Principles of Guarded Structural Indexing

On Guarded Simulations and Acyclic First Order Languages

François Picalausa [1] George H. L. Fletcher [2] Jan Hidders [3]
Stijn Vansummeren [1]

[1]Université Libre de Bruxelles (ULB), Belgium

[2]Eindhoven University of Technology, The Netherlands

[3]Delft University of Technology, The Netherlands

ICDT 2014

Background

- Increased interest in tree-based and graph-based data formats: XML, RDF, JSON, social networks

Background

- Increased interest in tree-based and graph-based data formats: XML, RDF, JSON, social networks
- Rise of specialized graph storage and query processing engines

Background

- Increased interest in tree-based and graph-based data formats: XML, RDF, JSON, social networks
- Rise of specialized graph storage and query processing engines
- Exploitation of graph topology for performance on large input graphs, e.g., structural indexes

Background

- Increased interest in tree-based and graph-based data formats: XML, RDF, JSON, social networks
- Rise of specialized graph storage and query processing engines
- Exploitation of graph topology for performance on large input graphs, e.g., structural indexes

Central Question

Can structural indexes be generalized for arbitrary relational databases?

Structural Indexes

Key Idea

- We consider a class of graph queries $\mathcal{Q}$
  - e.g., reachability queries, XPath queries, modal or temporal logic queries, ...

Structural Indexes

Key Idea

- We consider a class of graph queries $\mathcal{Q}$
  - e.g., reachability queries, XPath queries, modal or temporal logic queries, ...
- We group and merge the nodes of input graph G to obtain a more compact representation: the structural index for G with respect to $\mathcal{Q}$

Structural Indexes

Key Idea

- We consider a class of graph queries $\mathcal{Q}$
  - e.g., reachability queries, XPath queries, modal or temporal logic queries, ...
- We group and merge the nodes of input graph G to obtain a more compact representation: the structural index for G with respect to $\mathcal{Q}$
- We group nodes such that any query $Q \in \mathcal{Q}$ can be answered

Structural Indexes

Key Idea

- We consider a class of graph queries $\mathcal{Q}$
  - e.g., reachability queries, XPath queries, modal or temporal logic queries, ...
- We group and merge the nodes of input graph G to obtain a more compact representation: the structural index for G with respect to $\mathcal{Q}$
- We group nodes such that any query $Q \in \mathcal{Q}$ can be answered
  - directly on the structural index of G instead of on G itself, or

Structural Indexes

Key Idea

- We consider a class of graph queries $\mathcal{Q}$
  - e.g., reachability queries, XPath queries, modal or temporal logic queries, ...
- We group and merge the nodes of input graph G to obtain a more compact representation: the structural index for G with respect to $\mathcal{Q}$
- We group nodes such that any query $Q \in \mathcal{Q}$ can be answered
  - directly on the structural index of G instead of on G itself, or
  - directly on G but using pruning information from the index.

Structural Indexes

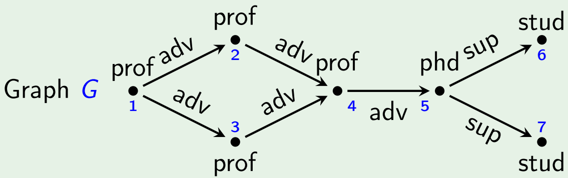
Key Idea

- We consider a class of graph queries $\mathcal{Q}$
  - e.g., reachability queries, XPath queries, modal or temporal logic queries, ...
- We group and merge the nodes of input graph G to obtain a more compact representation: the structural index for G with respect to $\mathcal{Q}$
- We group nodes such that any query $Q \in \mathcal{Q}$ can be answered
  - directly on the structural index of G instead of on G itself, or
  - directly on G but using pruning information from the index.
- Since the index is typically (much) smaller than G itself, this can be significantly faster than evaluating Q directly over G.

Structural Indexes

Example

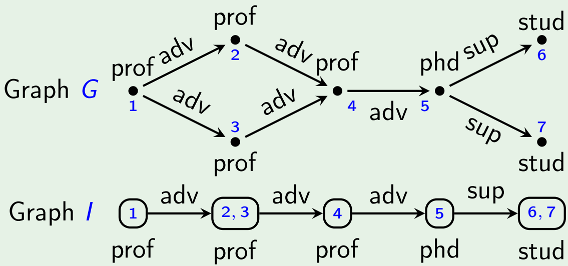
Example (Academic relations graph)



Structural Indexes

Example

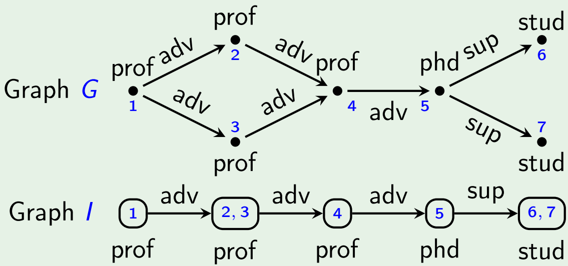
Example (Academic relations graph)



Structural Indexes

Example

Example (Academic relations graph)

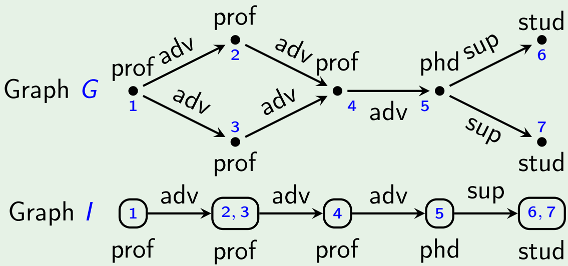


- Each node in I is actually a set of nodes in G.

Structural Indexes

Example

Example (Academic relations graph)



- Each node in I is actually a set of nodes in G.
- There is an edge between sets V and W in I if there is an edge between some $v \in V$ and some $w \in W$ in G.

(Bi)simulation on Labeled Graphs

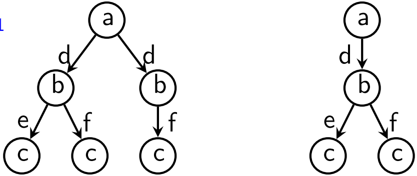
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

(Bi)simulation on Labeled Graphs

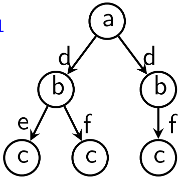
Definition

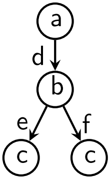
A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1





Graph G_2

A simulation of G_1 in G_2:

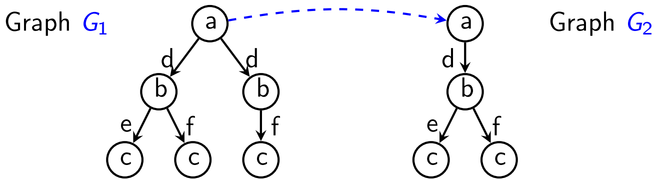
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



A simulation of G_1 in G_2:

(Bi)simulation on Labeled Graphs

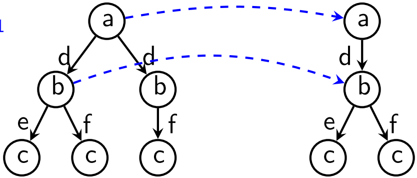
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

A simulation of G_1 in G_2:

(Bi)simulation on Labeled Graphs

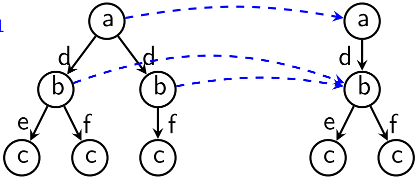
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

A simulation of G_1 in G_2:

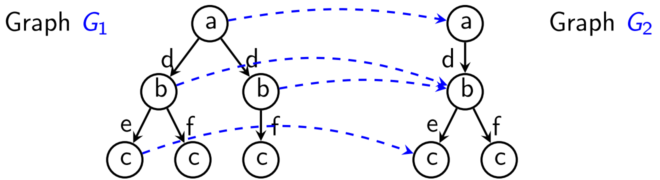
(Bi)simulation on Labeled Graphs

Definition

A **simulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



A simulation of G_1 in G_2:

(Bi)simulation on Labeled Graphs

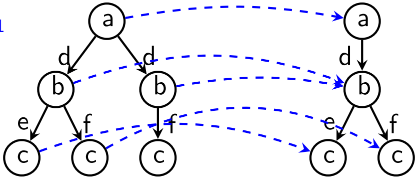
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

A simulation of G_1 in G_2:

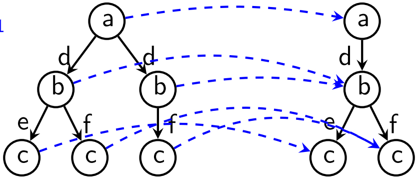
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

- (lab) it relates only nodes with the same label, and
- (forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

A simulation of G_1 in G_2:

(Bi)simulation on Labeled Graphs

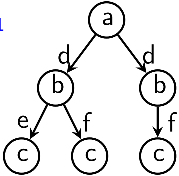
Definition

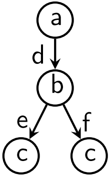
A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1





Graph G_2

A simulation of G_2 in G_1:

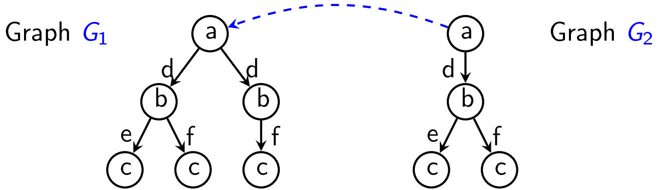
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



A simulation of G_2 in G_1:

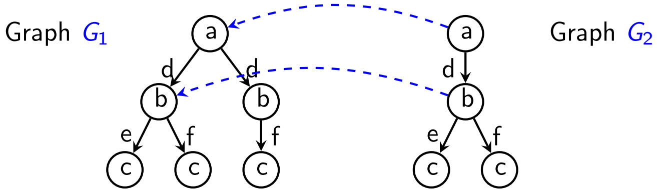
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



A simulation of G_2 in G_1:

(Bi)simulation on Labeled Graphs

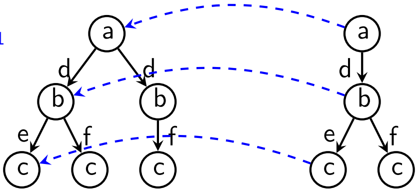
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

A simulation of G_2 in G_1:

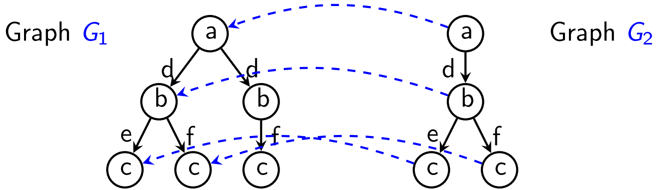
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



A simulation of G_2 in G_1:

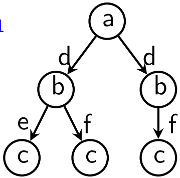
(Bi)simulation on Labeled Graphs

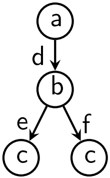
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

- (lab) it relates only nodes with the same label, and
- (forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1





Graph G_2

Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

(Bi)simulation on Labeled Graphs

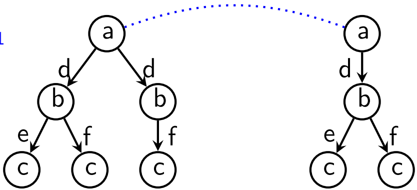
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

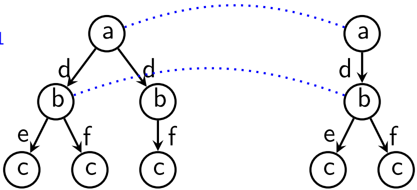
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

- (lab) it relates only nodes with the same label, and
- (forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

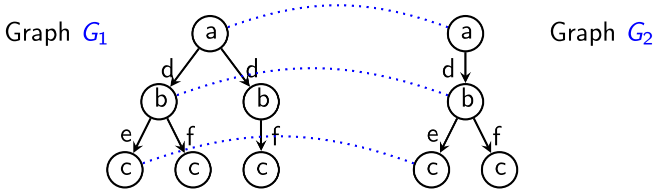
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

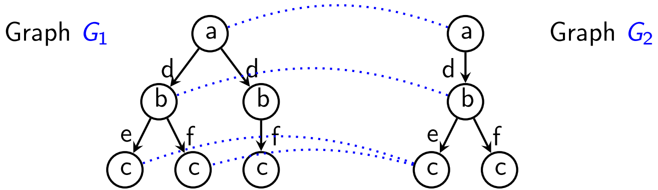
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

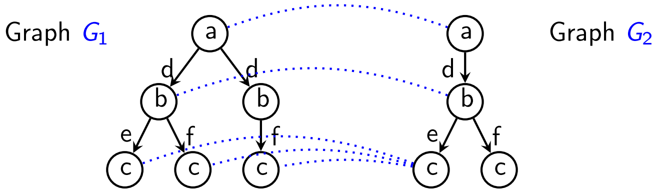
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



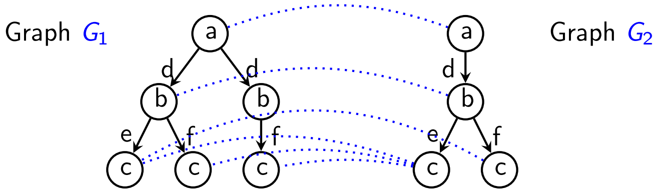
Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

- (lab) it relates only nodes with the same label, and
- (forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

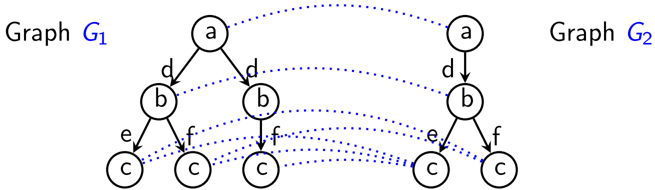
(Bi)simulation on Labeled Graphs

Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.



Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

(Bi)simulation on Labeled Graphs

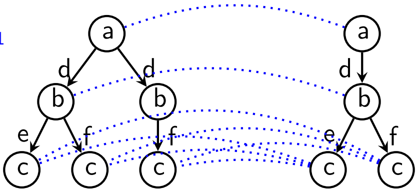
Definition

A simulation of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

Graph G_1



Graph G_2

Nodes n and m are called similar if there is a simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

(Bi)simulation on Labeled Graphs

Definition

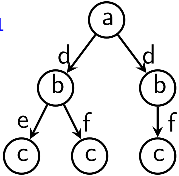
A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

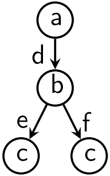
(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

(back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ ...

Graph G_1





Graph G_2

(Bi)simulation on Labeled Graphs

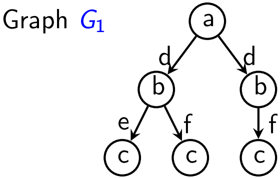
Definition

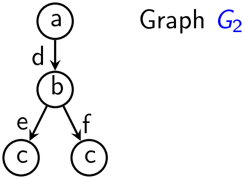
A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

(back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ ...





Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

(Bi)simulation on Labeled Graphs

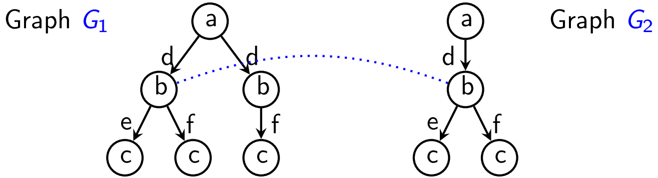
Definition

A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

(back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ ...



Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

(Bi)simulation on Labeled Graphs

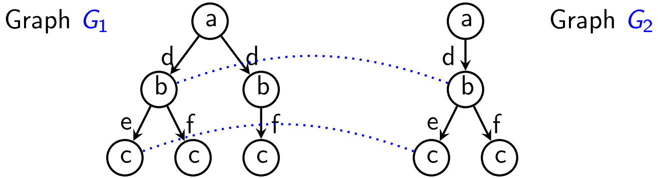
Definition

A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

(back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ . . .



Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

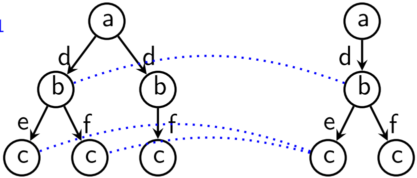
(Bi)simulation on Labeled Graphs

Definition

A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

- (lab) it relates only nodes with the same label, and
- (forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.
- (back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ . . .

Graph G_1



Graph G_2

Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

(Bi)simulation on Labeled Graphs

Definition

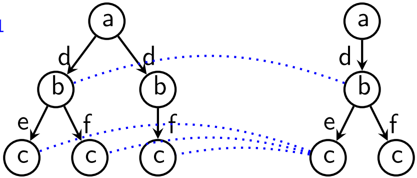
A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

(back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ . . .

Graph G_1



Graph G_2

Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

(Bi)simulation on Labeled Graphs

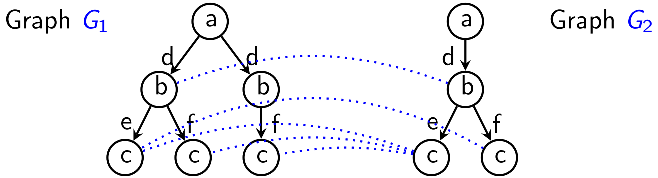
Definition

A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

(back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ ...



Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

(Bi)simulation on Labeled Graphs

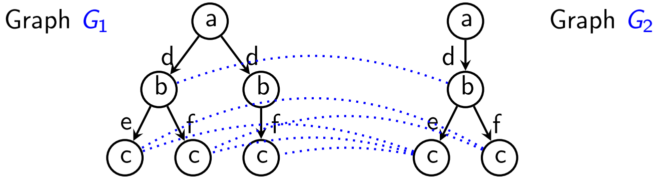
Definition

A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

(lab) it relates only nodes with the same label, and

(forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.

(back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ . . .



Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

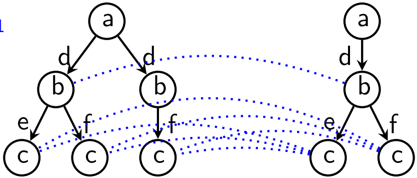
(Bi)simulation on Labeled Graphs

Definition

A **bisimulation** of G_1 in G_2 is a binary relation $T \subseteq V_1 \times V_2$ s.t.

- (lab) it relates only nodes with the same label, and
- (forth) for every $(n, m) \in T$ and every $(n, \lambda, n') \in E_1$ there exists $(m, \lambda, m') \in E_2$ such that $(n', m') \in T$.
- (back) for every $(n, m) \in T$ and every $(m, \lambda, m') \in E_2$ . . .

Graph G_1

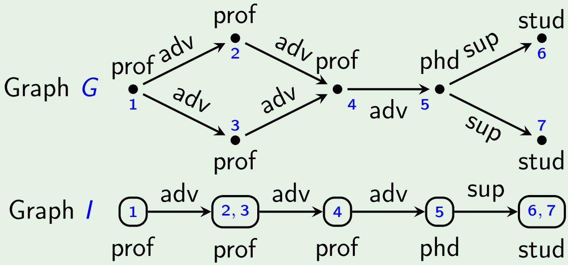


Graph G_2

Nodes n and m are called bisimilar if there is a bisimulation from G_1 to G_2 that maps n to m.

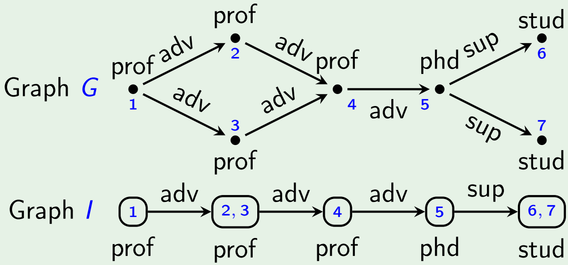
Simulation Relations for Structural Indexes

Example (Academic relations graph)



Simulation Relations for Structural Indexes

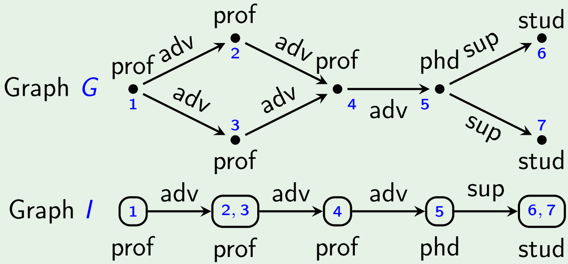
Example (Academic relations graph)



- In the context of XML (bi)simulation-based structural indexes are known to be covering for certain XPath fragments, i.e., query returns on the index (a pointer to) the exact answer.

Simulation Relations for Structural Indexes

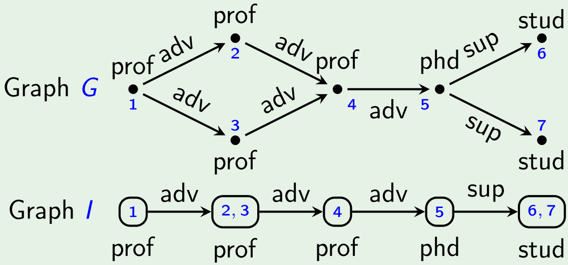
Example (Academic relations graph)



- In the context of XML (bi)simulation-based structural indexes are known to be covering for certain XPath fragments, i.e., query returns on the index (a pointer to) the exact answer.
- For example: Q is "select all professors that advised someone who is currently a professor who is advising a PhD student"

Simulation Relations for Structural Indexes

Example (Academic relations graph)



- In the context of XML (bi)simulation-based structural indexes are known to be covering for certain XPath fragments, i.e., query returns on the index (a pointer to) the exact answer.
- For example: Q is "select all professors that advised someone who is currently a professor who is advising a PhD student"
  - Applying Q on I gives the node $\{2, 3\}$ which is the correct answer

General Methodology for Structural Indexing

- We move to a more general relational setting.

General Methodology for Structural Indexing

- We move to a more general relational setting.
- Approach for covering indexes for a given target query language $\mathcal{Q}$:

General Methodology for Structural Indexing

- We move to a more general relational setting.
- Approach for covering indexes for a given target query language $\mathcal{Q}$:
  1. A language-independent structural characterization of query invariance, characterizing when data objects (in our setting: relational tuples) cannot be distinguished by any query in $\mathcal{Q}$.

General Methodology for Structural Indexing

- We move to a more general relational setting.
- Approach for covering indexes for a given target query language $\mathcal{Q}$:
  1. A language-independent structural characterization of query invariance, characterizing when data objects (in our setting: relational tuples) cannot be distinguished by any query in $\mathcal{Q}$.
  2. An efficient grouping algorithm for data objects that cannot be distinguished by any query in $\mathcal{Q}$.

General Methodology for Structural Indexing

- We move to a more general relational setting.
- Approach for covering indexes for a given target query language $\mathcal{Q}$:
  1. A language-independent structural characterization of query invariance, characterizing when data objects (in our setting: relational tuples) cannot be distinguished by any query in $\mathcal{Q}$.
  2. An efficient grouping algorithm for data objects that cannot be distinguished by any query in $\mathcal{Q}$.
  3. A data structure, i.e., the index, that exploits this grouping to support query answering by means of the index.

General Methodology for Structural Indexing

- We move to a more general relational setting.
- Approach for covering indexes for a given target query language $\mathcal{Q}$:
  1. A language-independent structural characterization of query invariance, characterizing when data objects (in our setting: relational tuples) cannot be distinguished by any query in $\mathcal{Q}$.
  2. An efficient grouping algorithm for data objects that cannot be distinguished by any query in $\mathcal{Q}$.
  3. A data structure, i.e., the index, that exploits this grouping to support query answering by means of the index.
- We focus here on structural characterization of query invariance for strict conjunctive queries, i.e., queries that select tuples

General Methodology for Structural Indexing

- We move to a more general relational setting.
- Approach for covering indexes for a given target query language $\mathcal{Q}$:
  1. A language-independent structural characterization of query invariance, characterizing when data objects (in our setting: relational tuples) cannot be distinguished by any query in $\mathcal{Q}$.
  2. An efficient grouping algorithm for data objects that cannot be distinguished by any query in $\mathcal{Q}$.
  3. A data structure, i.e., the index, that exploits this grouping to support query answering by means of the index.
- We focus here on structural characterization of query invariance for strict conjunctive queries, i.e., queries that select tuples
  - Formally: All variables in the head occur in a single atom in the body, e.g., $ans(b, c) \leftarrow R(a, b), S(b, c, d), R(b, d)$.

General Methodology for Structural Indexing

- We move to a more general relational setting.
- Approach for covering indexes for a given target query language $\mathcal{Q}$:
  1. A language-independent structural characterization of query invariance, characterizing when data objects (in our setting: relational tuples) cannot be distinguished by any query in $\mathcal{Q}$.
  2. An efficient grouping algorithm for data objects that cannot be distinguished by any query in $\mathcal{Q}$.
  3. A data structure, i.e., the index, that exploits this grouping to support query answering by means of the index.
- We focus here on structural characterization of query invariance for strict conjunctive queries, i.e., queries that select tuples
  - Formally: All variables in the head occur in a single atom in the body, e.g., $ans(b, c) \leftarrow R(a, b), S(b, c, d), R(b, d)$.
  - This keeps the indexes small

Indistinguishability under Conjunctive Queries

All conjunctive queries are invariant under homomorphisms:

Theorem ([Chandra & Harel, 1980])

For all databases db_1 and db_2 and all tuples $\overline{a}_1$ and $\overline{a}_2$, if there exists a homomorphism f from db_1 to db_2 such that $f(\overline{a}_1) = \overline{a}_2$, then for every conjunctive query Q, if $\overline{a}_1 \in Q(db_1)$ then also $\overline{a}_2 \in Q(db_2)$.

Indistinguishability under Conjunctive Queries

All conjunctive queries are invariant under homomorphisms:

Theorem ([Chandra & Harel, 1980])

For all databases db_1 and db_2 and all tuples $\overline{a}_1$ and $\overline{a}_2$, if there exists a homomorphism f from db_1 to db_2 such that $f(\overline{a}_1) = \overline{a}_2$, then for every conjunctive query Q, if $\overline{a}_1 \in Q(db_1)$ then also $\overline{a}_2 \in Q(db_2)$.

Invariance under homomorphisms in fact is a characterization of the conjunctive queries (modulo union):

Theorem ([Rossman, 2008])

A query expressible in first order logic (FO) is invariant under homomorphisms on finite structures if, and only if, it is equivalent in the finite to a union of conjunctive queries.

Tractable indistinguishability?

- So invariance under homomorphisms seems the “right” notion of indistinguishability.

Tractable indistinguishability?

- So invariance under homomorphisms seems the "right" notion of indistinguishability.
- But very expensive: deciding if given databases db_1 and db_2 and tuples $\overline{a}_1$ and $\overline{a}_2$, there exists a homomorphism f from db_1 to db_2 such that $f(\overline{a}_1) = \overline{a}_2$, is NP-complete.

Tractable indistinguishability?

- So invariance under homomorphisms seems the "right" notion of indistinguishability.
- But very expensive: deciding if given databases db_1 and db_2 and tuples $\bar{a}_1$ and $\bar{a}_2$, there exists a homomorphism f from db_1 to db_2 such that $f(\bar{a}_1) = \bar{a}_2$, is NP-complete.

Question

Is there a useful fragment of strict conjunctive queries that has a tractable notion of indistinguishability?

Tractable indistinguishability?

- So invariance under homomorphisms seems the "right" notion of indistinguishability.
- But very expensive: deciding if given databases db_1 and db_2 and tuples $\overline{a}_1$ and $\overline{a}_2$, there exists a homomorphism f from db_1 to db_2 such that $f(\overline{a}_1) = \overline{a}_2$, is NP-complete.

Question

Is there a useful fragment of strict conjunctive queries that has a tractable notion of indistinguishability?

- Two approaches:

Tractable indistinguishability?

- So invariance under homomorphisms seems the "right" notion of indistinguishability.
- But very expensive: deciding if given databases db_1 and db_2 and tuples $\bar{a}_1$ and $\bar{a}_2$, there exists a homomorphism f from db_1 to db_2 such that $f(\bar{a}_1) = \bar{a}_2$, is NP-complete.

Question

Is there a useful fragment of strict conjunctive queries that has a tractable notion of indistinguishability?

- Two approaches:
  - Start from well-known well-behaved fragments, such as acyclic conjunctive queries.

Tractable indistinguishability?

- So invariance under homomorphisms seems the "right" notion of indistinguishability.
- But very expensive: deciding if given databases db_1 and db_2 and tuples $\overline{a}_1$ and $\overline{a}_2$, there exists a homomorphism f from db_1 to db_2 such that $f(\overline{a}_1) = \overline{a}_2$, is NP-complete.

Question

Is there a useful fragment of strict conjunctive queries that has a tractable notion of indistinguishability?

- Two approaches:
  - Start from well-known well-behaved fragments, such as acyclic conjunctive queries.
  - Start from tractable relations such as simulation and bisimulation.

Tractable indistinguishability?

- So invariance under homomorphisms seems the "right" notion of indistinguishability.
- But very expensive: deciding if given databases db_1 and db_2 and tuples $\bar{a}_1$ and $\bar{a}_2$, there exists a homomorphism f from db_1 to db_2 such that $f(\bar{a}_1) = \bar{a}_2$, is NP-complete.

Question

Is there a useful fragment of strict conjunctive queries that has a tractable notion of indistinguishability?

- Two approaches:
  - Start from well-known well-behaved fragments, such as acyclic conjunctive queries.
  - Start from tractable relations such as simulation and bisimulation.
- Main informal result: leads to the same answer.

Guarded (Bi)simulation on Relational Databases

Setting up

Note: We give here alternative definitions of guarded (bi)similarity which

- are equivalent to the original ones, but
- illustrate better the link with labeled graphs.

Note: We give here alternative definitions of guarded (bi)similarity which

- are equivalent to the original ones, but
- illustrate better the link with labeled graphs.

Intuitive idea

Guarded (Bi)simulation on Relational Databases

Setting up

Note: We give here alternative definitions of guarded (bi)similarity which

- are equivalent to the original ones, but
- illustrate better the link with labeled graphs.

$r(a, b, c)$
$r(d, a, e)$
$r(f, a, g)$
$s(e, h, i)$
$s(d, j, k)$
$s(f, l, m)$
$\ldots$

Intuitive idea

- A database = set of facts over a fixed relational schema.

Guarded (Bi)simulation on Relational Databases

Setting up

Note: We give here alternative definitions of guarded (bi)similarity which

- are equivalent to the original ones, but
- illustrate better the link with labeled graphs.

$r(a, b, c)$
$r(d, a, e)$
$r(f, a, g)$
$s(e, h, i)$
$s(d, j, k)$
$s(f, l, m)$
$\ldots$

Intuitive idea

- A database = set of facts over a fixed relational schema.
- Facts are the basic units of information (not data values)

Guarded (Bi)simulation on Relational Databases

Setting up

Note: We give here alternative definitions of guarded (bi)similarity which

- are equivalent to the original ones, but
- illustrate better the link with labeled graphs.

$r(a, b, c)$
$r(d, a, e)$
$r(f, a, g)$
$s(e, h, i)$
$s(d, j, k)$
$s(f, l, m)$
$\ldots$

Intuitive idea

- A database = set of facts over a fixed relational schema.
- Facts are the basic units of information (not data values)
- So the facts become our nodes

Guarded (Bi)simulation on Relational Databases

Setting up

Note: We give here alternative definitions of guarded (bi)similarity which

- are equivalent to the original ones, but
- illustrate better the link with labeled graphs.

$r(a, b, c)$
$r(d, a, e)$
$r(f, a, g)$
$s(e, h, i)$
$s(d, j, k)$
$s(f, l, m)$
$\ldots$

Intuitive idea

- A database = set of facts over a fixed relational schema.
- Facts are the basic units of information (not data values)
- So the facts become our nodes
- But what are then the edges?

Guarded (Bi)simulation on Relational Databases

Equality types

Definition (Equality type)

For tuples $\overline{a} = (a_1, \ldots, a_k)$ and $\overline{b} = (b_1, \ldots, b_l)$ their equality type is $eqtp(\overline{a}, \overline{b}) := \{(i, j) \mid a_i = b_j\}$.

Definition (Equality type)

For tuples $\bar{a} = (a_1, \ldots, a_k)$ and $\bar{b} = (b_1, \ldots, b_l)$ their equality type is $eqtp(\bar{a}, \bar{b}) := \{(i, j) \mid a_i = b_j\}$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$

D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Guarded (Bi)simulation on Relational Databases

Equality types

Definition (Equality type)

For tuples $\bar{a} = (a_1, \ldots, a_k)$ and $\bar{b} = (b_1, \ldots, b_l)$ their equality type is $eqtp(\bar{a}, \bar{b}) := \{(i, j) \mid a_i = b_j\}$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$

D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

$$eqtp(t_1, t_2) = \{(1, 2)\}$$

Guarded (Bi)simulation on Relational Databases

Equality types

Definition (Equality type)

For tuples $\bar{a} = (a_1, \ldots, a_k)$ and $\bar{b} = (b_1, \ldots, b_l)$ their equality type is $eqtp(\bar{a}, \bar{b}) := \{(i, j) \mid a_i = b_j\}$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$

D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

$eqtp(t_1, t_2) = \{(1, 2)\}$ and $eqtp(t_1, t_1) = \{(1, 1), (2, 2), (3, 3)\}$.

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

- (lab) it relates only facts with the same relation name, and
- (forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	

D_2	

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

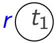
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$



D_2	

Guarded (Bi)simulation on Relational Databases

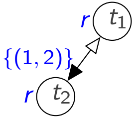
The (bi)simulation relations

Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

- (lab) it relates only facts with the same relation name, and
- (forth) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{c}) \in D_1$ there exists $s(\bar{d}) \in D_2$ such that $(s(\bar{c}), s(\bar{d})) \in T$ and $eqtp(\bar{a}, \bar{c}) \subseteq eqtp(\bar{b}, \bar{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$



D_2	

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

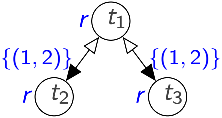
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$



D_2	

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

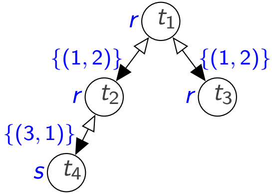
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$



D_2	

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

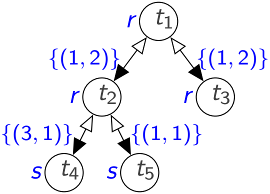
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$



D_2	

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

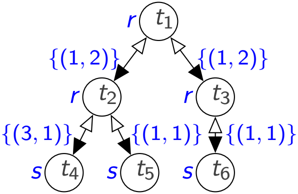
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

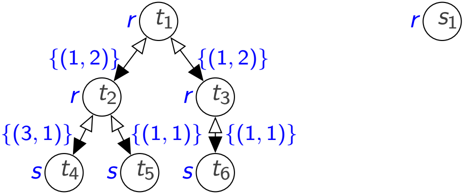
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

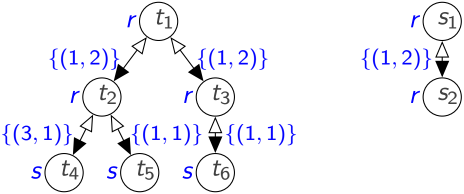
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

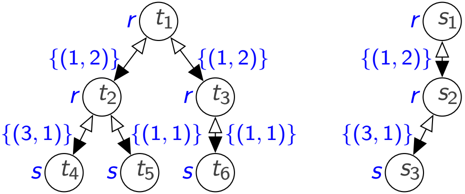
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

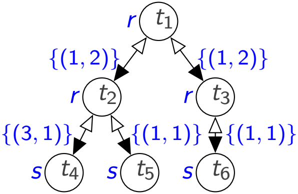
Definition

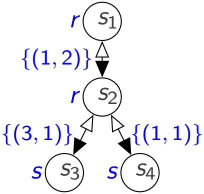
A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$





D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Guarded (Bi)simulation on Relational Databases

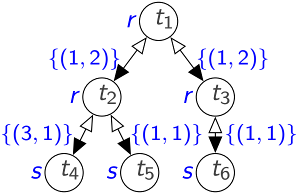
The (bi)simulation relations

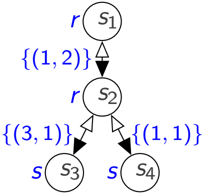
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

- (lab) it relates only facts with the same relation name, and
- (forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$





D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_1 in G_2:

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

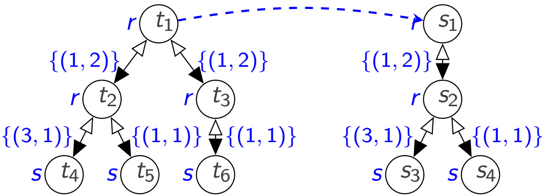
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_1 in G_2:

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

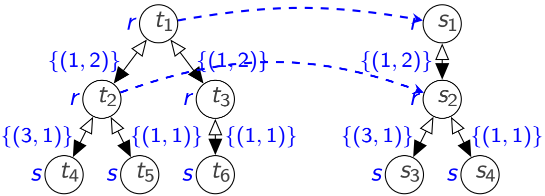
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{c}) \in D_1$ there exists $s(\bar{d}) \in D_2$ such that $(s(\bar{c}), s(\bar{d})) \in T$ and $eqtp(\bar{a}, \bar{c}) \subseteq eqtp(\bar{b}, \bar{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_1 in G_2:

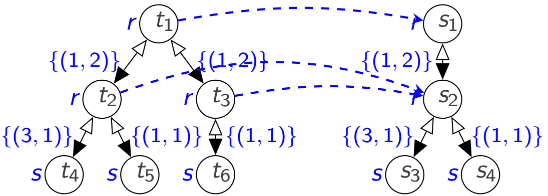
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_1 in G_2:

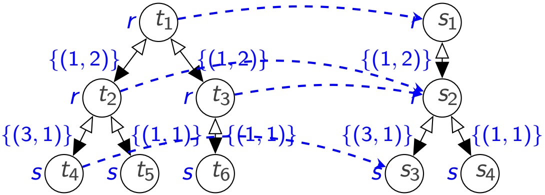
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

	D_1
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



	D_2
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_1 in G_2:

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

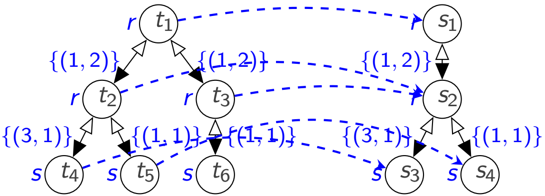
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_1 in G_2:

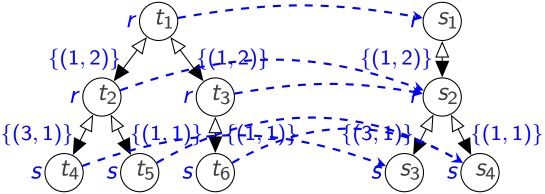
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_1 in G_2:

Guarded (Bi)simulation on Relational Databases

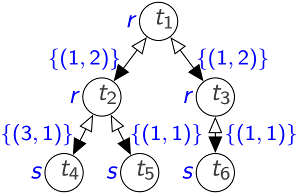
The (bi)simulation relations

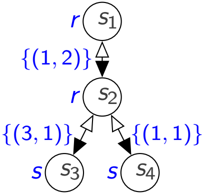
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

- (lab) it relates only facts with the same relation name, and
- (forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$





D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_2 in G_1:

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

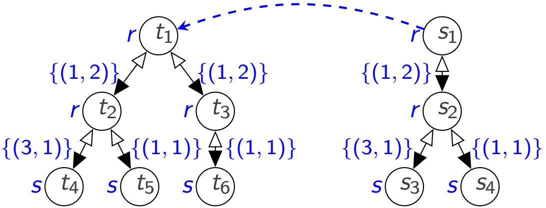
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_2 in G_1:

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

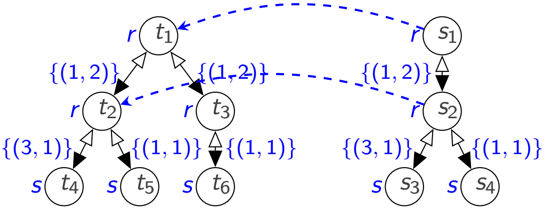
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

	D_1
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



	D_2
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_2 in G_1:

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

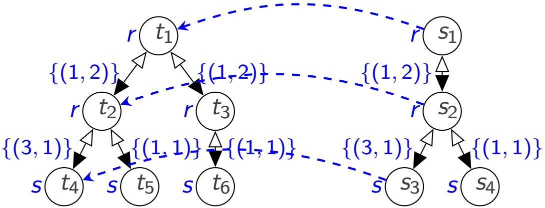
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

A guarded simulation of G_2 in G_1:

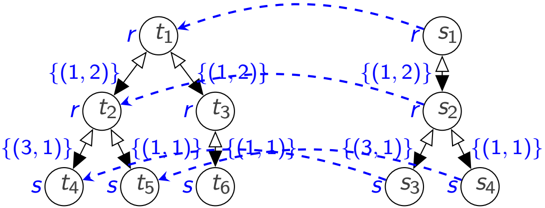
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a,b,c)$
t_2	$r(d,a,e)$
t_3	$r(f,a,g)$
t_4	$s(e,h,i)$
t_5	$s(d,j,k)$
t_6	$s(f,l,m)$



D_2	
s_1	$r(n,o,p)$
s_2	$r(q,n,r)$
s_3	$r(r,s,t)$
s_4	$r(q,u,v)$

A guarded simulation of G_2 in G_1:

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

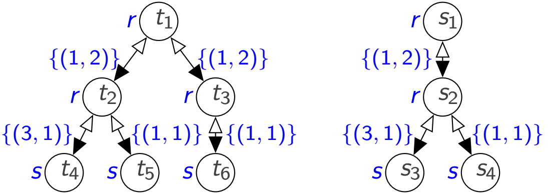
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Nodes n and m are called guarded similar if there is a guarded simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

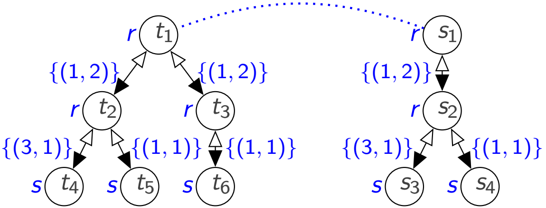
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Nodes n and m are called guarded similar if there is a guarded simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

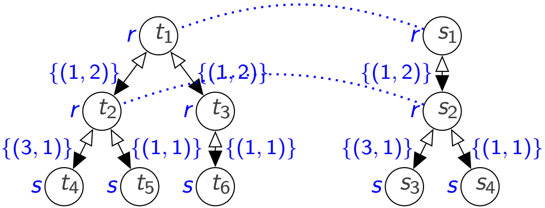
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Nodes n and m are called guarded similar if there is a guarded simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

Guarded (Bi)simulation on Relational Databases

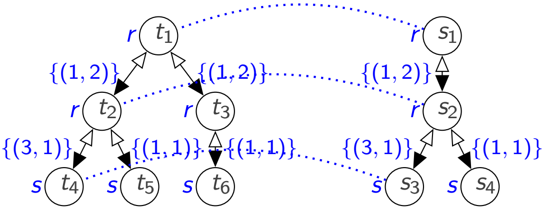
The (bi)simulation relations

Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

- (lab) it relates only facts with the same relation name, and
- (forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Nodes n and m are called guarded similar if there is a guarded simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

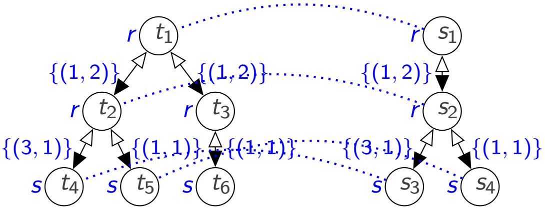
Definition

A guarded simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\overline{a}), r(\overline{b})) \in T$ and every $s(\overline{c}) \in D_1$ there exists $s(\overline{d}) \in D_2$ such that $(s(\overline{c}), s(\overline{d})) \in T$ and $eqtp(\overline{a}, \overline{c}) \subseteq eqtp(\overline{b}, \overline{d})$.

	D_1
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



	D_2
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Nodes n and m are called guarded similar if there is a guarded simulation from G_1 to G_2 that maps n to m, and one from G_2 to G_1 that maps m to n.

Guarded (Bi)simulation on Relational Databases

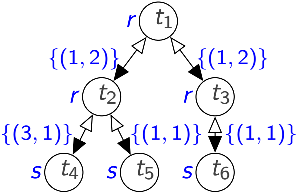
The (bi)simulation relations

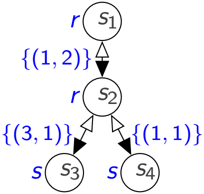
Definition

A guarded **bi**simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

- (lab) it relates only facts with the same relation name, and
- (forth) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{c}) \in D_1$ there exists $s(\bar{d}) \in D_2$ such that $(s(\bar{c}), s(\bar{d})) \in T$ and $eqtp(\bar{a}, \bar{c}) = eqtp(\bar{b}, \bar{d})$.
- (back) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{d}) \in D_2$ …

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$





D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

Definition

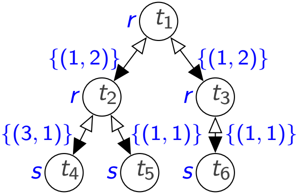
A guarded **bi**simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

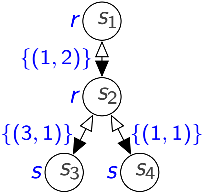
(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{c}) \in D_1$ there exists $s(\bar{d}) \in D_2$ such that $(s(\bar{c}), s(\bar{d})) \in T$ and $eqtp(\bar{a}, \bar{c}) = eqtp(\bar{b}, \bar{d})$.

(back) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{d}) \in D_2$ . . .

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$





D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Nodes n and m are called guarded bisimilar if there is a guarded bisimulation from G_1 to G_2 that maps n to m.

Guarded (Bi)simulation on Relational Databases

The (bi)simulation relations

Definition

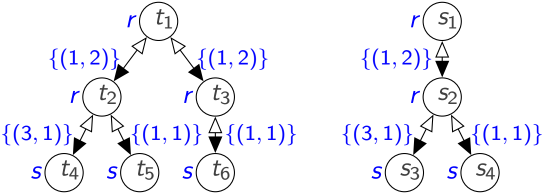
A guarded **bi**simulation of D_1 in D_2 is a binary relation $T \subseteq D_1 \times D_2$ s.t.

(lab) it relates only facts with the same relation name, and

(forth) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{c}) \in D_1$ there exists $s(\bar{d}) \in D_2$ such that $(s(\bar{c}), s(\bar{d})) \in T$ and $eqtp(\bar{a}, \bar{c}) = eqtp(\bar{b}, \bar{d})$.

(back) for every $(r(\bar{a}), r(\bar{b})) \in T$ and every $s(\bar{d}) \in D_2$ . . .

D_1	
t_1	$r(a, b, c)$
t_2	$r(d, a, e)$
t_3	$r(f, a, g)$
t_4	$s(e, h, i)$
t_5	$s(d, j, k)$
t_6	$s(f, l, m)$



D_2	
s_1	$r(n, o, p)$
s_2	$r(q, n, r)$
s_3	$r(r, s, t)$
s_4	$r(q, u, v)$

Nodes n and m are called guarded bisimilar if there is a guarded bisimulation from G_1 to G_2 that maps n to m. **None are!!**

Inspiring results

For the Guarded Fragment of FOL:

Theorem ([Andréka, Németi & van Benthem 1998][Otto 2012])

The GF is invariant under guarded bisimulation. Moreover, a query expressible in FO is invariant under guarded bisimulation on finite structures if, and only if, it is equivalent in the finite to a query expressible in GF.

Inspiring results

For the Guarded Fragment of FOL:

Theorem ([Andréka, Németi & van Benthem 1998][Otto 2012])

The GF is invariant under guarded bisimulation. Moreover, a query expressible in FO is invariant under guarded bisimulation on finite structures if, and only if, it is equivalent in the finite to a query expressible in GF.

Subsequently equivalence in expressive power was shown for:

Inspiring results

For the Guarded Fragment of FOL:

Theorem ([Andréka, Németi & van Benthem 1998][Otto 2012])

The GF is invariant under guarded bisimulation. Moreover, a query expressible in FO is invariant under guarded bisimulation on finite structures if, and only if, it is equivalent in the finite to a query expressible in GF.

Subsequently equivalence in expressive power was shown for:

- strict GF and strict acyclic FO [Flum, Frick & Grohe, 2002]

Inspiring results

For the Guarded Fragment of FOL:

Theorem ([Andréka, Németi & van Benthem 1998][Otto 2012])

The GF is invariant under guarded bisimulation. Moreover, a query expressible in FO is invariant under guarded bisimulation on finite structures if, and only if, it is equivalent in the finite to a query expressible in GF.

Subsequently equivalence in expressive power was shown for:

- strict GF and strict acyclic FO [Flum, Frick & Grohe, 2002]
- strict GF and the semi-join algebra [Leinders, Marx, Tyszkiewicz & Van den Bussche, 2005]

Inspiring results

For the Guarded Fragment of FOL:

Theorem ([Andréka, Németi & van Benthem 1998][Otto 2012])

The GF is invariant under guarded bisimulation. Moreover, a query expressible in FO is invariant under guarded bisimulation on finite structures if, and only if, it is equivalent in the finite to a query expressible in GF.

Subsequently equivalence in expressive power was shown for:

- strict GF and strict acyclic FO [Flum, Frick & Grohe, 2002]
- strict GF and the semi-join algebra [Leinders, Marx, Tyszkiewicz & Van den Bussche, 2005]
- primitive positive fragment of strict GF and acyclic strict conjunctive queries [Gottlob, Leone & Scarcello, 2003]

Our Main Result

- We define FACQ: the class of freely acyclic conjunctive queries:

Our Main Result

- We define FACQ: the class of freely acyclic conjunctive queries:
  - A conjunctive query of the form $\mathit{head} \leftarrow \mathit{body}$ is freely acyclic if the boolean conjunctive query $() \leftarrow \mathit{head}, \mathit{body}$ is acyclic.

Our Main Result

- We define FACQ: the class of freely acyclic conjunctive queries:
  - A conjunctive query of the form $head \leftarrow body$ is freely acyclic if the boolean conjunctive query $() \leftarrow head, body$ is acyclic.
  - FACQ includes acyclic boolean CQs and acyclic strict CQs, but not all acyclic CQs

Our Main Result

- We define FACQ: the class of freely acyclic conjunctive queries:
  - A conjunctive query of the form $head \leftarrow body$ is freely acyclic if the boolean conjunctive query $() \leftarrow head, body$ is acyclic.
  - FACQ includes acyclic boolean CQs and acyclic strict CQs, but not all acyclic CQs

Theorem (Main Result)

FACQs are invariant under guarded simulation. Moreover, a query expressible in FO is invariant under guarded simulation on finite structures if, and only if, it is equivalent in the finite to a union of FACQs.

Intuition of Proof

Why are cyclic strict CQs not invariant under guarded simulations?

Intuition of Proof

Why are cyclic strict CQs not invariant under guarded simulations?

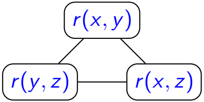
- Consider $\varphi(x, y) \leftarrow r(x, y), r(y, z), r(x, z)$.

Intuition of Proof

Why are cyclic strict CQs not invariant under guarded simulations?

- Consider $\varphi(x, y) \leftarrow r(x, y), r(y, z), r(x, z)$.

Dual graph of φ

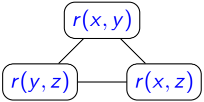


Intuition of Proof

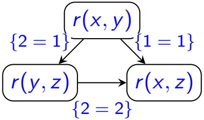
Why are cyclic strict CQs not invariant under guarded simulations?

- Consider $\varphi(x, y) \leftarrow r(x, y), r(y, z), r(x, z)$.

Dual graph of φ



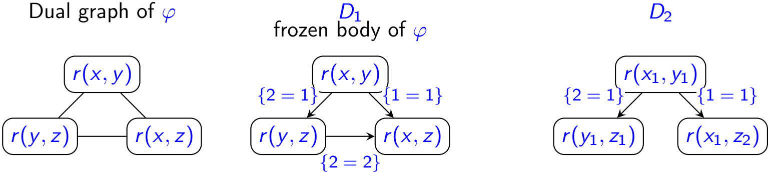
D_1
frozen body of φ



Intuition of Proof

Why are cyclic strict CQs not invariant under guarded simulations?

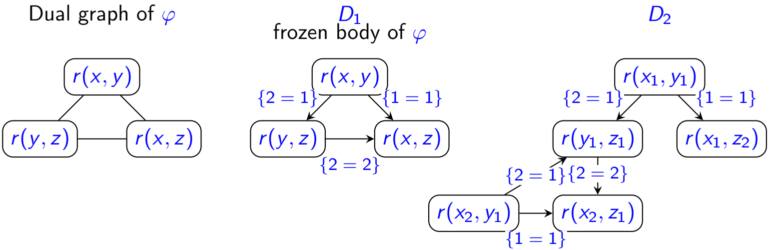
- Consider $\varphi(x, y) \leftarrow r(x, y), r(y, z), r(x, z)$.



Intuition of Proof

Why are cyclic strict CQs not invariant under guarded simulations?

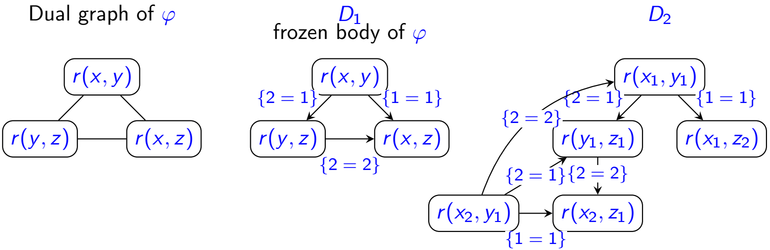
- Consider $\varphi(x, y) \leftarrow r(x, y), r(y, z), r(x, z)$.



Intuition of Proof

Why are cyclic strict CQs not invariant under guarded simulations?

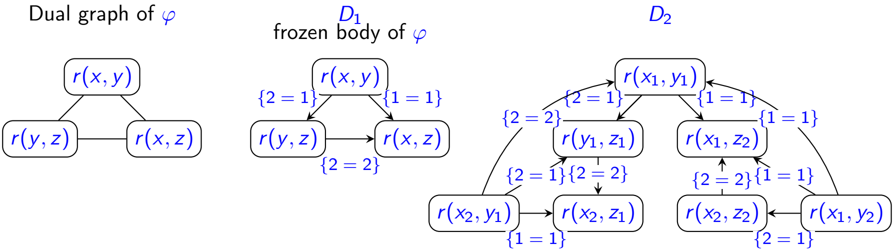
- Consider $\varphi(x, y) \leftarrow r(x, y), r(y, z), r(x, z)$.



Intuition of Proof

Why are cyclic strict CQs not invariant under guarded simulations?

- Consider $\varphi(x, y) \leftarrow r(x, y), r(y, z), r(x, z)$.



Guarded Structural Indexing

- We denote the fact that tuple $\overline{a}$ in db_1 is guarded similar to $\overline{b}$ in db_2 as $db_1, \overline{a} \sim_f db_2, \overline{b}$

Guarded Structural Indexing

- We denote the fact that tuple $\overline{a}$ in db_1 is guarded similar to $\overline{b}$ in db_2 as $db_1, \overline{a} \sim_f db_2, \overline{b}$

Definition (Guarded Simulation Index)

The guarded simulation index for db is a guarded structural index $\mathrm{sim}_g(db) = (db_\downarrow, \mathrm{lab})$ such that:

Guarded Structural Indexing

- We denote the fact that tuple $\overline{a}$ in db_1 is guarded similar to $\overline{b}$ in db_2 as $db_1, \overline{a} \sim_f db_2, \overline{b}$

Definition (Guarded Simulation Index)

The guarded simulation index for db is a guarded structural index $\mathrm{sim}_g(db) = (db_\downarrow, \mathrm{lab})$ such that:

1. $db_\downarrow$ is the smallest database such that for every $t \in db$ there exists a fact $u \in db_\downarrow$ with $db, t \sim_f db_\downarrow, u$.

Guarded Structural Indexing

- We denote the fact that tuple $\overline{a}$ in db_1 is guarded similar to $\overline{b}$ in db_2 as $db_1, \overline{a} \sim_f db_2, \overline{b}$

Definition (Guarded Simulation Index)

The guarded simulation index for db is a guarded structural index $\text{sim}_g(db) = (db_\downarrow, \text{lab})$ such that:

1. $db_\downarrow$ is the smallest database such that for every $t \in db$ there exists a fact $u \in db_\downarrow$ with $db, t \sim_f db_\downarrow, u$.
2. lab is the function that maps each fact $u \in db_\downarrow$ to the set $\{s \in db \mid db, s \sim_f db_\downarrow, u\}$.

Guarded Structural Indexing

- We denote the fact that tuple $\overline{a}$ in db_1 is guarded similar to $\overline{b}$ in db_2 as $db_1, \overline{a} \sim_f db_2, \overline{b}$

Definition (Guarded Simulation Index)

The guarded simulation index for db is a guarded structural index $\mathrm{sim}_g(db) = (db_\downarrow, \mathrm{lab})$ such that:

1. $db_\downarrow$ is the smallest database such that for every $t \in db$ there exists a fact $u \in db_\downarrow$ with $db, t \sim_f db_\downarrow, u$.
2. lab is the function that maps each fact $u \in db_\downarrow$ to the set $\{s \in db \mid db, s \sim_f db_\downarrow, u\}$.

- This indeed can be shown to be a cover for strict ACQs, i.e., if these are evaluated on $\mathrm{sim}_g(db)$ then from the lab of the retrieved nodes we get the query result up to projection.

Approximate Simulations

- In practice it sometimes happens that in a database most tuples/nodes are only similar to themselves.

Approximate Simulations

- In practice it sometimes happens that in a database most tuples/nodes are only similar to themselves.
- In that case we can use instead an approximate simulation relation that considers only the neighbourhood of nodes within a distance k

Approximate Simulations

- In practice it sometimes happens that in a database most tuples/nodes are only similar to themselves.
- In that case we can use instead an approximate simulation relation that considers only the neighbourhood of nodes within a distance k
  - The fact that tuple $\overline{a}$ in db_1 is k-simulated by $\overline{b}$ in db_2 is denoted as $db_1, \overline{a} \preceq_f^k db_2, \overline{b}$

Approximate Simulations

- In practice it sometimes happens that in a database most tuples/nodes are only similar to themselves.
- In that case we can use instead an approximate simulation relation that considers only the neighbourhood of nodes within a distance k
  - The fact that tuple $\overline{a}$ in db_1 is k-simulated by $\overline{b}$ in db_2 is denoted as $db_1, \overline{a} \preceq_f^k db_2, \overline{b}$
- Has an interesting relationship with the height of queries, if this is defined for query $head \leftarrow body$ as the minimum height of all join trees for $() \leftarrow head, body$ that are rooted at $head$.

Approximate Simulations

- In practice it sometimes happens that in a database most tuples/nodes are only similar to themselves.
- In that case we can use instead an approximate simulation relation that considers only the neighbourhood of nodes within a distance k
  - The fact that tuple $\overline{a}$ in db_1 is k-simulated by $\overline{b}$ in db_2 is denoted as $db_1, \overline{a} \preceq_f^k db_2, \overline{b}$
- Has an interesting relationship with the height of queries, if this is defined for query $head \leftarrow body$ as the minimum height of all join trees for $() \leftarrow head, body$ that are rooted at $head$.

Proposition

Let $k \geq 0$ be a natural number. The following are equivalent.

(1) $db_1, \overline{a} \preceq_f^k db_2, \overline{b}$

(2) *For all FACQs Q of height $\leq k$, if $\overline{a} \in Q(db_1)$ then $\overline{b} \in Q(db_2)$.*

Conclusions and Further Research

- Results:

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - ★ Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - ★ Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.
  - Accompanying results for structural indexes based on this characterization.

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.
  - Accompanying results for structural indexes based on this characterization.
- Further research:

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.
  - Accompanying results for structural indexes based on this characterization.
- Further research:
  - Efficient algorithms for computing and maintaining indexes on large real-world databases.

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - ★ Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.
  - Accompanying results for structural indexes based on this characterization.
- Further research:
  - Efficient algorithms for computing and maintaining indexes on large real-world databases.
  - Investigate evaluation strategies that profit from these indexes.

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.
  - Accompanying results for structural indexes based on this characterization.
- Further research:
  - Efficient algorithms for computing and maintaining indexes on large real-world databases.
  - Investigate evaluation strategies that profit from these indexes.
  - Extend characterisation for other relaxations of GF such as the loosely guarded fragment.

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - ★ Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.
  - Accompanying results for structural indexes based on this characterization.
- Further research:
  - Efficient algorithms for computing and maintaining indexes on large real-world databases.
  - Investigate evaluation strategies that profit from these indexes.
  - Extend characterisation for other relaxations of GF such as the loosely guarded fragment.
  - Acyclicity is known to be generalizable to hypertree decompositions; can our results be similarly extended?

Conclusions and Further Research

- Results:
  - Structural characterization of query invariance for strict acyclic conjunctive queries.
    - Plus a characterization of the guarded simulation invariant fragment of FO, in analogy to results of Andréka et al. for guarded bisimilar FO, and Rossman for homomorphically invariant FO.
  - Accompanying results for structural indexes based on this characterization.
- Further research:
  - Efficient algorithms for computing and maintaining indexes on large real-world databases.
  - Investigate evaluation strategies that profit from these indexes.
  - Extend characterisation for other relaxations of GF such as the loosely guarded fragment.
  - Acyclicity is known to be generalizable to hypertree decompositions; can our results be similarly extended?

Thank You