FO queries strongly distributing over components

F. Di Cosmo

CILC
Trieste, 20 June 2019
Connected Components

\[ D = \{ P(a, b), R(b, b), P(b, c), P(d, e) \} \]

\[ C_1 = \{ P(a, b), R(b, b), P(b, c) \} \]

\[ C_2 = \{ P(d, e) \} \]

\[ cc(D) = \{ C_1, C_2 \} \]

Definition

Two facts are directly connected, \( \sim_d \), if they share some individual. The connection relation, \( \sim \), is the transitive closure of \( \sim_d \).

The set of connected components of \( D \), \( cc(D) \), is the set of equivalence classes of \( \sim \).

F. Di Cosmo

FO queries strongly distributing over components
Connected Components

\[ D = \{ P(a, b), \]
\[ R(b, b), \]
\[ P(b, c), \]
\[ P(d, e) \} \]

Definition

Two facts are directly connected, \( \sim_d \), if they share some individual. The connection relation, \( \sim \), is the transitive closure of \( \sim_d \). The set of connected components of \( D \), \( cc(D) \), is the set of equivalence classes of \( \sim \).
Connected Components

\[ D = \{ P(a, b), \]
\[ R(b, b), \]
\[ P(b, c), \]
\[ P(d, e) \} \]

\[ C_1 = \{ P(a, b), R(b, b), P(b, c) \} \]
\[ C_2 = \{ P(d, e) \} \]

\[ cc(D) = \{ C_1, C_2 \} \]

Definition

Two facts are directly connected, \( \sim \), if they share some individual. The connection relation, \( \sim \), is the transitive closure of \( \sim \).

The set of connected components of \( D \), \( cc(D) \), is the set of equivalence classes of \( \sim \).
Connected Components

\[ D = \{ P(a, b), \]
\[ R(b, b), \]
\[ P(b, c), \]
\[ P(d, e) \} \]

\[ C_1 = \{ P(a, b), R(b, b), P(b, c) \} \]
\[ C_2 = \{ P(d, e) \} \]
\[ cc(D) = \{ C_1, C_2 \} \]

Two facts are directly connected, \( \sim_d \), if they share some individual. The connection relation, \( \sim \), is the transitive closure of \( \sim_d \).

FO queries strongly distributing over components.
Connected Components

\[ D = \{ P(a, b), \]
\[ R(b, b), \]
\[ P(b, c), \]
\[ P(d, e) \}\]

\[ C_1 = \{ P(a, b), R(b, b), P(b, c) \} \quad C_2 = \{ P(d, e) \} \]

\[ cc(D) = \{ C_1, C_2 \} \]

**Definition**

Two facts are **directly connected**, \( \sim_d \), if they share some individual.

The **connection relation**, \( \sim \), is the transitive closure of \( \sim_d \).

The **set of connected components** of \( D \), \( cc(D) \), is the set of equivalence classes of \( \sim \).
Distributed strategy

1) Each node hosts a local database;
2) Local databases are closed by connectivity: each node hosts some connected components, say one;
3) The query is computed on each node;
4) The answer sets are merged, obtaining $\bigcup_{C \in cc(D)} q(C)$.

Definition
A query $q$ distributes if, for each database $D$:
$$q(D) = \bigcup_{C \in cc(D)} q(C)$$
Distributed strategy

1) Each node hosts a local database;
1) Each node hosts a local database;
2) Local databases are closed by connectivity: each node hosts some connected components, say one;

Definition
A query $q$ distributes if, for each database $D$:
$$q(D) = \bigcup_{C \in cc(D)} q(C).$$
Distributed strategy

1) Each node hosts a local database;
2) Local databases are closed by connectivity: each node hosts some connected components, say one;
3) The query is computed on each node;

Definition

A query $q$ distributes if, for each database $D$:

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

F. Di Cosmo

FO queries strongly distributing over components
Distributed strategy

1) Each node hosts a local database;
2) Local databases are closed by connectivity: each node hosts some connected components, say one;
3) The query is computed on each node;
4) The answers sets are merged, obtaining $\bigcup_{C \in C(C(D))} q(C)$. 
Distributed strategy

1) Each node hosts a local database;
2) Local databases are closed by connectivity: each node hosts some connected components, say one;
3) The query is computed on each node;
4) The answers sets are merged, obtaining \( \bigcup_{C \in CC(D)} q(C) \).

Definition

A query \( q \) **distributes** if, for each database \( D \):

\[
q(D) = \bigcup_{C \in CC(D)} q(C)
\]
Main question

Which queries distribute?

Results for Datalog

\[ \neg (Ameloot \ 2017), \text{exploiting recursion:} \]

Syntactic Characterization

A query distributes iff it has a connected specification.

\[ E(x, z) \leftarrow R(x, y), P(z, w), \neg R(x, z) \text{ is not connected.} \]

\[ E(x, z) \leftarrow R(x, y), P(y, z), R(x, z) \text{ is connected.} \]

Decision Problem

To establish whether a given query distributes is undecidable.

What if recursion is omitted?

F. Di Cosmo

FO queries strongly distributing over components
Main question

Which queries distribute?

Results for Datalog

¬ (Ameloot 2017), exploiting recursion:

Syntactic Characterization

A query distributes iff it has a connected specification.

\[ E(x, z) \leftarrow R(x, y), P(z, w), \neg R(x, z) \]

is not connected.

\[ E(x, z) \leftarrow R(x, y), P(y, z), R(x, z) \]

is connected.

Decision Problem

To establish whether a given query distributes is undecidable.

What if recursion is omitted?

F. Di Cosmo

FO queries strongly distributing over components
Main question

Which queries distribute?

Results for Datalog¬ (Ameloot 2017), exploiting recursion:

Syntactic Characterization

A query distributes iff it has a connected specification.

\[ E(x, z) \leftarrow R(x, y), P(z, w), \neg R(x, z) \]

is not connected.

\[ E(x, z) \leftarrow R(x, y), P(y, z), R(x, z) \]

is connected.

Decision Problem

To establish whether a given query distributes is undecidable.

What if recursion is omitted?

F. Di Cosmo

FO queries strongly distributing over components
Main question

Which queries distribute?

Results for Datalog\(\neg\) (Ameloot 2017), exploiting recursion:

**Syntactic Characterization**

A query distributes iff it has a connected specification.

\[
E(x, z) \leftarrow R(x, y), P(z, w), \neg R(x, z) \text{ is not connected.}
\]

\[
E(x, z) \leftarrow R(x, y), P(y, z), R(x, z) \text{ is connected.}
\]
Main question

Which queries distribute?

Results for Datalog\(\neg\) (Ameloot 2017), exploiting recursion:

**Syntactic Characterization**

A query distributes iff it has a connected specification.

\[
E(x, z) \leftarrow R(x, y), P(z, w), \neg R(x, z) \text{ is not connected.}
\]

\[
E(x, z) \leftarrow R(x, y), P(y, z), \quad R(x, z) \text{ is connected.}
\]

**Decision Problem**

To establish whether a given query distributes is undecidable.
Main question

Which queries distribute?

Results for Datalog\neg (Ameloot 2017), exploiting recursion:

Syntactic Characterization

A query distributes iff it has a connected specification.

\[ E(x, z) \leftarrow R(x, y), P(z, w), \neg R(x, z) \text{ is not connected.} \]

\[ E(x, z) \leftarrow R(x, y), P(y, z), \quad R(x, z) \text{ is connected.} \]

Decision Problem

To establish whether a given query distributes is undecidable.

What if recursion is omitted?
NR Datalog $\rightarrow$ FO relational queries.

Given $\phi(x)$ and $S$,

$q(\phi(S)) = \{a \mid S|_a = \phi(a)\}$

Structures:

Databases $\rightarrow$ FO relational structures.

$L = \{P_2/2, R_2/2\}$

$S = (\text{dom}(S), P_S, R_S)$

$\text{dom}(S) = \{a, b, c\}$,

$P_S = \{(a, b), (b, c), (d, e)\}$,

$R_S = \{(d, e)\}$

Problem:

Distribution $\rightarrow$ Strong distribution.

$q(S) = \bigcup S' \subseteq S \ q(S') = \bigcup C \in \text{cc}(S) \ q(C)$

• Monotony: $S' \subseteq S \Rightarrow q(S') \subseteq q(S)$

• Locality: $a \in q(S) \Rightarrow \exists C \in \text{cc}(S) \ a \in q(C)$
Language: NR Datalog \( \iff \) FO relational queries.

Given \( \varphi(x) \) and \( S \), \( q_\varphi(S) = \{ a | S \models \varphi(a) \} \)
Preliminaries

Language: NR Datalog $\iff$ FO relational queries.

Given $\varphi(x)$ and $S$, $q_{\varphi}(S) = \{a | S \models \varphi(a)\}$

Structures: Databases $\iff$ FO relational structures.

$L = \{P/2, R/2\} \quad S = (\text{dom}(S), P^S, R^S)$

$\text{dom}(S) = \{a, b, c\}, \quad P^S = \{(a, b), (b, c), (d, e)\}, \quad R^A = \{(d, e)\}$
Preliminaries

Language: NR Datalog \( \Rightarrow \) FO relational queries.

Given \( \varphi(x) \) and \( S \),

\( q_\varphi(S) = \{ a | S, a = \varphi(a) \} \)

Structures: Databases \( \Rightarrow \) FO relational structures.

\[ \mathcal{L} = \{ P/2, R/2 \} \quad S = (\text{dom}(S), P^S, R^S) \]

\( \text{dom}(S) = \{ a, b, c \}, P^S = \{ (a, b), (b, c), (d, e) \}, R^A = \{ (d, e) \} \)

Problem: Distribution \( \Rightarrow \) Strong distribution.

\[
q(S) = \bigcup_{S' \subset S} q(S') = \bigcup_{C \in \text{CC}(S)} q(C)
\]
Preliminaries

Language: NR Datalog $\iff$ FO relational queries.

Given $\varphi(x)$ and $S$, $q_{\varphi}(S) = \{a \mid S \models \varphi(a)\}$

Structures: Databases $\iff$ FO relational structures.

$L = \{P/2, R/2\} \quad S = (\text{dom}(S), P^S, R^S)$

$\text{dom}(S) = \{a, b, c\}, P^S = \{(a, b), (b, c), (d, e)\}, R^A = \{(d, e)\}$

Problem: Distribution $\iff$ Strong distribution.

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

- Monotony: $S' \subset S \Rightarrow q(S') \subset q(S)$
- Locality: $a \in q(S) \Rightarrow \exists C \in \text{cc}(S) \; a \in q(C)$
Monotony

Theorem (Łoś - Tarski)

An FO formula $\phi$ is preserved under superstructures ($\phi$ is monotone) iff $\phi$ is equivalent to a $\psi \in \Sigma^1$ (existential formulas).

Consider $\phi \equiv \psi \in \Sigma^1$?

- $\phi$ is satisfiable
  - No
  - Build $\psi$ via $|\phi \leftrightarrow \psi$?
  - Yes
  - SAT($\psi$)
- $\phi$ is not satisfiable
  - Yes
  - No
Theorem (Łoś - Tarski)

An FO formula $\varphi$ is preserved under superstructures ($q_\varphi$ is monotone) iff $\varphi$ is equivalent to a $\psi \in \Sigma_1$ (existential formulas).
Theorem (Łoś - Tarski)

An FO formula $\varphi$ is preserved under superstructures ($q_\varphi$ is monotone) iff $\varphi$ is equivalent to a $\psi \in \Sigma_1$ (existential formulas).

Consider $\varphi$
Monotony

Theorem (Łoś - Tarski)

An FO formula $\varphi$ is preserved under superstructures ($q_\varphi$ is monotone) iff $\varphi$ is equivalent to a $\psi \in \Sigma_1$ (existential formulas).

Consider $\varphi$

$\varphi \equiv \psi \in \Sigma_1$?
Theorem (Łoś - Tarski)

An FO formula \( \varphi \) is preserved under superstructures (\( q_\varphi \) is monotone) iff \( \varphi \) is equivalent to a \( \psi \in \Sigma_1 \) (existential formulas).

Consider \( \varphi \)

\[ \varphi \equiv \psi \in \Sigma_1 ? \]

\( \varphi \) is satisfiable
Theorem (Łoś - Tarski)

An FO formula $\varphi$ is preserved under superstructures ($q_{\varphi}$ is monotone) iff $\varphi$ is equivalent to a $\psi \in \Sigma_1$ (existential formulas).

Consider $\varphi$

$\varphi \equiv \psi \in \Sigma_1$?

No

Yes

$\varphi$ is satisfiable

Build $\psi$ via $\models \varphi \leftrightarrow \psi$
Theorem (Łoś - Tarski)

An FO formula \( \varphi \) is preserved under superstructures (\( q_\varphi \) is monotone) iff \( \varphi \) is equivalent to a \( \psi \in \Sigma_1 \) (existential formulas).

Consider \( \varphi \)

- No: \( \varphi \equiv \psi \in \Sigma_1 ? \)
- Yes: Build \( \psi \) via \( \models \varphi \leftrightarrow \psi \)

\( \varphi \) is satisfiable

\( SAT(\psi) ? \)
Theorem (Łoś - Tarski)

An FO formula $\varphi$ is preserved under superstructures ($q_\varphi$ is monotone) iff $\varphi$ is equivalent to a $\psi \in \Sigma_1$ (existential formulas).

Consider $\varphi$

- If $\varphi \equiv \psi \in \Sigma_1$?
  - If $\varphi$ is satisfiable:
    - Yes
    - $\varphi$ is not satisfiable
  - No: Build $\psi$ via $| = \varphi \leftrightarrow \psi$
    - If $SAT(\psi)$?
      - If $SAT(\psi)$: Yes
      - No

Theorem (Łoś - Tarski)

An FO formula $\varphi$ is preserved under superstructures ($q_\varphi$ is monotone) iff $\varphi$ is equivalent to a $\psi \in \Sigma_1$ (existential formulas).

Consider $\varphi$

- $\varphi \equiv \psi \in \Sigma_1$?
  - No: $\varphi$ is not satisfiable
  - Yes: Build $\psi$ via $\models \varphi \leftrightarrow \psi$

- $\varphi$ is satisfiable
  - Yes: $\text{SAT}(\psi)$?
  - No: $\varphi$ is satisfiable
Locality

Theorem

\[ A \land_{i \in I} L_i \text{ is local iff:} \]

- it is contradictory or
- all literals are negated and there is only one variable or
- each variable in a negated literal occurs also in an asserted one (safeness).

\[ a_1, \ldots, a_n - 1, a_n \]

\[ \varphi(x_1, \ldots, x_n) \]

\[ x_n \text{ only in } \neg A \]

S' . Di Cosmo

FO queries strongly distributing over components
Locality

No safeness... no locality.
No safeness... no locality.

Theorem

\( A \bigwedge_{i \in I} L_i \) is local iff:

- it is contradictory or
- all literals are negated and there is only one variable or
- each variable in a negated literal occurs also in an asserted one (safeness).
No safeness... no locality.

Theorem

$A \land_{i \in I} L_i$ is local iff:
- it is contradictory or
- all literals are negated and there is only one variable or
- each variable in a negated literal occurs also in an asserted one (safeness).

$\varphi(x_1, \ldots, x_n)$

$x_n$ only in $\neg A$

$\quad a_1, \ldots, a_{n-1}, a_n \quad b_n$

$S \quad \text{passive}$

$S'$

FO queries strongly distributing over components
Conclusions

• Undecidability.

• Safeness (locality).

• Strong distribution vs Simple distribution.

• Constraints on structures.
Ameelot, Tom J. et al.  
*Datalog queries distributing over components*  

Berger, Gerald and Pieris, Andreas  
*Ontology mediated queries distributing over components*  
IJCAI, 2016.