View-based query determinacy and rewritings over graph databases

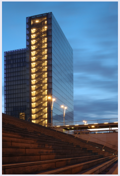
Nadime Francis

LSV at ENS-Cachan, Inria

Tuesday, December, 8th

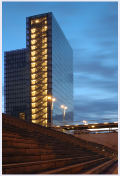
Introduction

Bibliothèque François Mitterand



- 30 000 users
- 11 million documents
- 2000 new documents per day

Bibliothèque François Mitterand



- 30 000 users
- 11 million documents
- 2000 new documents per day

On the Internet

- 1000 billion web pages
- Facebook: 55 million updates per day
- Twitter: 500 million tweets per day
- Google: 3.5 billion queries per day

Databases

- Store data
- Retrieve data / answer queries
- Provide services

Databases

- Store data
- Retrieve data / answer queries
- Provide services

Databases

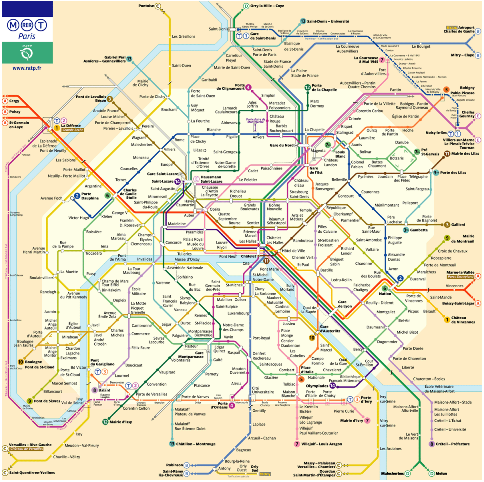
- Store data
- Retrieve data / answer queries
- Provide services

Graph databases

- Nodes: individual data
- Edges: links between data

Motivations

- Data naturally presented as graphs:
  $\rightarrow$ Social networks, Internet, crime detection, semantic Web.
- Queries: paths, cycles, connected components



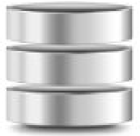
Queries

- How can I go from Bagneux to Porte d'Orléans?
- Does any trip require more than 3 changes?
- What stations are closest to both my wife's and my work places?

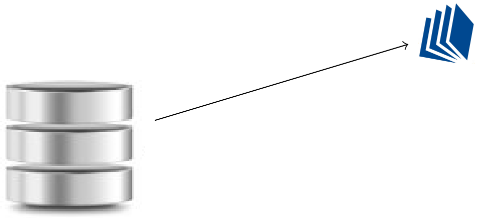
Queries

- Who are the friends of my friends with whom I share a common interest?
- Who are my friends that know each other?
- How small is the world?

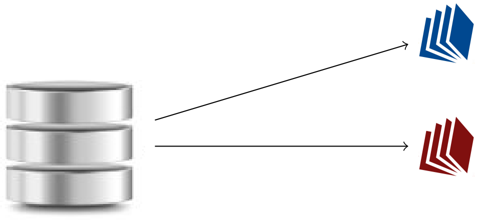
View-based query processing



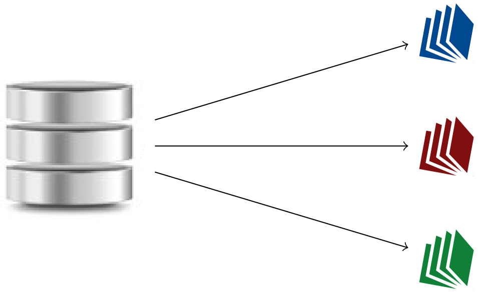
View-based query processing



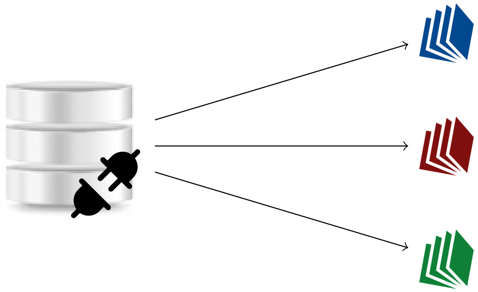
View-based query processing



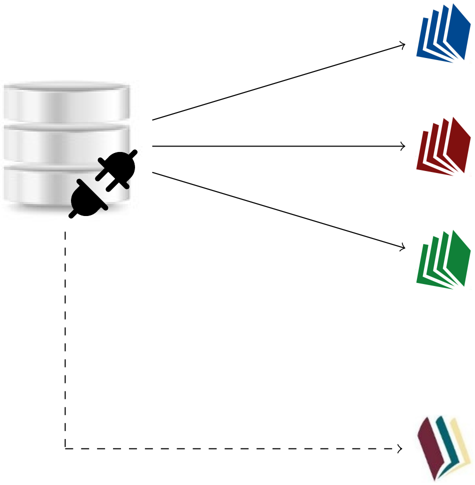
View-based query processing



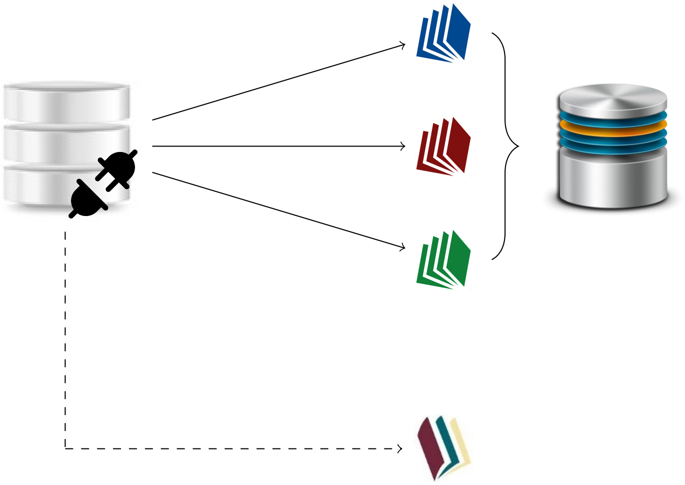
View-based query processing



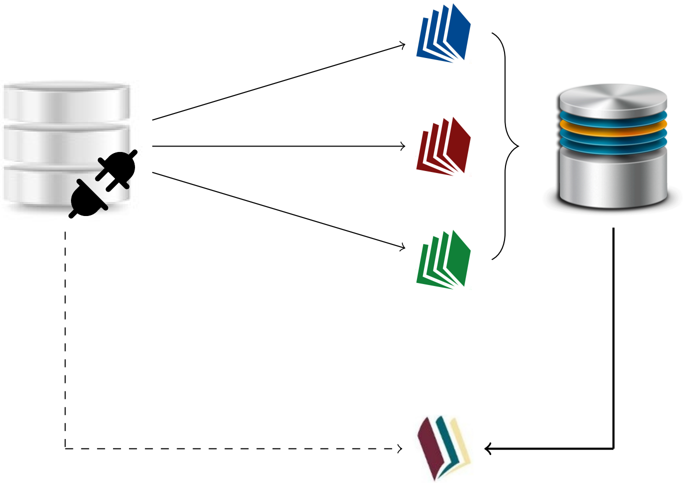
View-based query processing



View-based query processing

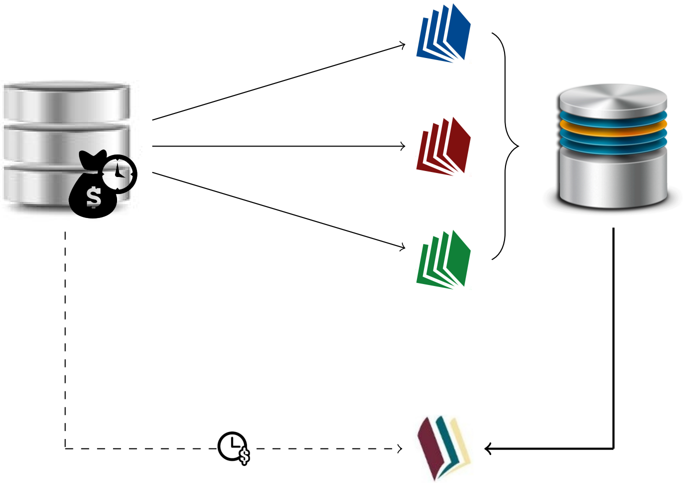


View-based query processing



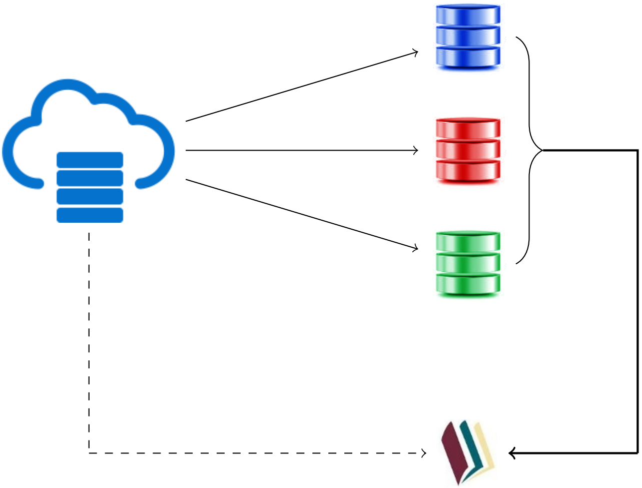
View-based query processing

Application: Query optimization and caching



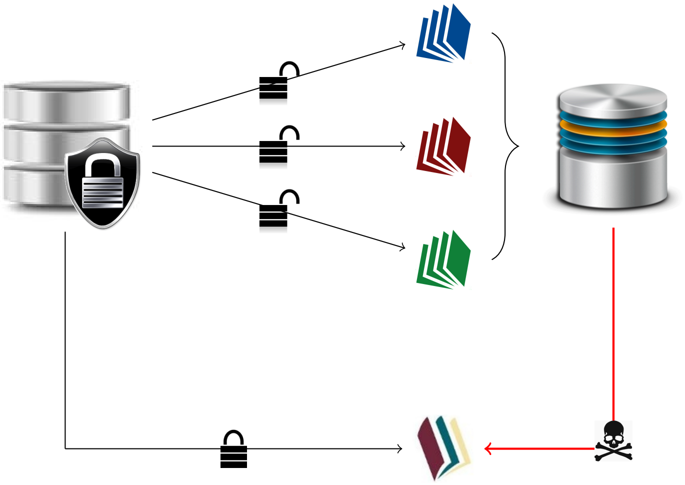
View-based query processing

Application: Data integration

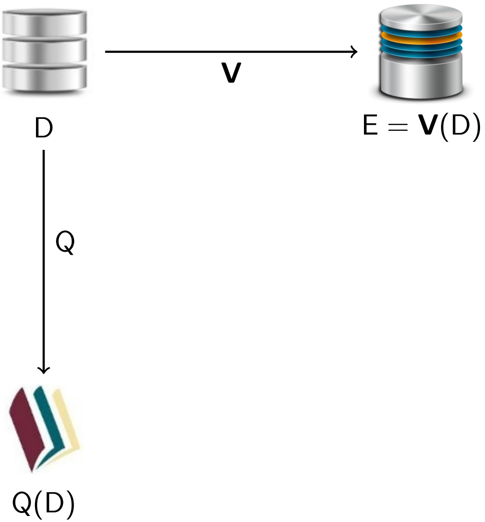


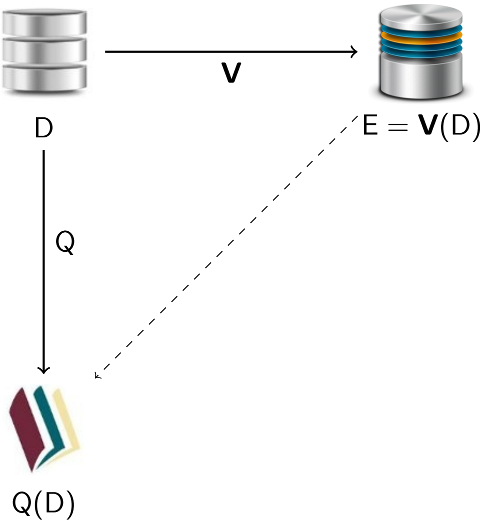
View-based query processing

Application: Privacy and security

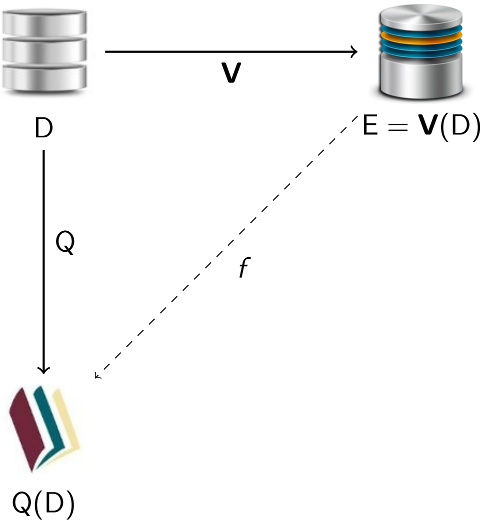


DETERMINACY AND QUERY REWRITING

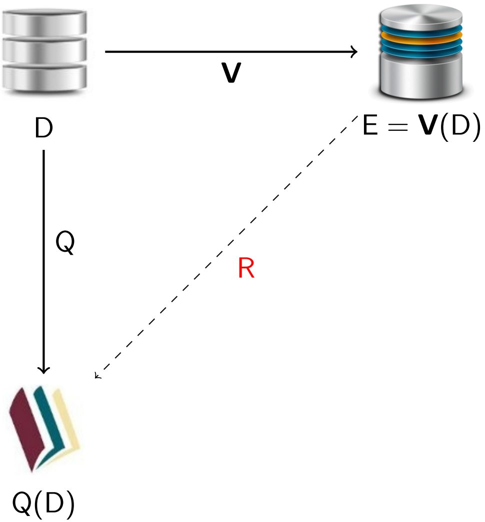

V
D
E = V(D)
Q
Q(D)



1. Can I get Q(D) from E?



1. Can I get Q(D) from E?
2. Does it work for all D?



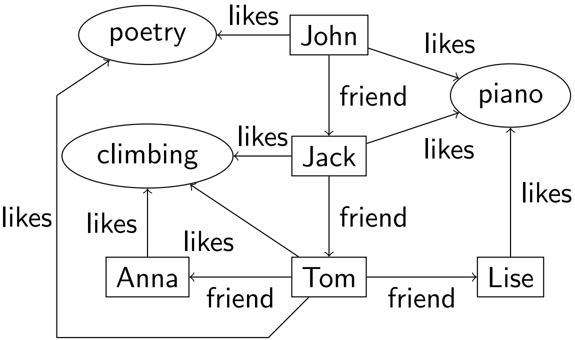
1. Can I get Q(D) from E?
2. Does it work for all D?
3. If so, can I compute R? (in what language?)

Setting

- D: Graph databases
- Q: Regular Path Queries (RPQ)
- **V**: Regular Path Views
- E: View instances and view images

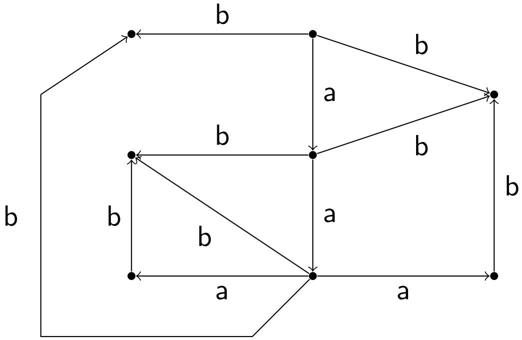
Setting

- D: **Graph databases**
- Q: Regular Path Queries (RPQ)
- **V**: Regular Path Views
- E: View instances and view images



Setting

- D: **Graph databases**
- Q: Regular Path Queries (RPQ)
- **V**: Regular Path Views
- E: View instances and view images



Setting

- D: Graph databases
- Q: **Regular Path Queries (RPQ)**
- **V**: Regular Path Views
- E: View instances and view images

Queries $\mathrm{Q} = \langle L \rangle$ for some regular language L

$$\mathrm{Q}(\mathrm{D}) = \{(x, y) \in \mathrm{D} \mid x \xrightarrow{w} y \text{ in } \mathrm{D}, \text{ with } w \in L\}$$

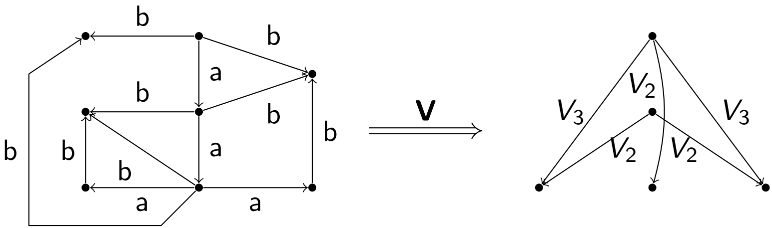
Ex: $\mathrm{Q} = \langle a^* \rangle$, $\mathrm{Q} = \langle a^+ b \rangle$
Special cases:

- Single path queries (SPQ): $\mathrm{Q} = \langle w \rangle$, $\mathrm{Q}_i = \langle a^i \rangle$
- Unions of single path queries (UPQ): $\mathrm{Q} = \langle w_1, \ldots, w_n \rangle$

Setting

- D: Graph databases
- Q: Regular Path Queries (RPQ)
- **V**: **Regular Path Views**
- E: **View instances and view images**

View **V**: set of queries.
Ex: $V_2 = \langle a^2 \rangle$, $V_3 = \langle a^3 \rangle$, $\mathbf{V} = \{V_2, V_3\}$



Setting

- D: Graph databases
- Q: Regular Path Queries (RPQ)
- $\mathbf{V}$: Regular Path Views
- E: View instances and view images

Determinacy

- $\forall \mathrm{D}, \mathrm{D}' \qquad \mathbf{V}(\mathrm{D}) = \mathbf{V}(\mathrm{D}') \;\Rightarrow\; \mathrm{Q}(\mathrm{D}) = \mathrm{Q}(\mathrm{D}')$
- $\mathrm{Q}, \mathbf{V} \rightsquigarrow f$: function induced by Q using $\mathbf{V}$

Setting

- D: Graph databases
- Q: Regular Path Queries (RPQ)
- $\mathbf{V}$: Regular Path Views
- E: View instances and view images

Determinacy

- $\forall \mathrm{D}, \mathrm{D}' \qquad \mathbf{V}(\mathrm{D}) = \mathbf{V}(\mathrm{D}') \;\Rightarrow\; \mathrm{Q}(\mathrm{D}) = \mathrm{Q}(\mathrm{D}')$
- $\mathrm{Q}, \mathbf{V} \rightsquigarrow f$: function induced by Q using $\mathbf{V}$

Rewriting

- Query R such that
  $\forall \mathrm{D} \qquad \mathrm{R}(\mathbf{V}(\mathrm{D})) = f(\mathbf{V}(\mathrm{D}))$

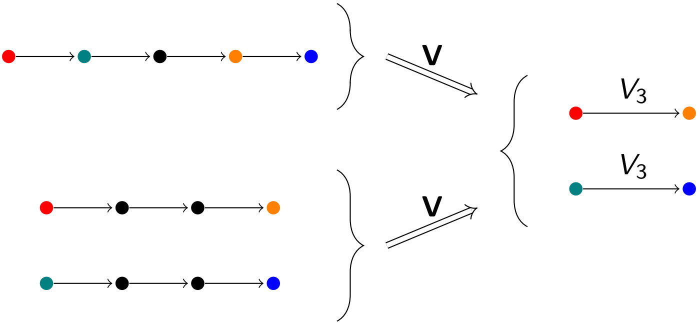
Example 1

$$V_3 = \langle a^3 \rangle \qquad Q_4 = \langle a^4 \rangle$$

Example 1

$$V_3 = \langle a^3 \rangle \qquad Q_4 = \langle a^4 \rangle$$

The view does not determine the query.



Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{\mathrm{R}(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

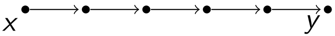
$Q \Rightarrow R$

Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$$Q \Rightarrow R$$



Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{\mathrm{R}(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

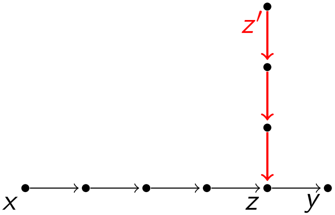
$\mathrm{Q} \Rightarrow \mathrm{R}$

Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$$Q \Rightarrow R$$

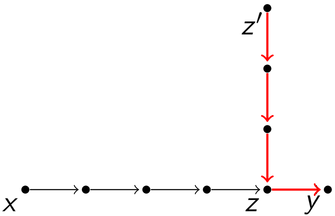


Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{\mathrm{R}(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$$\mathrm{Q} \Rightarrow \mathrm{R}$$

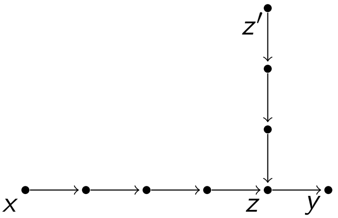


Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x,y) = \exists z, V_4(x,z) \wedge (\forall z' V_3(z',z) \Rightarrow V_4(z',y))}$$

$Q \Rightarrow R$ ✓

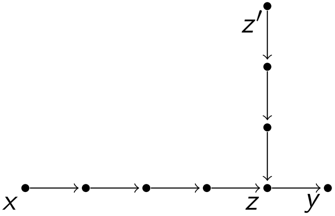


Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$Q \Rightarrow R$ ✓ $\qquad\qquad$ $R \Rightarrow Q$

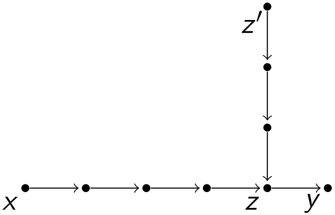


Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x,y) = \exists z, V_4(x,z) \wedge (\forall z'\, V_3(z',z) \Rightarrow V_4(z',y))}$$

$Q \Rightarrow R$ ✓ $\qquad\qquad$ $R \Rightarrow Q$

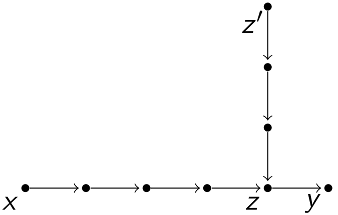


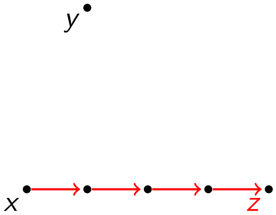
Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$Q \Rightarrow R$ ✓ $\qquad\qquad R \Rightarrow Q$





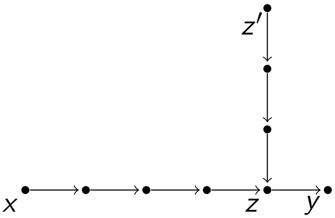
Example 2 [Afrati]

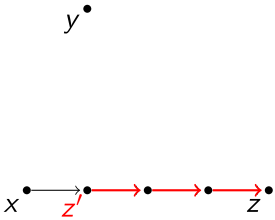
$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$Q \Rightarrow R$ ✓

$R \Rightarrow Q$





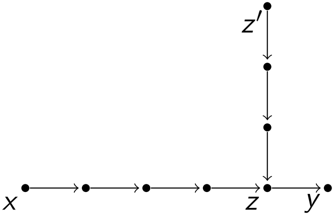
Example 2 [Afrati]

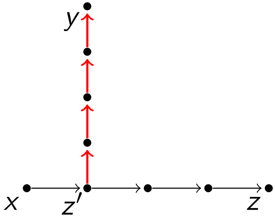
$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x,y) = \exists z, V_4(x,z) \wedge (\forall z' V_3(z',z) \Rightarrow V_4(z',y))}$$

$Q \Rightarrow R$ ✓

$R \Rightarrow Q$



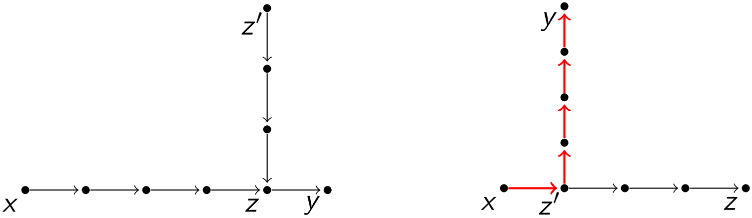


Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$Q \Rightarrow R$ ✓ $\qquad\qquad$ $R \Rightarrow Q$

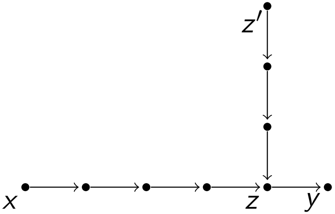


Example 2 [Afrati]

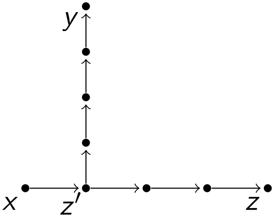
$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{R(x,y) = \exists z, V_4(x,z) \wedge (\forall z' V_3(z',z) \Rightarrow V_4(z',y))}$$

$Q \Rightarrow R$ ✓



$R \Rightarrow Q$ ✓



Example 2 [Afrati]

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q_5 = \langle a^5 \rangle$$

$$\boxed{\mathrm{R}(x, y) = \exists z, V_4(x, z) \wedge (\forall z' V_3(z', z) \Rightarrow V_4(z', y))}$$

$$\boxed{\mathrm{R}'(x, y) = \exists z, t, V_4(x, z) \wedge V_3(x, t) \wedge \forall z' V_3(z', z) \Rightarrow V_4(z', y)}$$

R′ is also a rewriting :

$$\forall \mathrm{D}, \mathrm{R}(\mathbf{V}(\mathrm{D})) = \mathrm{R}'(\mathbf{V}(\mathrm{D}))$$

R and R′ behave differently outside of view images :

$$\exists \mathrm{E}, \mathrm{R}(\mathrm{E}) \neq \mathrm{R}'(\mathrm{E})$$

Certain answers [Abiteboul, Duschka]

$$cert^{exact}_{Q,\mathbf{V}}(E) = \bigcap_{D \mid \mathbf{V}(D) = E} Q(D) \qquad \text{(exact views)}$$

$$cert^{sound}_{Q,\mathbf{V}}(E) = \bigcap_{D \mid \mathbf{V}(D) \supseteq E} Q(D) \qquad \text{(sound views)}$$

Certain answers [Abiteboul, Duschka]

$$cert^{exact}_{Q,\mathbf{V}}(E) = \bigcap_{D \mid \mathbf{V}(D) = E} Q(D) \qquad \text{(exact views)}$$

$$cert^{sound}_{Q,\mathbf{V}}(E) = \bigcap_{D \mid \mathbf{V}(D) \supseteq E} Q(D) \qquad \text{(sound views)}$$

Remarks

- If $\mathbf{V}$ determines Q, then $cert^{exact}_{Q,\mathbf{V}}$ is a rewriting.
- $cert_{Q,\mathbf{V}}(\cdot)$ has coNP-complete data complexity.

Certain answers [Abiteboul, Duschka]

$$cert^{exact}_{Q,\mathbf{V}}(E) = \bigcap_{D \mid \mathbf{V}(D) = E} Q(D) \qquad \text{(exact views)}$$

$$cert^{sound}_{Q,\mathbf{V}}(E) = \bigcap_{D \mid \mathbf{V}(D) \supseteq E} Q(D) \qquad \text{(sound views)}$$

Remarks

- If $\mathbf{V}$ determines Q, then $cert^{exact}_{Q,\mathbf{V}}$ is a rewriting.
- $cert_{Q,\mathbf{V}}(\cdot)$ has coNP-complete data complexity.

Theorem [F., Segoufin, Sirangelo]

If an RPQ view $\mathbf{V}$ determines an RPQ query Q, then there exist a rewriting of Q using $\mathbf{V}$ with NP data complexity, and one with coNP data complexity.

$\rightarrow$ $cert^{exact}_{Q,\mathbf{V}}$ is likely not optimal!

Positive and negative results

- [Afrati]: Determinacy is decidable for single path queries and views. Rewritings are in FO.
- [Calvanese, De Giacomo, Lenzerini, Vardi]: Monotone determinacy is decidable for RPQ views and queries.
- [Nash, Segoufin, Vianu]: Determinacy is undecidable for UCQ views and queries.
- [F., Segoufin, Sirangelo]: Determinacy is undecidable for CRPQ views and RPQ queries.

Open questions

- Decidability is still open for two key query languages:
  - Conjunctive queries and views. (WIP - [Marcinkowski et al.])
  - Regular path queries and views.

Main contributions

Theorem [ICDT 2014]

If an RPQ view **V** determines an RPQ query Q in a monotone way, then there exists a Datalog rewriting of Q using **V**.

$\rightarrow$ Cor: there exists a rewriting with PTime data complexity.
$\rightarrow$ Cor: given **V** and Q, it is decidable whether there exists a Datalog rewriting of Q using **V**.

Main contributions

Theorem [ICDT 2014]

If an RPQ view **V** determines an RPQ query Q in a monotone way, then there exists a Datalog rewriting of Q using **V**.

→ Cor: there exists a rewriting with PTime data complexity.
→ Cor: given **V** and Q, it is decidable whether there exists a Datalog rewriting of Q using **V**.

Theorem [ICDT 2015, Best student paper]

The asymptotic determinacy problem is decidable for UPQ views and SPQ queries over graphs, and results in first-order rewritings.

→ Cor: first-order is almost complete for rewritings of SPQ queries using UPQ views.

Monotone Determinacy

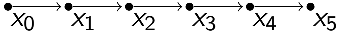
Example 2 revisited

$$V_3 = \langle a^3 \rangle \qquad V_4 = \langle a^4 \rangle \qquad Q = \langle a^5 \rangle$$

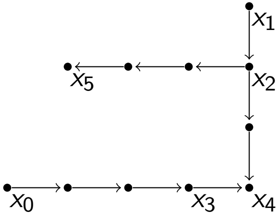
$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ but } f \text{ is not monotone.}}$

$\mathrm{Q}(\mathrm{D}) = \{(x_0, x_5)\}$

$\mathbf{V}(\mathrm{D}) \subseteq \mathbf{V}(\mathrm{D}') \text{ but } \mathrm{Q}(\mathrm{D}') = \emptyset$



D



D′

Monotone determinacy

- $\mathbf{V} \twoheadrightarrow \mathrm{Q}$ and f is monotone.
- Equivalently:
  $\forall \mathrm{D}, \mathrm{D}' \qquad \mathbf{V}(\mathrm{D}) \subseteq \mathbf{V}(\mathrm{D}') \;\Rightarrow\; \mathrm{Q}(\mathrm{D}) \subseteq \mathrm{Q}(\mathrm{D}')$

Motivations

- Many interesting languages are monotone: conjunctive queries, regular path queries, Datalog, etc.
- [Nash, Segoufin, Vianu]: for CQ views and queries, monotone determinacy $\Rightarrow$ CQ rewriting.

Key observation

If $\mathbf{V}$ determines Q in a monotone way, then $cert^{sound}_{\mathrm{Q},\mathbf{V}}$ is a rewriting.

$\rightarrow$ Idea: look for approximations of $cert^{sound}_{\mathrm{Q},\mathbf{V}}$.

Constraint Satisfaction Problems as Queries

- Fix a template structure T with source and target nodes
- $(\mathrm{E}, u, v) \in \mathrm{CSP}(T)$ iff there exists h such that
  - $\mathrm{E} \xrightarrow{h} T$
  - $h(u)$ is a source node
  - $h(v)$ is a target node
- $\mathrm{CSP}(T)$ selects all u, v such that $(\mathrm{E}, u, v) \in \mathrm{CSP}(T)$

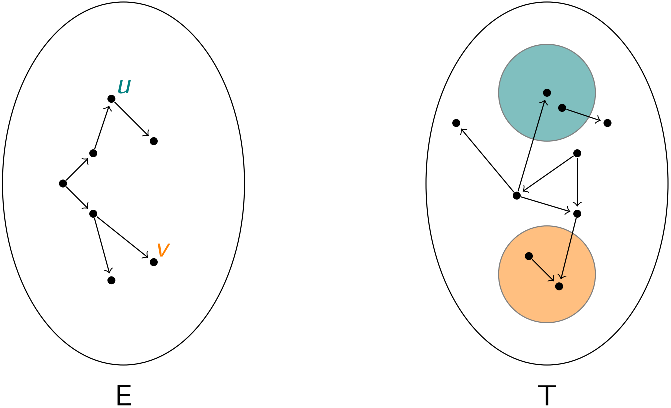
Theorem [Calvanese, De Giacomo, Lenzerini, Vardi]

There exists a structure $T_{\mathrm{Q},\mathbf{V}}$ such that, for all D,

$$(u, v) \in cert^{sound}_{\mathrm{Q},\mathbf{V}}(\mathbf{V}(\mathrm{D})) \Leftrightarrow (\mathbf{V}(\mathrm{D}), u, v) \notin \mathrm{CSP}(T_{\mathrm{Q},\mathbf{V}})$$

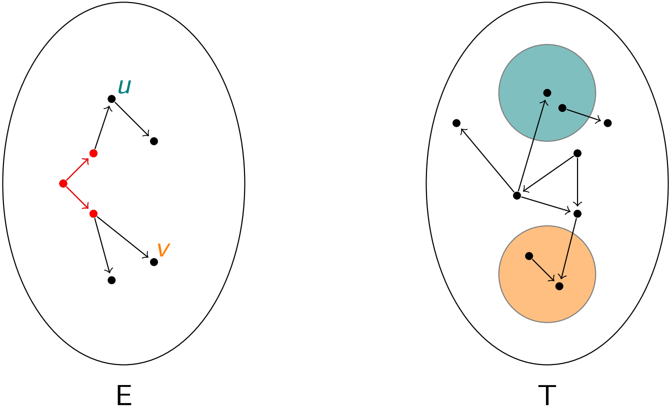
A CSP approximation: the (ℓ, k)-two-player-game

Arena : (E, u, v) and T with source and target nodes.



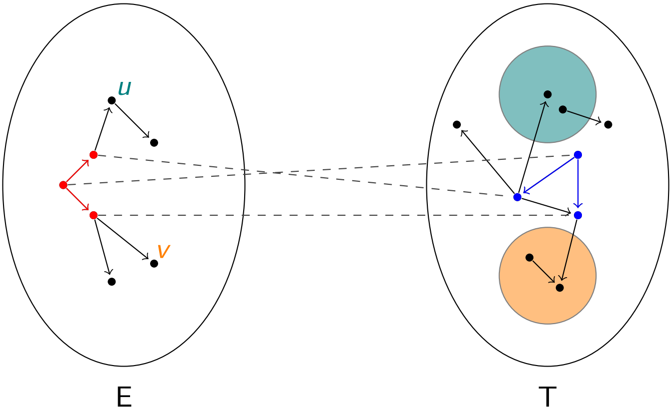
A CSP approximation: the (ℓ, k)-two-player-game

Player 1 starts by selecting k nodes on E.



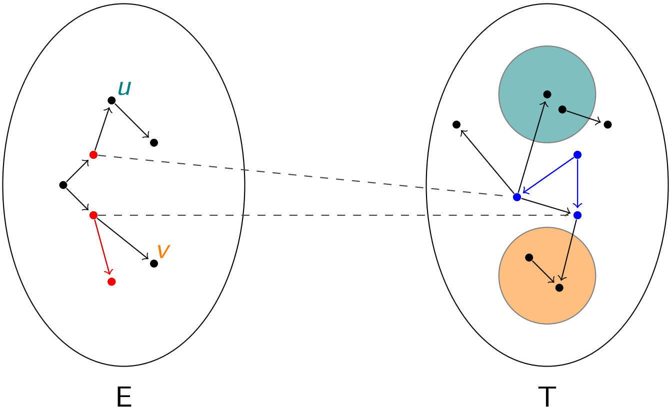
A CSP approximation: the (ℓ, k)-two-player-game

Player 2 provides a homomorphism from those nodes to T.



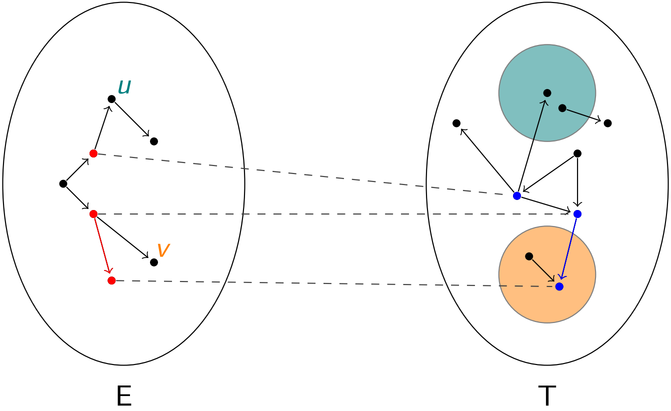
A CSP approximation: the (ℓ, k)-two-player-game

Player 1 selects k nodes, keeping at most ℓ previous nodes.



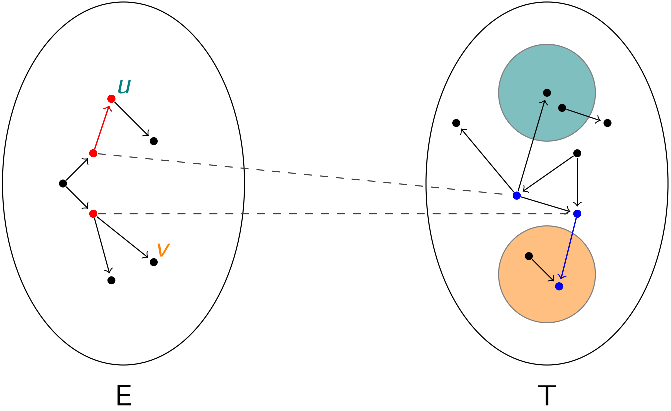
A CSP approximation: the (ℓ, k)-two-player-game

Player 2 extends previous homomorphism to new nodes.



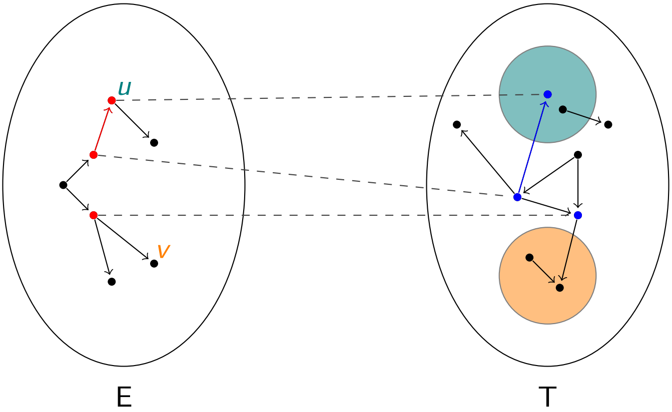
A CSP approximation: the (ℓ, k)-two-player-game

If Player 1 selects u...



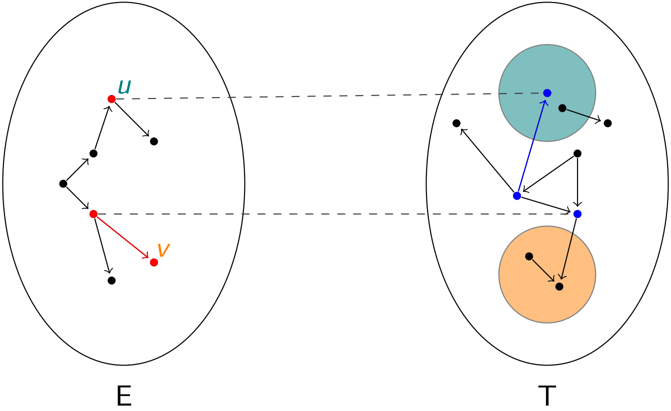
A CSP approximation: the (ℓ, k)-two-player-game

Then Player 2 must send it to a source node.



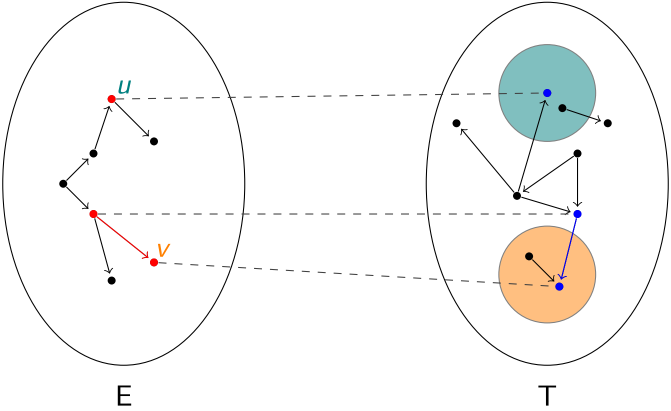
A CSP approximation: the (ℓ, k)-two-player-game

Similarly, if Player 1 selects v...



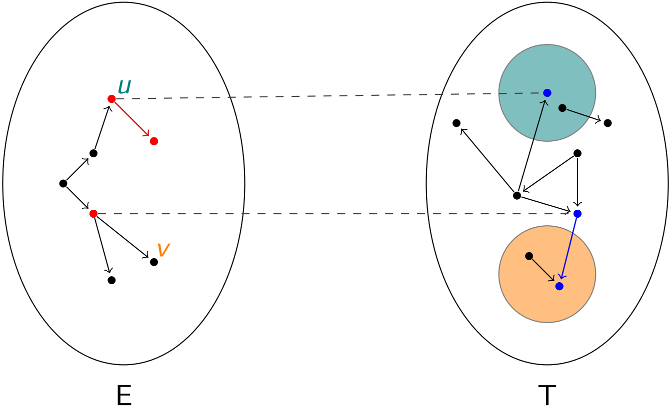
A CSP approximation: the (ℓ, k)-two-player-game

Then Player 2 must send it to a target node.



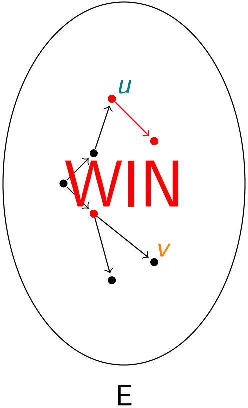
A CSP approximation: the (ℓ, k)-two-player-game

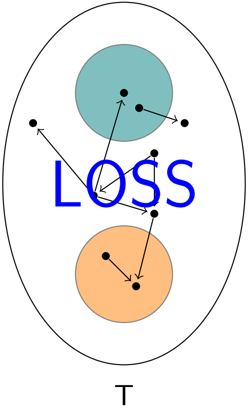
If Player 2 has no legal move...



A CSP approximation: the (ℓ, k)-two-player-game

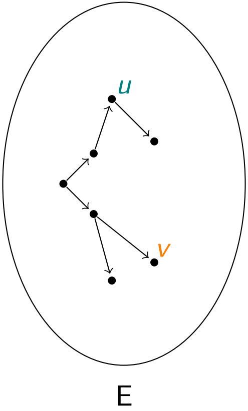
Then Player 1 wins.

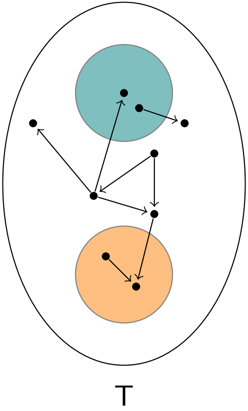




A CSP approximation: the (ℓ, k)-two-player-game

Player 2 wins if she can play forever.





Approximations

- The game on $T_{\mathsf{Q},\mathbf{V}}$ over-approximates $\mathsf{CSP}(T_{\mathsf{Q},\mathbf{V}})$.
  $\rightarrow$ If $(E, u, v) \in \mathsf{CSP}(T_{\mathsf{Q},\mathbf{V}})$, then P2 has a winning strategy.
- The game on $T_{\mathsf{Q},\mathbf{V}}$ under-approximates $cert^{sound}_{\mathsf{Q},\mathbf{V}}$.
  $\rightarrow$ If P1 has a winning strategy, then $(u, v) \in cert^{sound}_{\mathsf{Q},\mathbf{V}}(E)$.

Theorem [Feder, Vardi]

There exists a $\mathsf{Datalog}_{\ell,k}$ program $Q_{\ell,k}$ such that $(u, v) \in Q_{\ell,k}(\mathrm{E})$ if and only if Player 1 has a winning strategy for the game played on (E, u, v) and $T_{\mathsf{Q},\mathbf{V}}$.

The case of simple paths

- $\exists \ell, k$ such that $Q_{\ell,k}$ is exact on views of simple paths.
- $(u, v) \in Q_{\ell,k}(\mathbf{V}(\pi))$ if and only if $(u, v) \in cert^{sound}_{\mathrm{Q},\mathbf{V}}(\mathbf{V}(\pi))$.
- This is the technical part of the proof!

Lifting paths to arbitrary databases

Assume that some candidate query R is:

- a rewriting on views of simple paths,
- an under-approximation of a rewriting,
- closed under homomorphism.

Then it is a rewriting on views of arbitrary databases.

$$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ and } f \nearrow}$$

$$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ and } f \nearrow}$$

$$\vdots$$

$$\boxed{\text{Certain answers are a (coNP-hard) rewriting.}}$$

$$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ and } f \nearrow}$$

$\vdots$

$$\boxed{\text{Certain answers are a (coNP-hard) rewriting.}}$$

$\vdots$

$$\boxed{\text{Certain answers} \equiv \neg\text{CSP}}$$ [Calvanese et al.]

$$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ and } f \nearrow}$$

$\vdots$

$$\boxed{\text{Certain answers are a (coNP-hard) rewriting.}}$$

$\vdots$

$$\boxed{\text{Certain answers} \equiv \neg\mathrm{CSP}}$$

[Calvanese et al.]

$\vdots$

$$\boxed{\neg\mathrm{CSP} \sim \mathrm{Q}_{\ell,k} \text{ in Datalog}}$$

[Feder, Vardi]

$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ and } f \nearrow}$

$\vdots$

$\boxed{\text{Certain answers are a (coNP-hard) rewriting.}}$

$\vdots$

$\boxed{\text{Certain answers} \equiv \neg\mathrm{CSP}}$ [Calvanese et al.]

$\vdots$

$\boxed{\neg\mathrm{CSP} \sim \mathrm{Q}_{\ell,k} \text{ in Datalog}}$ [Feder, Vardi]

$\vdots$

$\boxed{\mathrm{Q}_{\ell,k} \text{ is exact on simple paths, for some } \ell, k}$

$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ and } f \nearrow}$

$\vdots$

Certain answers are a (coNP-hard) rewriting.

$\vdots$

Certain answers $\equiv \neg$CSP [Calvanese et al.]

$\vdots$

$\neg$CSP $\sim \mathrm{Q}_{\ell,k}$ in Datalog [Feder, Vardi]

$\vdots$

$\mathrm{Q}_{\ell,k}$ is exact on simple paths, for some ℓ, k

$\vdots$

Paths can be lifted to arbitrary databases.

$\boxed{\mathbf{V} \twoheadrightarrow \mathrm{Q} \text{ and } f \nearrow}$

$\vdots$

Certain answers are a (coNP-hard) rewriting.

$\vdots$

Certain answers $\equiv \neg$CSP [Calvanese et al.]

$\vdots$

$\neg$CSP $\sim \mathrm{Q}_{\ell,k}$ in Datalog [Feder, Vardi]

$\vdots$

$\mathrm{Q}_{\ell,k}$ is exact on simple paths, for some ℓ, k

$\vdots$

Paths can be lifted to arbitrary databases.

$\vdots$

$\mathrm{Q}_{\ell,k}$ is a (PTime) rewriting.

Asymptotic Determinacy

Restricted setting

- Graph databases → Graphs
- Regular path queries → Single path queries
  Notation: $\langle a^k \rangle \to \langle k \rangle$
- Regular path views "→" Union of single path queries
  Notation: $\langle a^i, a^j \rangle \to \langle i, j \rangle$

Restricted setting

- Graph databases $\rightarrow$ Graphs
- Regular path queries $\rightarrow$ Single path queries
  Notation: $\langle a^k \rangle \rightarrow \langle k \rangle$
- Regular path views "$\rightarrow$" Union of single path queries
  Notation: $\langle a^i, a^j \rangle \rightarrow \langle i, j \rangle$

Example 3

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,3} = \langle 2, 3 \rangle$$
$$Q_3 = \langle 3 \rangle \qquad Q_5 = \langle 5 \rangle$$

$\boxed{\mathbf{V} \not\twoheadrightarrow Q_3}$ $\boxed{\mathbf{V} \twoheadrightarrow Q_5}$

Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

By contraposition, if $\forall c, \mathrm{Q}_c \notin \mathbf{V}$, then $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_k$.
Proof by example, with $V_{2,4} \not\twoheadrightarrow \mathrm{Q}_4$.

$$x_0 \to x_1 \to x_2 \to x_3 \to x_4$$

Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

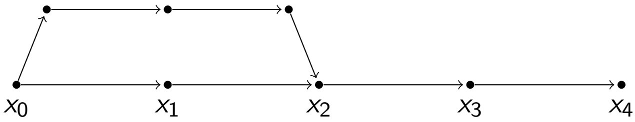
By contraposition, if $\forall c, \mathrm{Q}_c \notin \mathbf{V}$, then $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_k$.
Proof by example, with $V_{2,4} \not\twoheadrightarrow \mathrm{Q}_4$.

$$\begin{tikzcd}
x_0 \arrow[r] \arrow[rr, bend left, "V_{2,4}"] & x_1 \arrow[r] & x_2 \arrow[r] & x_3 \arrow[r] & x_4
\end{tikzcd}$$

Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

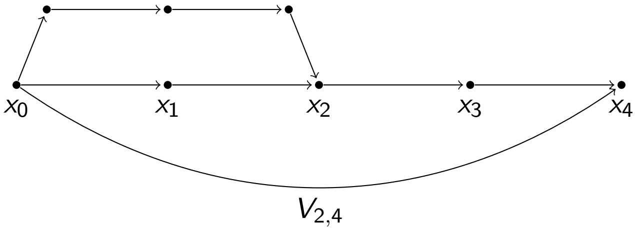
By contraposition, if $\forall c, \mathrm{Q}_c \notin \mathbf{V}$, then $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_k$.
Proof by example, with $V_{2,4} \not\twoheadrightarrow \mathrm{Q}_4$.



Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

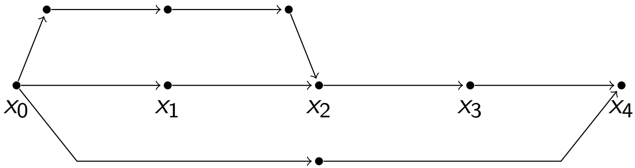
By contraposition, if $\forall c, \mathrm{Q}_c \notin \mathbf{V}$, then $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_k$.
Proof by example, with $V_{2,4} \not\twoheadrightarrow \mathrm{Q}_4$.



Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

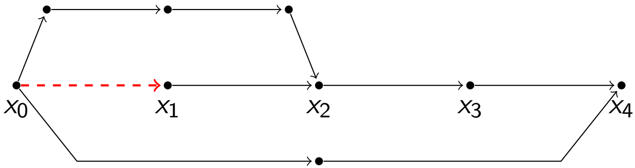
By contraposition, if $\forall c, \mathrm{Q}_c \notin \mathbf{V}$, then $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_k$.
Proof by example, with $V_{2,4} \not\twoheadrightarrow \mathrm{Q}_4$.



Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

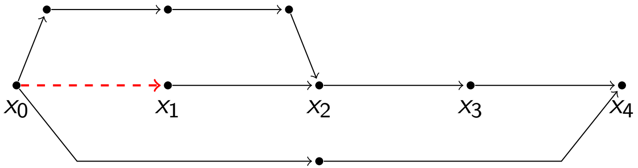
By contraposition, if $\forall c, \mathrm{Q}_c \notin \mathbf{V}$, then $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_k$.
Proof by example, with $V_{2,4} \not\twoheadrightarrow \mathrm{Q}_4$.



Key observation 1

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\exists c, \mathrm{Q}_c \in \mathbf{V}$.

By contraposition, if $\forall c, \mathrm{Q}_c \notin \mathbf{V}$, then $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_k$.
Proof by example, with $V_{2,4} \not\twoheadrightarrow \mathrm{Q}_4$.



Key observation 2

If $\mathbf{V} \twoheadrightarrow \mathrm{Q}_k$ then $\mathbf{V} \twoheadrightarrow \mathrm{Q}_{k+c}$.

Observations 1 & 2 $\rightarrow$ possible behaviors for a view $\mathbf{V}$:

Observations 1 & 2 → possible behaviors for a view **V**:

$\mathbf{V} \twoheadrightarrow^?$	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_c

Observations 1 & 2 $\rightarrow$ possible behaviors for a view $\mathbf{V}$:

$\mathbf{V} \twoheadrightarrow^{?}$	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_c
	Q_{1+c}	Q_{2+c}	Q_{3+c}	Q_{4+c}	Q_{5+c}	$\cdots$	Q_{2c}

Observations 1 & 2 $\rightarrow$ possible behaviors for a view $\mathbf{V}$:

$\mathbf{V} \twoheadrightarrow^{?}$	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_c
	Q_{1+c}	Q_{2+c}	Q_{3+c}	Q_{4+c}	Q_{5+c}	$\cdots$	Q_{2c}
	Q_{1+2c}	Q_{2+2c}	Q_{3+2c}	Q_{4+2c}	Q_{5+2c}	$\cdots$	Q_{3c}

Observations 1 & 2 $\rightarrow$ possible behaviors for a view $\mathbf{V}$:

$\mathbf{V} \twoheadrightarrow^{?}$	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_c
	Q_{1+c}	Q_{2+c}	Q_{3+c}	Q_{4+c}	Q_{5+c}	$\cdots$	Q_{2c}
	Q_{1+2c}	Q_{2+2c}	Q_{3+2c}	Q_{4+2c}	Q_{5+2c}	$\cdots$	Q_{3c}
	Q_{1+3c}	Q_{2+3c}	Q_{3+3c}	Q_{4+3c}	Q_{5+3c}	$\cdots$	Q_{4c}

Observations 1 & 2 $\rightarrow$ possible behaviors for a view $\mathbf{V}$:

$\mathbf{V} \twoheadrightarrow^{?}$	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_c
	Q_{1+c}	Q_{2+c}	Q_{3+c}	Q_{4+c}	Q_{5+c}	$\cdots$	Q_{2c}
	Q_{1+2c}	Q_{2+2c}	Q_{3+2c}	Q_{4+2c}	Q_{5+2c}	$\cdots$	Q_{3c}
	Q_{1+3c}	Q_{2+3c}	Q_{3+3c}	Q_{4+3c}	Q_{5+3c}	$\cdots$	Q_{4c}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
	$Q_{1+\alpha c}$	$Q_{2+\alpha c}$	$Q_{3+\alpha c}$	$Q_{4+\alpha c}$	$Q_{5+\alpha c}$	$\cdots$	$Q_{c+\alpha c}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$

Observations 1 & 2 $\rightarrow$ possible behaviors for a view $\mathbf{V}$:

$\mathbf{V} \twoheadrightarrow^{?}$	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_c
	Q_{1+c}	Q_{2+c}	Q_{3+c}	Q_{4+c}	Q_{5+c}	$\cdots$	Q_{2c}
	Q_{1+2c}	Q_{2+2c}	Q_{3+2c}	Q_{4+2c}	Q_{5+2c}	$\cdots$	Q_{3c}
	Q_{1+3c}	Q_{2+3c}	Q_{3+3c}	Q_{4+3c}	Q_{5+3c}	$\cdots$	Q_{4c}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$
	$Q_{1+\alpha c}$	$Q_{2+\alpha c}$	$Q_{3+\alpha c}$	$Q_{4+\alpha c}$	$Q_{5+\alpha c}$	$\cdots$	$Q_{c+\alpha c}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$

Asymptotic determinacy problem

Given $\mathbf{V}$, produce full picture at some line α past stabilization.

Main theorem restated

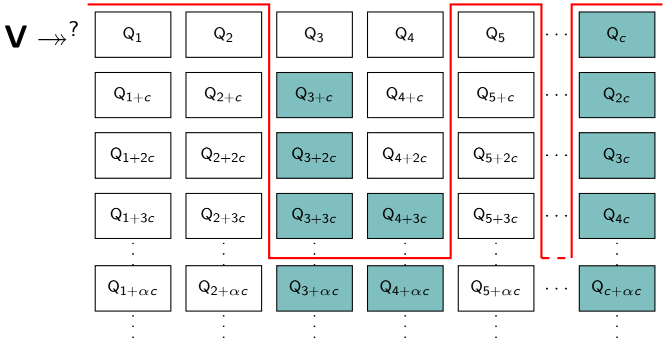
Given a UPQ view **V**, we can:

- Compute some α past stabilization;
- Produce first-order rewritings for determined queries after α;
- Produce counter-examples for all other queries after α.
  - Note that these work also before α, due to Obs. 2.

Main theorem restated

Given a UPQ view **V**, we can:

- Compute some α past stabilization;
- Produce first-order rewritings for determined queries after α;
- Produce counter-examples for all other queries after α.
  - Note that these work also before α, due to Obs. 2.



Example 4

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

Example 4

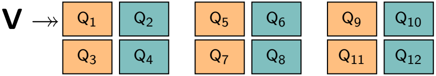
$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

$\mathbf{V} \twoheadrightarrow$

Q_1	Q_2	Q_5	Q_6	Q_9	Q_{10}
Q_3	Q_4	Q_7	Q_8	Q_{11}	Q_{12}

Example 4

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$



- $\mathbf{V} \twoheadrightarrow Q_{2k}$ for all k.

Example 4

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

- $\mathbf{V} \twoheadrightarrow Q_{2k}$ for all k.
- $\mathbf{V} \twoheadrightarrow Q_{11}$.

Example 4

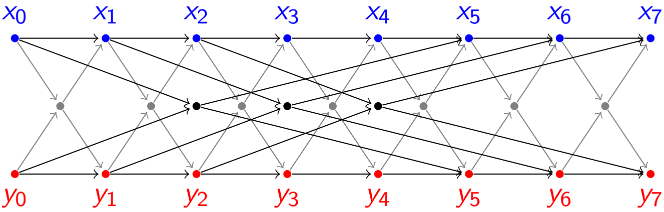
$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

- $\mathbf{V} \twoheadrightarrow Q_{2k}$ for all k.
- $\mathbf{V} \twoheadrightarrow Q_{11}$.
- $\mathbf{V} \not\twoheadrightarrow Q_1, Q_3, Q_5, Q_7$.

Example 4

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

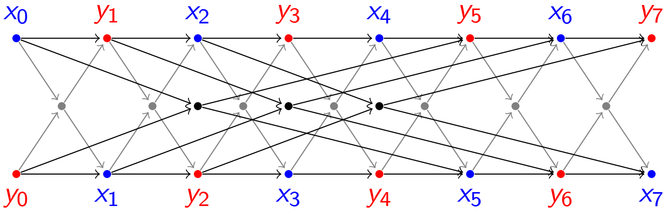
- $\mathbf{V} \twoheadrightarrow Q_{2k}$ for all k.
- $\mathbf{V} \twoheadrightarrow Q_{11}$.
- $\mathbf{V} \not\twoheadrightarrow Q_1, Q_3, Q_5, Q_7$.



Example 4

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

- $\mathbf{V} \twoheadrightarrow \mathrm{Q}_{2k}$ for all k.
- $\mathbf{V} \twoheadrightarrow \mathrm{Q}_{11}$.
- $\mathbf{V} \not\twoheadrightarrow \mathrm{Q}_1, \mathrm{Q}_3, \mathrm{Q}_5, \mathrm{Q}_7$.



Example 4

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

- What about Q_9?

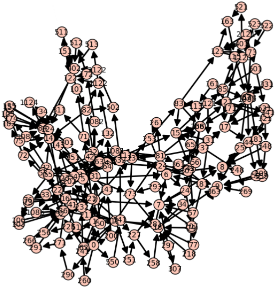
Example 4

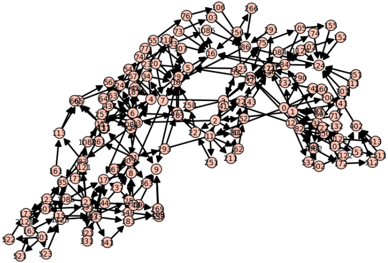
$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,5} = \langle 2, 5 \rangle$$

$\mathbf{V} \twoheadrightarrow$

Q_1	Q_2	Q_5	Q_6	Q_9	Q_{10}
Q_3	Q_4	Q_7	Q_8	Q_{11}	Q_{12}

- What about Q_9?
  - $\mathbf{V} \not\twoheadrightarrow Q_9$!





$\boxed{\text{PERSPECTIVES}}$

Questions left open

On regular path queries and regular path views

- Monotone determinacy is decidable.
- What about general determinacy?
  $\rightarrow$ Decidability: unknown.
  $\rightarrow$ Answering: best known bounds: NP$\cap$coNP.

On unions of single path queries

- Asymptotic determinacy is decidable for UPQ to SPQ.
- What about small queries?
- What about labelled graphs?

Related questions

On rewriting languages

- What is a good language for rewriting?
  - For RPQ queries and views: unknown.
  - For SPQ queries and UPQ views: FO?
- Here: tool for achieving complexity bounds.
- Other relevant properties?

On the connexion between Datalog and CSP

- [Barto]: If CSP(T) is solvable in Datalog, it is solvable in $\text{Datalog}_{2,3}$.
- Here: $\text{CSP}(T_{\mathrm{Q},\mathbf{V}})$ is solvable in $\text{Datalog}_{\ell,k}$, but on restricted domain. Can we do better?

Thank you!

$\boxed{\text{APPENDIX}}$

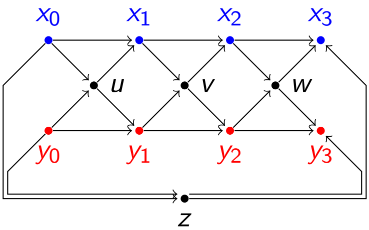
Example 3

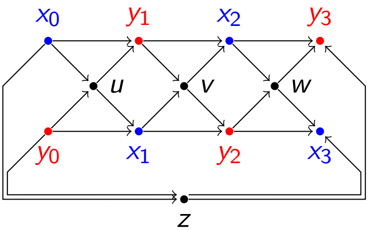
$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,3} = \langle 2, 3 \rangle$$
$$\mathsf{Q}_3 = \langle 3 \rangle \qquad \mathsf{Q}_5 = \langle 5 \rangle$$

Example 3

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,3} = \langle 2, 3 \rangle$$
$$Q_3 = \langle 3 \rangle \qquad Q_5 = \langle 5 \rangle$$

$$\boxed{\mathbf{V} \not\twoheadrightarrow Q_3}$$





Example 3

$$V_2 = \langle 2 \rangle \qquad V_{1,2} = \langle 1, 2 \rangle \qquad V_{2,3} = \langle 2, 3 \rangle$$
$$Q_3 = \langle 3 \rangle \qquad Q_5 = \langle 5 \rangle$$

$$\boxed{\mathbf{V} \twoheadrightarrow Q_5}$$

$$R(x,y) = \quad \exists x_0, x_1, x_2, x_3, x_4, x_5, \quad x = x_0 \wedge y = x_5$$

$$\textstyle\bigwedge_i V_2(x_i, x_{i+2})$$
$$\textstyle\bigwedge_i V_{1,2}(x_i, x_{i+1}) \wedge V_{1,2}(x_i, x_{i+2})$$
$$\textstyle\bigwedge_i V_{2,3}(x_i, x_{i+2}) \wedge V_{2,3}(x_i, x_{i+3})$$

$$\forall z,\ V_{1,2}(z, x_3) \Rightarrow V_2(z, x_3) \vee V_2(z, x_4)$$

Example 5

$$V_1 = \langle a, aa \rangle \qquad V_2 = \langle aa, aaa \rangle \qquad Q = \langle a(a^6)^* \mid aa(a^6)^* \rangle$$

$$\boxed{R(x,y) = \exists z \; V_1(x,z) \wedge T^*(z,y)}$$

where $T(x,y)$ is defined as :