph. The set of connected components of S is denoted $cc(S)$. If S admits only one connected component, then we say that S is connected.

Directed graphs are simple examples of relational structures. For instance the graph with vertex set $V = \{v_1, v_2, v_3\}$ and edge set $E = \{(v_1, v_2), (v_3, v_3)\}$ can be considered as a structure S over the purely relational language $\mathscr{L} = \{F/2\}$, where $Dom(S) = V$ and $F^S = E$. A substructure of S could be S' with $Dom(S') = \{v_1, v_3\}$ and $F^{S'} = \{(v_3, v_3)\}$, while the connected components of S are the structures $(\{v_1, v_2\}, \{(v_1, v_2)\})$ and $(\{v_3, v_3\}, \{(v_3, v_3)\})$.

2.2 Preservation theorems

A FO formula $\varphi(x_1, \ldots, x_n)$ over a vocabulary $\mathcal{L}$ is preserved under superstructures iff, for any two structures $S' \subset S$ over $\mathcal{L}$:

$$\forall a_1, \ldots, a_n \in Dom(S') \quad S' \models \varphi(a_1, \ldots, a_n) \Rightarrow S \models \varphi(a_1, \ldots, a_n)$$

[2] Note that no function symbols are allowed.
[3] Therefore, $c^S \in Dom(S')$ holds.

whereas, φ is preserved under substructures iff the vice versa is valid, i.e.:

$$\forall a_1, \ldots, a_n \in Dom(S') \quad S \models \varphi(a_1, \ldots, a_n) \Rightarrow S' \models \varphi(a_1, \ldots, a_n)$$

The following theorems [4] hold also for vocabularies involving function symbols.

Theorem 1 (Preservation theorems).

1. *A formula φ is preserved under superstructures iff it is equivalent to a formula in Σ_1, the set of prenex existential formulas.*
2. *A formula φ is preserved under substructures iff it is equivalent to a formula in Π_1, the set of prenex universal formulas.*

For example, $\exists x\, x = x$ is a valid Σ_1 sentence.[4] Since it is valid, it is preserved under substructures and, by preservation theorem, it must be equivalent to a (valid) Π_1 sentence, like $\forall x\, x = x$. Finally, note that a quantifier-free formula is both Σ_1 and Π_1.

2.3 FO queries

Let $\mathcal{L}$ be a purely relational vocabulary, i.e. a relational vocabulary also without constant symbols. A FO specification over $\mathcal{L}$ is a couple (φ, Ξ), where φ is a FO formula over $\mathcal{L}$ and Ξ is a sequence of all elements in $Var(\varphi)$,[5] the set of free variables in φ. A FO specification $(\varphi(x_1, \ldots, x_n), \Xi)$ over $\mathcal{L}$ specify the FO query $q_{(\varphi,\Xi)}$ such that, for any structure S over $\mathcal{L}$:

$$q(S) = \{(h(x))_{x \in \Xi} \,|S \models \varphi(h(x_1), \ldots, h(x_n))\}$$

Given a FO query q we say that:

1. q is monotonic iff, for any two structures $S' \subset S$:

$$q(S') \subset q(S)$$

2. q is local iff, for any structure S and $a \in q(S)$:

$$\exists C \in cc(S) \quad a \in q(C)$$

It is straightforward to prove that a FO query strongly distributes (as defined in sec. 1) iff it is both monotonic and local. Hence, we will split the study of strongly distributive queries in the study of monotonicity and locality. Lastly, it will be useful the notion of closure by constants:

Definition 1. *Given a FO formula $\varphi(x_1, \ldots, x_n)$ over a vocabulary $\mathcal{L}$, let $\mathcal{L}'$ be the extension of $\mathcal{L}$ with n new constant symbols $c_1, \ldots, c_n$. The closure by constants φ_c of φ is the sentence over $\mathcal{L}'$ obtained replacing in φ each free variables x_i with the constant c_i, for any $i \in \{1, \ldots, n\}$.*

[4] It is valid because the domain of a structure cannot be empty.

[5] Eventually with repetitions.

3 Monotonicity

We now characterize monotonic FO queries as those queries specifiable by a Σ_1 formula.

Proposition 1. *Let $\varphi(x_1, \ldots, x_n)$ be a FO formula over a purely relational vocabulary $\mathcal{L}$. A FO query $q_{(\varphi,\Xi)}$ is monotonic iff φ is preserved under superstructures.*

Proof.

$\Rightarrow$: Let $S' \subset S$ and $a_1, \ldots, a_n \in Dom(S')$ such that $S' \models \varphi(a_1, \ldots, a_n)$. Hence, there is at least a valuation h such that $h(x_i) = a_i$ for any $i \in \{1, \ldots, n\}$, $(h(x))_{x\in\Xi} \in q(S')$ and, by hypothesis, $(h(x))_{x\in\Xi} \in q(S)$. So $S \models \varphi(a_1, \ldots, a_n)$.
$\Leftarrow$: Similar to previous.

Applying the preservation theorem 1, we obtain the following theorem.

Theorem 2. *A FO query $q_{(\varphi,\Xi)}$ is monotonic iff φ (φ_c) is equivalent to a Σ_1 formula (sentence).*

Therefore, to check whether a FO formula $q_{(\varphi,\Xi)}$ is monotonic amounts to check if the closure by constants φ_c is equivalent to a Σ_1 sentence. We now prove that this last check is not algorithmically possible.

Definition 2. *Let F_1, F_2 be two fragments of FO sentences over a vocabulary $\mathcal{L}$. With $Eq(F_1, F_2)$ we denote the decision problem of establishing whether, given a sentence $\varphi_1 \in F_1$,*

$$\exists \varphi_2 \in F_2 \quad \varphi_1 \leftrightarrow \varphi_2$$

Lemma 1. *Let F_1, F_2 be two fragments of FO over a vocabulary $\mathcal{L}$ such that:*

1. *$SAT(F_1)$, the decision problem of satisfiability of a sentence in F_1, is undecidable;*
2. *$SAT(F_2)$ is decidable;*
3. *F_1 contains all contradictory sentences.*

Then, $Eq(F_1, F_2)$ is undecidable.

Proof. If $Eq(F_1, F_2)$ were decidable through an algorithm A, then, given a sentence $\varphi \in F_1$ as input to A,

- If the output is negative, then φ is not contradictory, namely satisfiable;
- If the output is positive, then there is at least one $\varphi' \in F_2$ such that $\varphi \leftrightarrow \varphi'$ is valid and, by completeness of FO, also derivable. Since the set of derivable sentences is recursively enumerable, it is possible to algorithmically build up at least one such φ'. Recalling that $SAT(F_2)$ is decidable, it is possible to decide if φ', so φ, is satisfiable.

In both cases it would be possible to decide whether φ is satisfiable and $SAT(F_1)$ would be decidable, which contradicts the hypothesis 1.

Theorem 3. *Let $\mathcal{L}$ be a relational vocabulary sufficiently expressive, i.e. with at least one relational symbol R/n with $n \geq 2$. Let also σ_1 and π_1 be, respectively, the set of existential sentences and universal sentences over $\mathcal{L}$. Denoting with $\overline{FO}$ the set of all FO sentences over $\mathcal{L}$, $Eq(\overline{FO}, \sigma_1)$ and $Eq(\overline{FO}, \pi_1)$ are both undecidable.*

Proof. It is well-known that the Bernays-Schönfinkel-Ramsey fragment (BSR), i.e. the set of FO prenex sentences (without function symbols)[6] and with a prefix like $\exists^*\forall^*$, is such that $SAT(BSR)$ is decidable [5]. Clearly, BSR contains both σ_1 and π_1 and they both contain all contradictory sentences.[7] Since $\mathcal{L}$ is sufficiently expressive, $SAT(FO)$ is undecidable [6]. Thereby, we can apply the previous lemma and obtain the thesis.

We can summarize previous results in the following corollary:

Corollary 1. *The decision problem of establishing whether, given a FO query $q_{(\varphi,\Xi)}$ over a sufficiently expressive vocabulary, $q_{(\varphi,\Xi)}$ is monotonic is undecidable.*

Since any contradictory formula specifies a local FO query,[8] the same argument used in lemma 1 can be reused for the decision problem of establishing whether a FO formula φ specifies a strongly distributive query, i.e. if φ is such that:

1. φ specifies a local FO query;
2. φ_c is equivalent to a Σ_1 sentence.

So, we can conclude also the following corollary:

Corollary 2. *The decision problem of establishing wheter, given a FO query $q_{(\varphi,\Xi)}$ over a sufficiently expressive vocabulary, $q_{(\varphi,\Xi)}$ strongly distributes is undecidable.*

4 Locality

Due to the previous section, we focus only on Σ_1 formulas. Moreover, here we consider only quantifier-free disjunctive normal form (DNF) formulas without $=$.[9] However, we report only about two syntactic bonds over conjunctions and

[6] Recall that $\mathcal{L}$ is a purely relational vocabulary, hence function simbols would not occour anyway.

[7] All contradictions are equivalent and $\exists x \quad x \neq x$ and $\forall x \quad x \neq x$ are two of them, the first in σ_1 and the second in π_1.

[8] Because, for any structures S, $q(S) = \emptyset$ holds.

[9] Anyway, we are confident that adding equality would not change the core ideas of what follows, but would only require some more technicality, like an equality propagation procedure as the union-find algorithm in [7]. Yet, we do not prove it here.

disjunctions necessary to admit locality, referable to safeness of Datalog¬ rules through the process of rectification and unfolding [3].[10]

We say that a formula $\varphi(x_1, \ldots, x_n)$ is local iff it specifies only local queries, i.e. iff, for any structure S and any $a_1, \ldots, a_n \in Dom(S)$:[11]

$$S \models \varphi(a_1, \ldots, a_n) \Rightarrow \exists C \in cc(S) \quad C \models \varphi(a_1, \ldots, a_n)$$

4.1 Conjunctions

Since a contradictory conjunction of literals is local, we will consider only satisfiable formulas. First, we focus on negative conjunctions, i.e. where all literals are negated, then, we take into account the remaining.

Theorem 4. *Let $\mathcal{L}$ be a purely relational vocabulary and $\varphi(x_1, \ldots, x_n)$ be a satisfiable negative conjunction of literals over $\mathcal{L}$. Then φ is local iff $n = 1$.*

Proof. Since φ is satisfiable, it is not possible that in φ occur both an asserted literal and its negation.

$\Rightarrow$: Let S be a structure such that:
- $Dom(S) = \{a_1, \ldots, a_n\}$, where $a_i \neq a_j$ if $i \neq j$;
- for any relation symbol $R/m \in \mathcal{L}$, $S^R = \emptyset$.[12]

Therefore, each $a \in Dom(S)$ forms a connected component and, since φ is negative:

$$S \models \varphi(a_1, \ldots, a_n)$$

Since φ is local, $(a_1, \ldots, a_n)$ must lay on a single connected component. This is possible only if $n = 1$.

$\Leftarrow$: By hypothesis, φ is of the form $\varphi(x)$. Then, let S be a structure and $a \in Dom(S)$ such that:

$$S \models \varphi(a)$$

Clearly, $\exists C \in cc(S)$ such that $a \in Dom(C)$ and, by preservation theorem 2, also:[13]

$$C \models \varphi(a)$$

By arbitrariness of S and a, φ is local.

Theorem 5. *Let $\mathcal{L}$ be a purely relational vocabulary and $\varphi(x_1, \ldots, x_n)$ a not negative satisfiable conjunction of literals over $\mathcal{L}$. If φ is local, then φ is safe, i.e. any variable occurring in a negated literal ocours also in an asserted literal.*

[10] Rectification and unfolding are those processes that allow to translate Datalog¬ without recursion in FO.

[11] Clearly, it follows also that $a_1, \ldots, a_n \in C$.

[12] I.e. S can be considered a plain set.

[13] Each quantifier-free formula is also a Π_1 formula.

Proof. Let $\varphi(x_1, \ldots, x_n)$ be a not negative satisfiable conjunction of literals $\bigwedge_{i \in I} L_i$, where $x \in \mathit{Var}(\varphi)$ occurs only in negated literals. Say that x is x_1. Since φ is not contradictory, then there is a structure S and $a_1, \ldots, a_n \in \mathit{Dom}(S)$ such that: $S \models \varphi(a_1, \ldots, a_n)$. Now consider the structure S', obtained from S adding a new element a to $\mathit{Dom}(S)$. Therefore, a does not take part in the interpretative part of S' and, clearly:

$$S' \models \varphi(a, a_2, \ldots, a_n)$$

Since $\{a\}$ is the domain of a connected component of S', the sequence $(a, a_2, \ldots, a_n)$ does not lay on a single connected component and so φ is not local.

4.2 Disjunctions

Theorem 6. *Let $\mathcal{L}$ be a purely relational vocabulary and $\varphi(x_1, \ldots, x_n)$ a disjunction $\bigvee_{i \in I} \varphi_i$, where φ_i is a satisfiable Σ_1 formula over $\mathcal{L}$ for each $i \in I$. If φ is local, then φ is regular, i.e. all disjuncts share the same set of free variables.*

Proof. Let $\varphi(x_1, \ldots, x_n)$ be a satisfiable disjunction $\bigvee_{i \in I} \varphi_i$, where $\varphi_i \in \Sigma_1$ for each $i \in I$. Suppose there are indices $i, j \in I$ such that $\mathit{Var}(\varphi_i) \neq \mathit{Var}(\varphi_j)$, say $x \in \mathit{Var}(\varphi_i) \setminus \mathit{Var}(\varphi_j)$ and suppose that x is x_1. Since $\varphi_j(y_1, \ldots, y_m)$ is satisfiable, there is a structure S and $a_1, \ldots, a_m \in \mathit{Dom}(S)$ such that:

$$S \models \varphi_j(a_1, \ldots, a_m)$$

Let h be a valuation such that $h(y_i) = a_i$ for each $i \in \{1, \ldots, m\}$. Now, as in the previous proof, consider a structure S', obtained from S adding an element a to $\mathit{Dom}(S)$. Then, the valuation k, obtained from h putting $k(x) = a$, still satisfies φ_j in S' (because $x \notin \mathit{Var}(\varphi_j)$). So, by semantic of disjunction:

$$S' \models \varphi(k(x_1), \ldots, k(x_n))$$

Since $\{k(x_1)\} = \{a\}$ is the domain of a connected component of S', $(k(x_1), \ldots, k(x_n))$ does not lay on a single connected component and so φ is not local.

5 Conclusions and further work

We have proved that, for sufficiently expressive vocabularies, the decision problem of establishing if a FO query strongly distributes over components is undecidable. This has been possible through a contrast with the *Entscheidungsproblem* of satisfiability of *FO* formulas. However, we remark that we considered FO in its full expressive power, ignoring those syntactical bonds stemming from rectification and unfolding, i.e. reflecting Datalog¬ safeness. Would something change if those bonds were considered? In fact, tackling the problem of locality, we showed that those bonds, in the form of safe conjunctions and regular disjunctions, are necessary conditions to allow locality of DNF quantifier-free formulas. This preliminary result should be extended to a full classification of FO formulas specifying local FO queries.

References

1. Ameloot, T.J., Ketsman, B., Neven, F., Zinn, D.: Datalog queries distributing over components. ACM TOCL **18**(1), 1–35 (2017)
2. Ameloot, T.J., Ketsman, B., Neven, F., Zinn, D.: Weaker forms on monotonicity for declarative networking: a more fine-grained answer to the CALM-conjecture. ACM TODS **40**(4), 1–45 (2016)
3. Ullman, J.D.: Principles of Database and Knowledge-Base Systems, Volume I. Computer Science Press, Rockville, Maryland (1989)
4. Chang, C.C., Keisler, H.: Model theory. Elsevier (1990)
5. Ramsey, F.P..: On a problem of formal logic. In: Gessel, I., Rota, G., CLASSIC PAPERS IN COMBINATORICS 2009, pp. 1–24. Birkhäuser Boston (Springer) (2009)
6. Börger, E., Grädel, E., Gurevich, Y.: The classical decision problem.2nd edn. Springer Verlag (Springer Science & Business Media) (2001)
7. Aho, A.V., Hopcroft, J. E.: The design and analysis of computer algorithms. Addison-Wesley Longman Publishing Company, Inc., Boston, MA, USA (1974)