Optimal Broadcasting Strategies for Conjunctive Queries over Distributed Data

Bas Ketsman, Frank Neven

Hasselt University & transnational University of Limburg

Outline

1. Setting and Context
2. Oblivious Broadcasting Functions
3. Correctness & Optimality
4. Broadcast Dependency Sets
5. Conclusion & Future Work

Context

CALM conjecture: "Monotonic = No-coordination"

[Hellerstein, 2010]

Context

CALM conjecture: "Monotonic = No-coordination"

[Hellerstein, 2010]

- True

[Ameloot, Neven, Van den Bussche, 2011]

Context

CALM conjecture: “Monotonic = No-coordination”

[Hellerstein, 2010]

- True

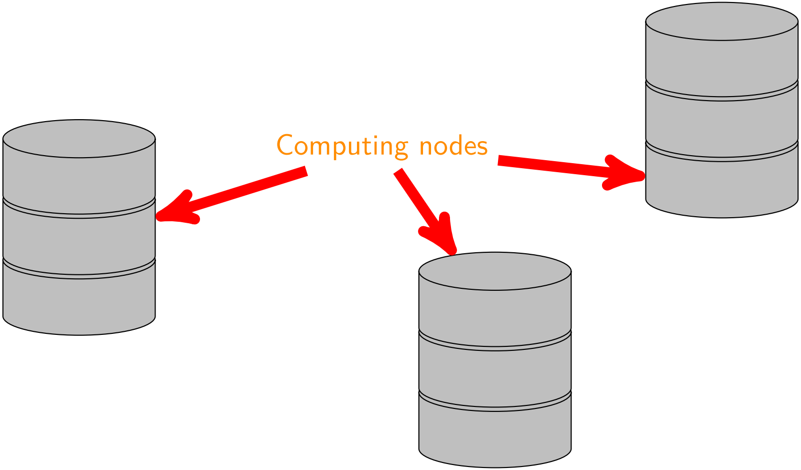
  [Ameloot, Neven, Van den Bussche, 2011]

- Generalization

  [Ameloot, Neven, K., Zinn, 2014]

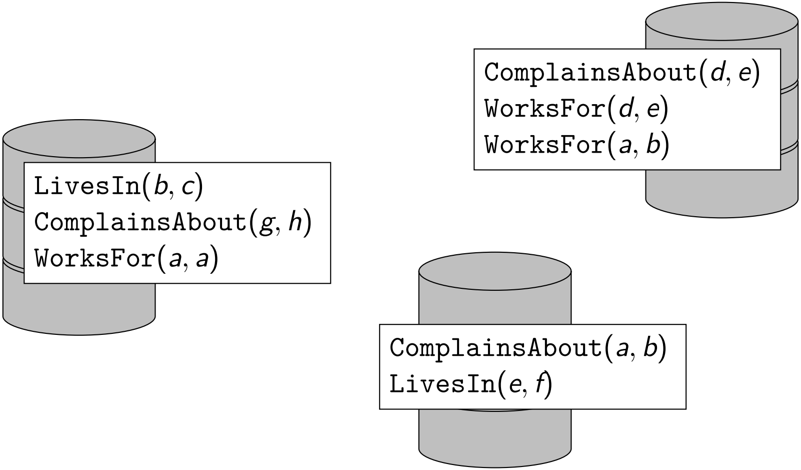
Setting

A network $\mathcal{N}$ is a set of computing nodes.



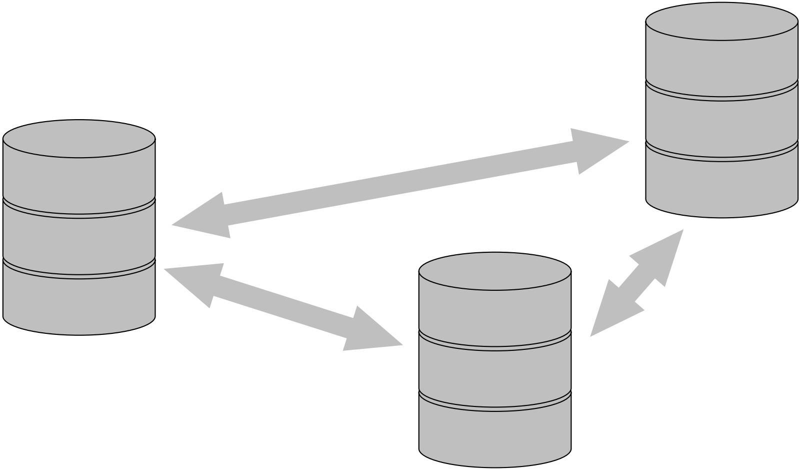
Setting

A distribution is a mapping from nodes onto instances.



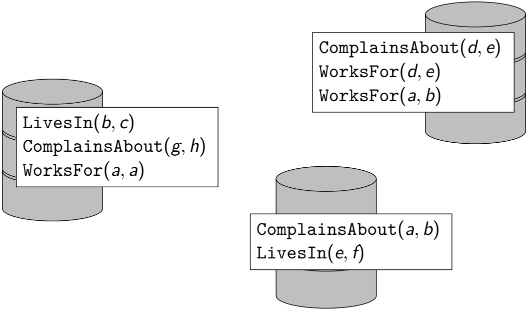
Setting

Communication: Asynchronous



Monotonic $\subseteq$ No-coordination

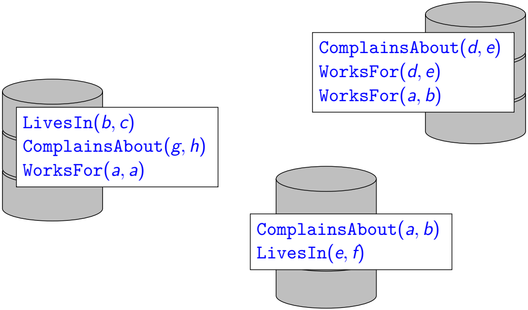
Running Example:
$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$



Monotonic $\subseteq$ No-coordination

Running Example:

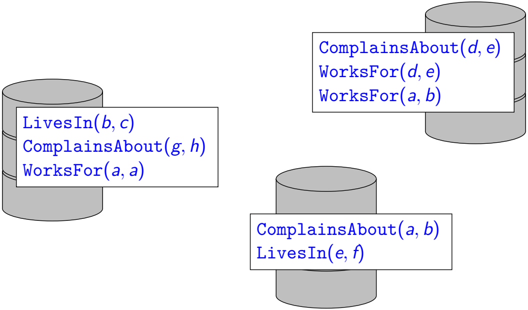
$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$



- Let every node broadcast all of its data;

Monotonic $\subseteq$ No-coordination

Running Example:
$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

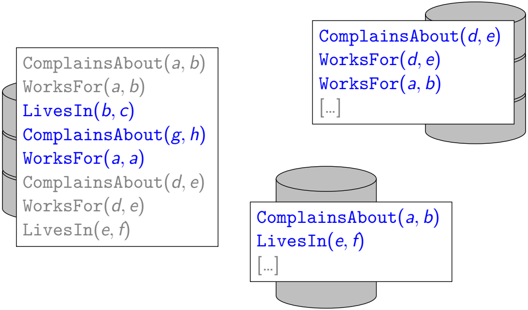


- Let every node broadcast all of its data;
- Periodically run $\mathcal{Q}$ locally on every node

Monotonic $\subseteq$ No-coordination

Running Example:

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

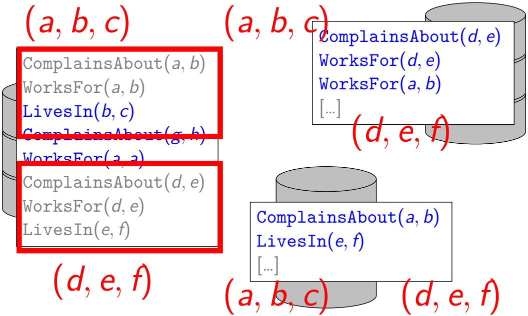


- Let every node broadcast all of its data;
- Periodically run $\mathcal{Q}$ locally on every node

Monotonic ⊆ No-coordination

Running Example:

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

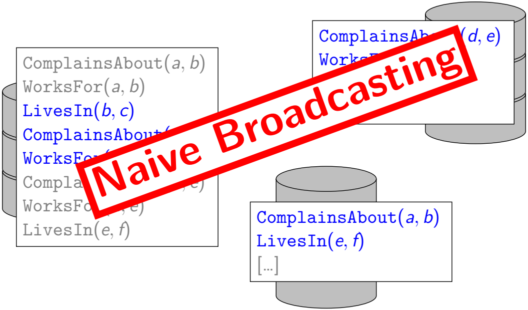


- Let every node broadcast all of its data;
- Periodically run $\mathcal{Q}$ locally on every node

Monotonic ⊆ No-coordination

Running Example:

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$



- Let every node broadcast all of its data;
- Periodically run $\mathcal{Q}$ locally on every node

Current Work

No-coordination + Broadcast all

⇩

No-coordination + Selective broadcasting

Current Work

No-coordination + Broadcast all

⇩

No-coordination + Selective broadcasting

Full CQs without self-joins

1. $Q(x, y, z) \leftarrow R(x, y), S(y, z)$
2. $Q(x, y) \leftarrow R(x, y), S(y, z)$
3. $Q(x, y) \leftarrow R(x, y), R(y, x)$

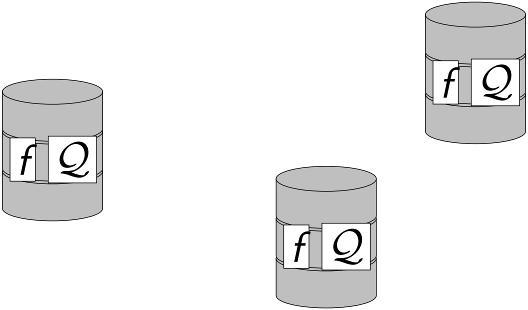
Outline

1. Setting and Context
2. Oblivious Broadcasting Functions
3. Correctness & Optimality
4. Broadcast Dependency Sets
5. Conclusion & Future Work

Oblivious Broadcasting Functions

Definition

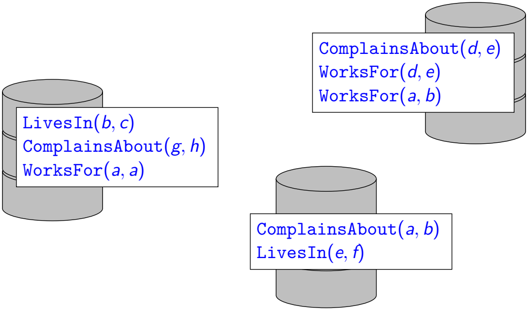
Let f be a total function from instances to instances. We call f an oblivious broadcasting function (OBF) if f is generic, and $f(I) \subseteq I$ for every instance I.



Running Example: Naive Broadcasting

$$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$$

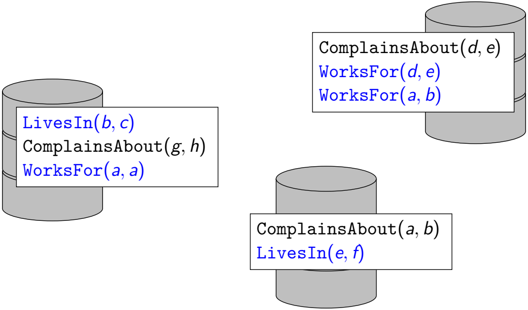
OBF: broadcast everything



Running Example: Relation-Based

$$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$$

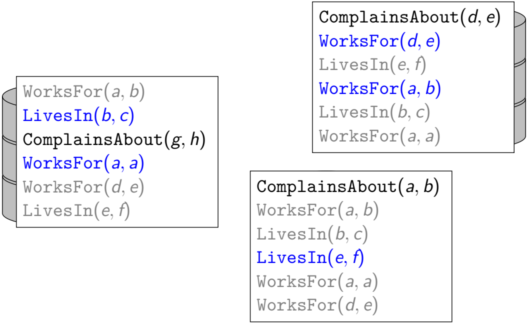
OBF: don't broadcast $\mathbf{ComplainsAbout}(x, y)$



Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

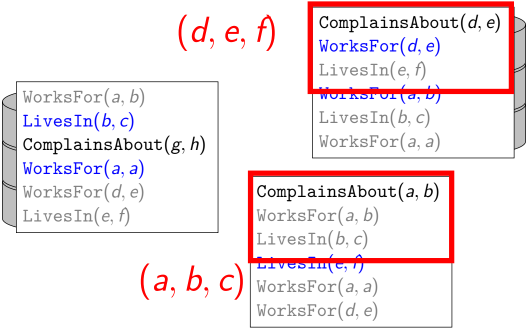
OBF: don't broadcast $\textbf{ComplainsAbout}(x, y)$



Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF: don't broadcast **$\texttt{ComplainsAbout}(x, y)$**



Oblivious Broadcasting Functions

Let $\mathcal{N}$ be a network, I an instance, H a distribution of I over $\mathcal{N}$.

Definition

Let f be a total function from instances to instances. We call f an oblivious broadcasting function (OBF) if f is generic, and $f(I) \subseteq I$ for every instance I.

Broadcast Facts

$$B(f, H) \stackrel{\text{def}}{=} \bigcup_{c \in \mathcal{N}} f(H(c)).$$

Oblivious Broadcasting Functions

Let $\mathcal{N}$ be a network, I an instance, H a distribution of I over $\mathcal{N}$.

Definition

Let f be a total function from instances to instances. We call f an oblivious broadcasting function (OBF) if f is generic, and $f(I) \subseteq I$ for every instance I.

Broadcast Facts

$$B(f, H) \stackrel{\text{def}}{=} \bigcup_{c \in \mathcal{N}} f(H(c)).$$

Distributed Output

$$eval(\mathcal{Q}, f, H) \stackrel{\text{def}}{=} \bigcup_{c \in \mathcal{N}} \mathcal{Q}(H(c) \cup B(f, H))$$

Outline

1. Setting and Context
2. Oblivious Broadcasting Functions
3. Correctness & Optimality
4. Broadcast Dependency Sets
5. Conclusion & Future Work

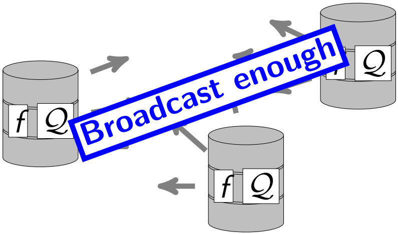
Correctness

Definition

An OBF is correct for CQ $\mathcal{Q}$ if $\mathcal{Q}(I) = eval(\mathcal{Q}, f, H)$ for every instance I and distribution H for I.

Correctness

Definition
An OBF is correct for CQ $\mathcal{Q}$ if $\mathcal{Q}(I) = eval(\mathcal{Q}, f, H)$ for every instance I and distribution H for I.



Correctness

Definition

Let $\mathcal{Q}$ be a CQ, $\boldsymbol{f}$ and $\boldsymbol{g}$ be two distinct facts. We say that $\boldsymbol{f}$ and $\boldsymbol{g}$ are compatible, written $\boldsymbol{f} \sim_{\mathcal{Q}} \boldsymbol{g}$, if there is a valuation V for $\mathcal{Q}$ that requires them both.

Correctness

Definition

Let $\mathcal{Q}$ be a CQ, $\boldsymbol{f}$ and $\boldsymbol{g}$ be two distinct facts. We say that $\boldsymbol{f}$ and $\boldsymbol{g}$ are compatible, written $\boldsymbol{f} \sim_{\mathcal{Q}} \boldsymbol{g}$, if there is a valuation V for $\mathcal{Q}$ that requires them both.

Example

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

- $\texttt{ComplainsAbout}(a, b) \sim_{\mathcal{Q}} \texttt{LivesIn}(b, c)$
- $\texttt{ComplainsAbout}(a, b) \not\sim_{\mathcal{Q}} \texttt{LivesIn}(a, c)$

Correctness

Definition

Let $\mathcal{Q}$ be a CQ, $\boldsymbol{f}$ and $\boldsymbol{g}$ be two distinct facts. We say that $\boldsymbol{f}$ and $\boldsymbol{g}$ are compatible, written $\boldsymbol{f} \sim_{\mathcal{Q}} \boldsymbol{g}$, if there is a valuation V for $\mathcal{Q}$ that requires them both.

Lemma

Let $\mathcal{Q}$ be a CQ and f be an OBF. Then, the following are equivalent:

1. f is correct for $\mathcal{Q}$; and
2. there are no instances I, J, and facts $\boldsymbol{f}$, $\boldsymbol{g}$, with $\boldsymbol{f} \sim_{\mathcal{Q}} \boldsymbol{g}$, $\boldsymbol{g} \notin I$, $\boldsymbol{f} \notin J$ such that $\boldsymbol{f} \notin f(I \cup \{\boldsymbol{f}\})$ and $\boldsymbol{g} \notin f(J \cup \{\boldsymbol{g}\})$.

Optimality

Ideally: "One OBF that is always at least as good as all others"

Optimality

Ideally: "One OBF that is always at least as good as all others"

$$||B(f, H)|| \stackrel{\text{def}}{=} \sum_{c \in \mathcal{N}} |f(H(c))|$$

Definition

An OBF f for a CQ $\mathcal{Q}$ is optimal if $||B(f, H)|| \leq ||B(g, H)||$ for every other OBF g for $\mathcal{Q}$ and for every instance I and distribution H.

Optimality

Ideally: "One OBF that is always at least as good as all others"

$$||B(f, H)|| \stackrel{\text{def}}{=} \sum_{c \in \mathcal{N}} |f(H(c))|$$

Definition

An OBF f for a CQ $\mathcal{Q}$ is optimal if $||B(f, H)|| \leq ||B(g, H)||$ for every other OBF g for $\mathcal{Q}$ and for every instance I and distribution H.

No such OBF exists

Proof: No Optimal OBF exists

- Arbitrary query: $\mathcal{Q}(\mathsf{x}) \leftarrow \mathtt{R}_1(\mathsf{y}_1), \ldots, \mathtt{R}_\mathtt{k}(\mathsf{y}_2)$ $(k \geq 2)$

Proof: No Optimal OBF exists

- Arbitrary query: $\mathcal{Q}(\mathsf{x}) \leftarrow \mathtt{R}_1(\mathsf{y}_1), \ldots, \mathtt{R}_\mathtt{k}(\mathsf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$

Proof: No Optimal OBF exists

- Arbitrary query: $\mathcal{Q}(\mathsf{x}) \leftarrow \mathtt{R}_1(\mathsf{y}_1), \ldots, \mathtt{R}_\mathtt{k}(\mathsf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$
- Arbitrary valuation

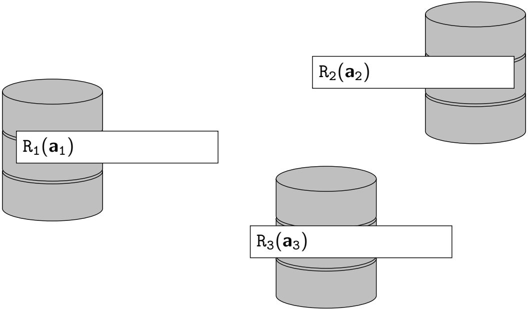
$\mathtt{R}_2(\mathbf{a}_2)$

$\mathtt{R}_1(\mathbf{a}_1)$

$\mathtt{R}_2(\mathbf{a}_3)$

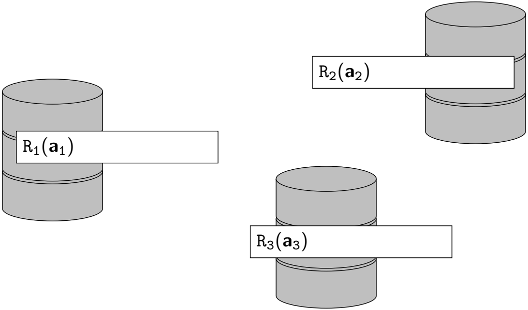
Proof: No Optimal OBF exists

- Arbitrary query: $\mathcal{Q}(\mathbf{x}) \leftarrow \mathtt{R}_1(\mathbf{y}_1), \ldots, \mathtt{R}_k(\mathbf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$
- Arbitrary valuation



Proof: No Optimal OBF exists

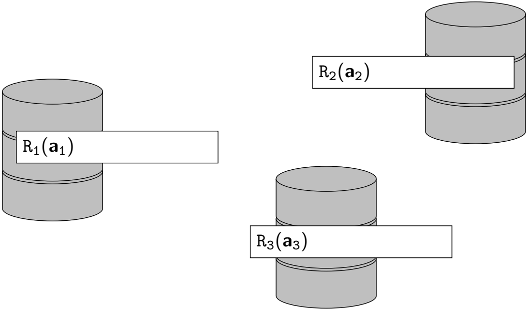
- Arbitrary query: $\mathcal{Q}(\mathsf{x}) \leftarrow \mathtt{R}_1(\mathsf{y}_1), \ldots, \mathtt{R}_k(\mathsf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$
- Arbitrary valuation



- At least two of these facts **must** be broadcast

Proof: No Optimal OBF exists

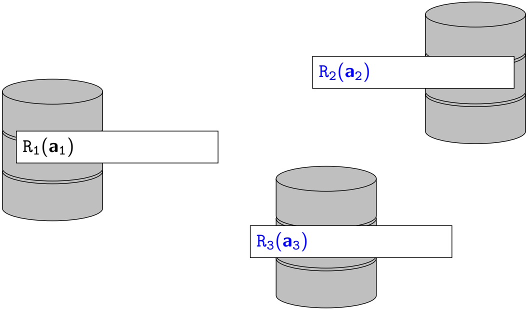
- Arbitrary query: $\mathcal{Q}(\mathsf{x}) \leftarrow \mathtt{R}_1(\mathsf{y}_1), \ldots, \mathtt{R}_k(\mathsf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$
- Arbitrary valuation



- At least two of these facts **must** be broadcast
- OBFs exist that broadcast **only** two of them

Proof: No Optimal OBF exists

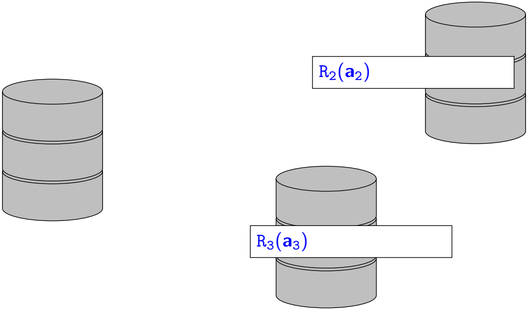
- Arbitrary query: $\mathcal{Q}(\mathbf{x}) \leftarrow \mathtt{R}_1(\mathbf{y}_1), \ldots, \mathtt{R}_k(\mathbf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$
- Arbitrary valuation



W.l.o.g: OBF does not broadcast $\mathtt{R}_1(\mathtt{a}_1)$

Proof: No Optimal OBF exists

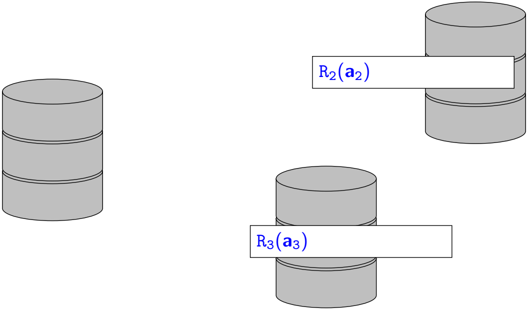
- Arbitrary query: $\mathcal{Q}(\mathbf{x}) \leftarrow \mathtt{R}_1(\mathbf{y}_1), \ldots, \mathtt{R}_k(\mathbf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$
- Arbitrary valuation



W.l.o.g: OBF does not broadcast $\mathtt{R}_1(\mathrm{a}_1)$

Proof: No Optimal OBF exists

- Arbitrary query: $\mathcal{Q}(\mathbf{x}) \leftarrow \mathtt{R}_1(\mathbf{y}_1), \ldots, \mathtt{R}_k(\mathbf{y}_2)$ $(k \geq 2)$
- Assume: Optimal OBF f for $\mathcal{Q}$
- Arbitrary valuation



OBF exists that broadcasts less

Locally-optimal OBFs

Let f and g be OBFs.

Inclusion: $f \subseteq g$ if $f(I) \subseteq g(I)$ for every instance I

Locally-optimal OBFs

Let f and g be OBFs.

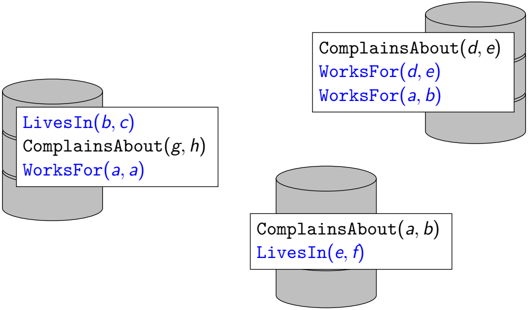
Inclusion: $f \subseteq g$ if $f(I) \subseteq g(I)$ for every instance I

Definition

An OBF f that is correct for a CQ $\mathcal{Q}$ is locally optimal if for every other OBF g that is correct for $\mathcal{Q}$, $g \subseteq f$ implies $f = g$.

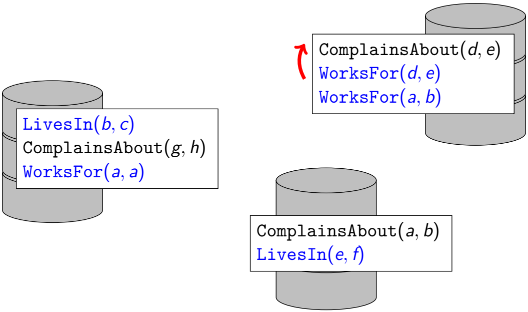
Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$
OBF: Don't broadcast **$\texttt{ComplainsAbout}(x, y)$**



Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$
OBF: Don't broadcast $\mathbf{ComplainsAbout}(\boldsymbol{x}, \boldsymbol{y})$

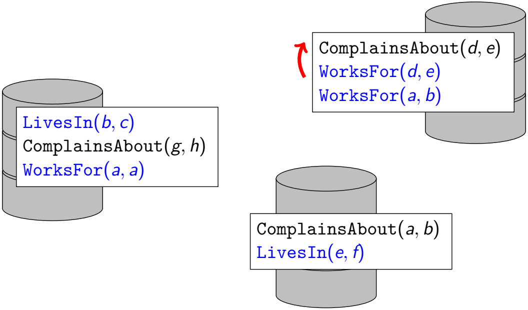


- $\texttt{WorksFor}(d, e)$ requires $\texttt{ComplainsAbout}(d, e)$

Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF: Don't broadcast $\mathbf{ComplainsAbout}(\boldsymbol{x}, \boldsymbol{y})$

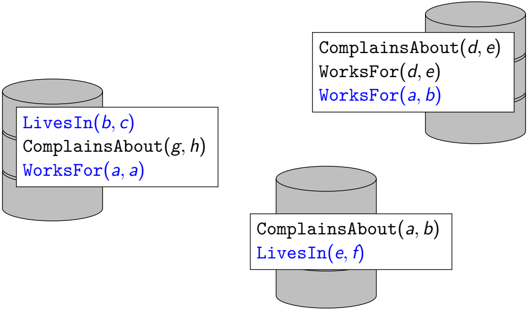


- $\texttt{WorksFor}(d, e)$ requires $\texttt{ComplainsAbout}(d, e)$
- valuations requiring $\texttt{ComplainsAbout}(d, e)$ satisfy locally

Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF: Don't broadcast **ComplainsAbout(x, y)** + don't broadcast **WorksFor(x, y)** if ComplainsAbout(x, y) is present

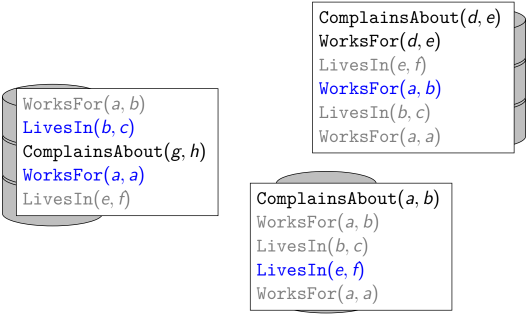


Not necessary to broadcast WorksFor(d, e)

Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

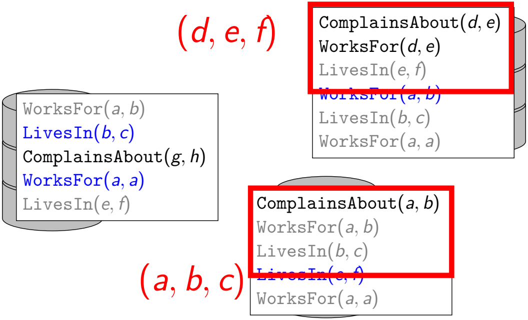
OBF: Don't broadcast **ComplainsAbout(x, y)** + don't broadcast **WorksFor(x, y)** if ComplainsAbout(x, y) is present



Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF: Don't broadcast $\texttt{ComplainsAbout}(x, y)$ + don't broadcast $\texttt{WorksFor}(x, y)$ if $\texttt{ComplainsAbout}(x, y)$ is present



Locally-optimal OBFs

Lemma

Let $\mathcal{Q}$ be a CQ and let f be an OBF for $\mathcal{Q}$. The following are equivalent:

1. f is locally optimal; and
2. for every instance I and fact $\boldsymbol{f}$ for which $\boldsymbol{f} \in f(I \cup \{\boldsymbol{f}\})$, there is an instance J and a fact $\boldsymbol{g}$ such that $\boldsymbol{f} \sim_{\mathcal{Q}} \boldsymbol{g}$, $\boldsymbol{g} \notin I$, $\boldsymbol{f} \notin J$, and $\boldsymbol{g} \in f(J \cup \{\boldsymbol{g}\})$.

Outline

1. Setting and Context
2. Oblivious Broadcasting Functions
3. Correctness & Optimality
4. Broadcast Dependency Sets
5. Conclusion & Future Work

Broadcast Dependency Sets

Building blocks: Equality types

Broadcast Dependency Sets

Building blocks: Equality types

Example

- $\texttt{WorksFor}(x, y), x \neq y$
- $\texttt{ComplainsAbout}(x, y), x = y$

Broadcast Dependency Sets

Building blocks: Equality types

A Broadcast Dependency set is a set of tuples (τ, T), where

- τ is an equality type consistent with atom of $\mathcal{Q}$ (key)
- T is a set of equality types consistent with atoms of $\mathcal{Q}$ (dependency set)
- + additional restrictions

Broadcast Dependency Sets

Building blocks: Equality types

A Broadcast Dependency set is a set of tuples (τ, T), where

- τ is an equality type consistent with atom of $\mathcal{Q}$ (key)
- T is a set of equality types consistent with atoms of $\mathcal{Q}$ (dependency set)
- $+$ additional restrictions

Semantics: Broadcast a fact only if

- it has a consistent equality type; and
- either
  - it does not correspond to a key in the BDS; or
  - the facts represented by the corresponding dependency set are not all present.

Complexity Results

Theorem: Deciding whether BDS is correct for $\mathcal{Q}$ is coNP-complete

Complexity Results

Theorem: Deciding whether BDS is correct for $\mathcal{Q}$ is coNP-complete

Theorem: Deciding whether correct BDS for $\mathcal{Q}$ is locally optimal is in coNP

Complexity Results

Theorem: Deciding whether BDS is correct for $\mathcal{Q}$ is coNP-complete

Theorem: Deciding whether correct BDS for $\mathcal{Q}$ is locally optimal is in coNP

Theorem: Complete characterization for locally optimal, correct OBFs.

OBF Construction

Parameter: sequence $\mathcal{S}$ of all consistent equality-types for $\mathcal{Q}$.

- $D \stackrel{\text{def}}{=} \emptyset$
- Consume types $\tau \in \mathcal{S}$ one-by-one:
  - $values \stackrel{\text{def}}{=} \emptyset$
  - For every key τ' in D compatible with τ, check condition and add to $values$
  - On failure: ignore τ and jump to the next type
  - On success: add $(\tau, values)$ to D.

Output: D

OBF Construction

Parameter: sequence $\mathcal{S}$ of all consistent equality-types for $\mathcal{Q}$.

- $D \stackrel{\text{def}}{=} \emptyset$
- Consume types $\tau \in \mathcal{S}$ one-by-one:
  - $values \stackrel{\text{def}}{=} \emptyset$
  - For every key τ' in D compatible with τ, check condition and add to $values$
  - On failure: ignore τ and jump to the next type
  - On success: add $(\tau, values)$ to D.

Output: D

Theorem: In general: exponential in $\mathcal{Q}$

OBF Construction

Parameter: sequence $\mathcal{S}$ of all consistent equality-types for $\mathcal{Q}$.

- $D \stackrel{\text{def}}{=} \emptyset$
- Consume types $\tau \in \mathcal{S}$ one-by-one:
  - $values \stackrel{\text{def}}{=} \emptyset$
  - For every key τ' in D compatible with τ, check condition and add to $values$
  - On failure: ignore τ and jump to the next type
  - On success: add $(\tau, values)$ to D.

Output: D

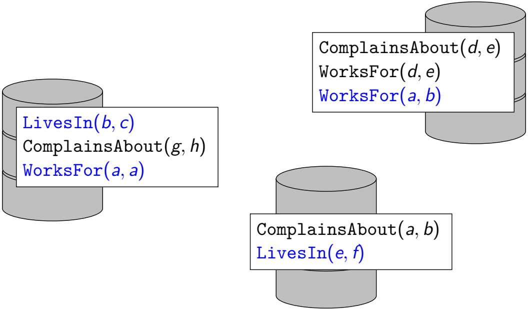
Theorem: In general: exponential in $\mathcal{Q}$

Theorem: polynomial in $\mathcal{Q}$ if only considering relations

Running Example: Relation-Based

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF: Don't broadcast **ComplainsAbout(*x*, *y*)** + don't broadcast **WorksFor(*x*, *y*)** if ComplainsAbout(x, y) is present



Running Example: Partial Atomic Types

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF:

- don't broadcast $\boldsymbol{\texttt{ComplainsAbout}(x, y)}$

Running Example: Partial Atomic Types

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF:

- don't broadcast $\boldsymbol{\texttt{ComplainsAbout}(x, y)}, x \neq y$

Running Example: Partial Atomic Types

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF:

- don't broadcast $\boldsymbol{\texttt{ComplainsAbout}(x, y)}, x \neq y$
- don't broadcast $\boldsymbol{\texttt{WorksFor}(x, y)}, x = y$

Running Example: Partial Atomic Types

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

OBF:

- don't broadcast $\mathbf{ComplainsAbout(}\boldsymbol{x, y}\mathbf{)}, x \neq y$
- don't broadcast $\mathbf{WorksFor(}\boldsymbol{x, y}\mathbf{)}, x = y$
- don't broadcast $\mathbf{ComplainsAbout(}\boldsymbol{x, y}\mathbf{)}, x = y$ if $\texttt{WorksFor}(x, y)$ is present
- don't broadcast $\mathbf{WorksFor(}\boldsymbol{x, y}\mathbf{)}, x \neq y$ if $\texttt{ComplainsAbout}(x, y)$ is present

Running Example: Partial Atomic Types

$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$

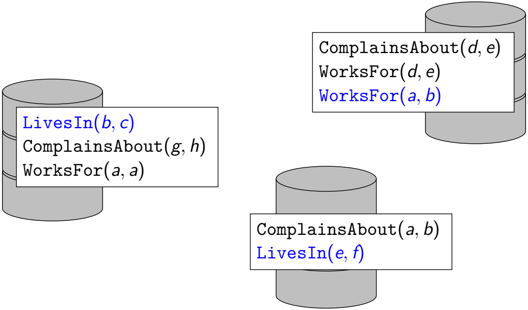
OBF:

- don't broadcast $\texttt{ComplainsAbout}(x, y), x \neq y$
- don't broadcast $\texttt{WorksFor}(x, y), x = y$
- don't broadcast $\texttt{ComplainsAbout}(x, y), x = y$ if $\texttt{WorksFor}(x, y)$ is present
- don't broadcast $\texttt{WorksFor}(x, y), x \neq y$ if $\texttt{ComplainsAbout}(x, y)$ is present

Correct + locally optimal

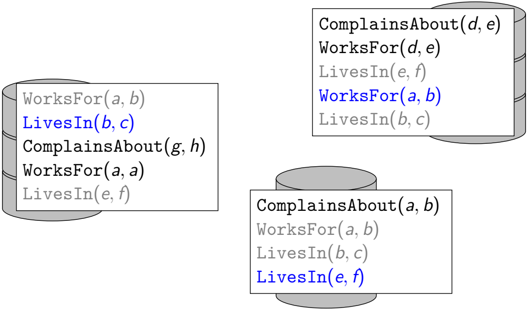
Running Example: Partial Atomic Types

$$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$$



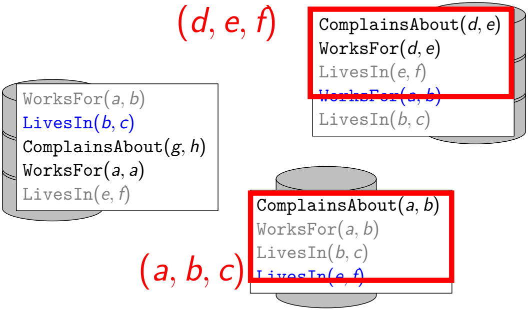
Running Example: Partial Atomic Types

$$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$$



Running Example: Partial Atomic Types

$$\mathcal{Q}(x, y, z) \leftarrow \texttt{ComplainsAbout}(x, y), \texttt{WorksFor}(x, y), \texttt{LivesIn}(y, z)$$



Conclusion & Future Work

Summary: Semantical and syntactical characterization of OBFs, complexity results, and construction.

Conclusion & Future Work

Summary: Semantical and syntactical characterization of OBFs, complexity results, and construction.

Future work:

- Beyond full CQs without self-joins
- Less restrictions on messages
- Alternative notions of optimality