Northeastern University
Khoury College of Computer Sciences

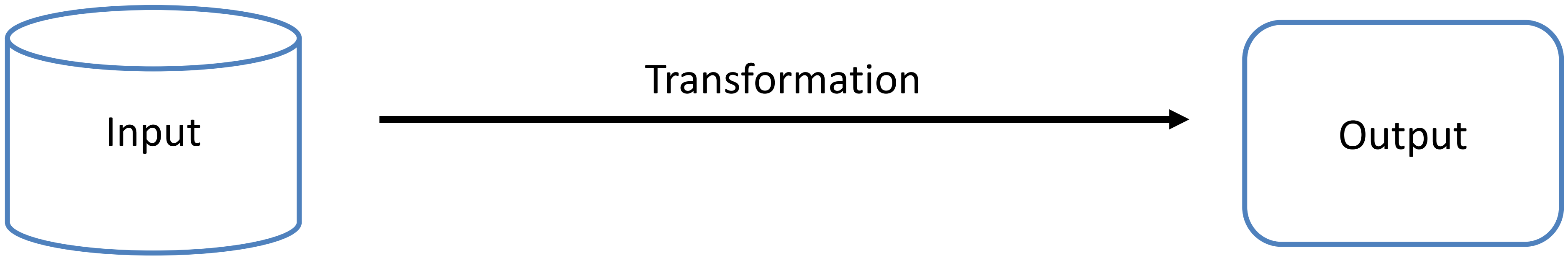
DATALAB

A Unified Approach For Reverse Data Management

Neha Makhija
Northeastern University
(Joint work with Wolfgang Gatterbauer)

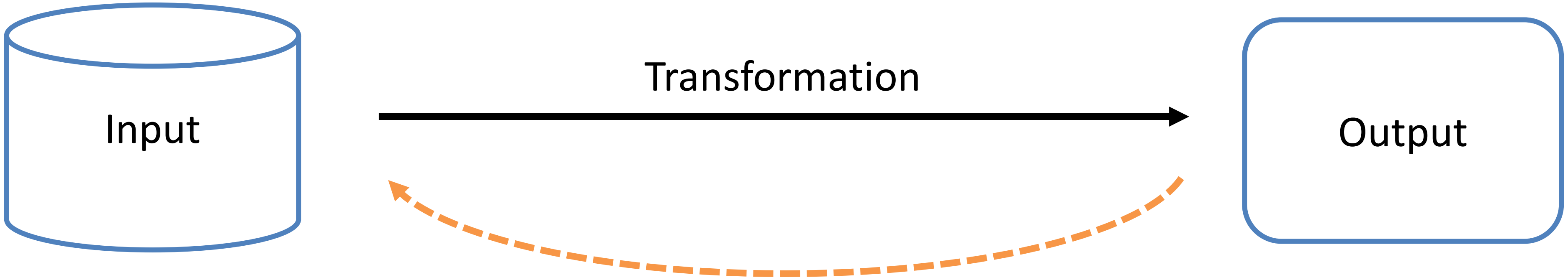
https://northeastern-datalab.github.io/unified-reverse-data-management/

Data Management



But sometimes, results are
- unexpected
- undesirable
- not understandable

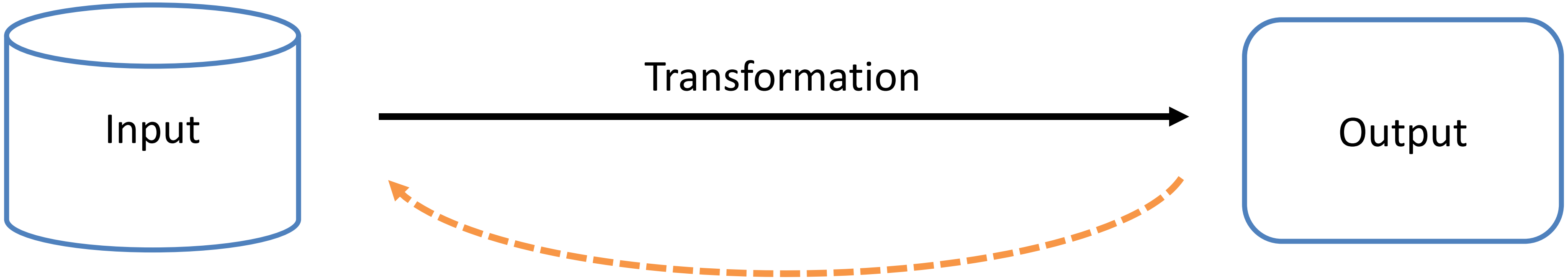
Reverse Data Management



But sometimes, results are
- unexpected
- undesirable
- not understandable

Then we need to reason about the input in terms of the output!

Reverse Data Management

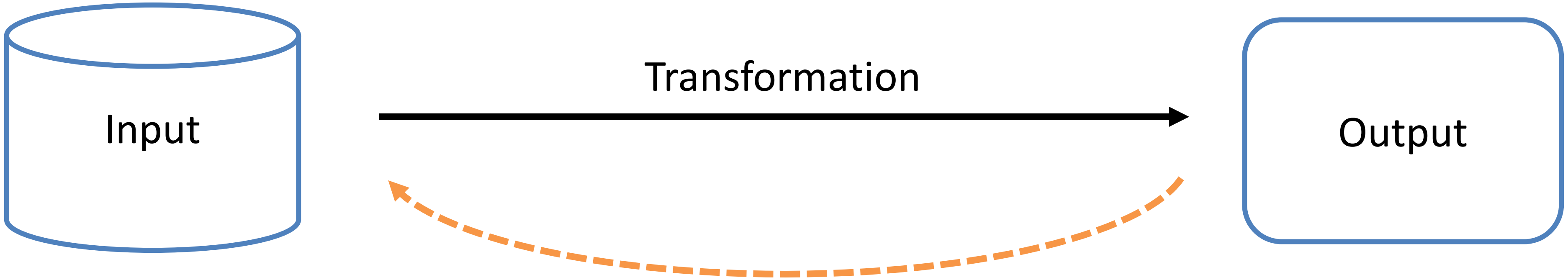


"What change would it take to achieve output X?"

"The handling of transformations that perform actions on the input data, on behalf of desired outcomes in the output data"

Meliou, Gatterbauer, Suciu. Reverse Data Management, VLDB 2011 https://doi.org/10.14778/3402755.3402803

Reverse Data Management: Example



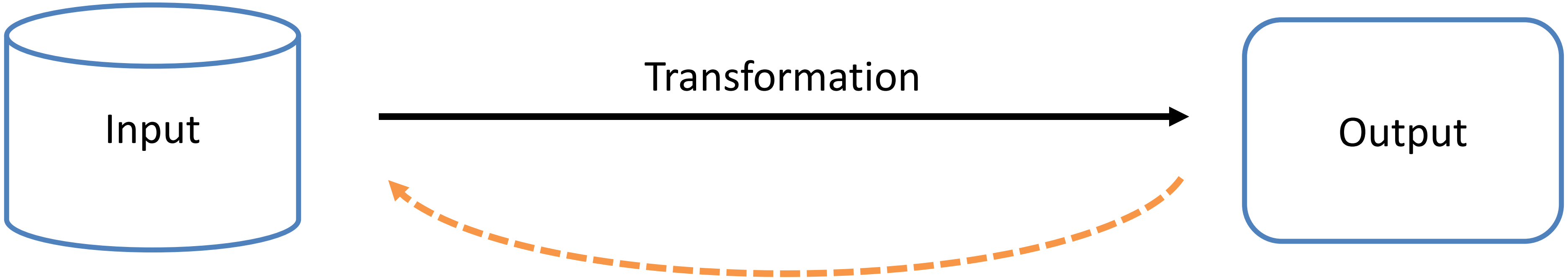
"What change would it take to **delete** **some values from the output**?"

Deletion Propagation

- Minimize Source Side Effect
- Minimize View Side Effect

Dayal, Bernstein. On the correct translation of update operations on relational views, TODS 1982 https://doi.org/10.1145/319732.319740
Buneman, Khanna, Tan. On propagation of deletions and annotations through views, PODS 2002 https://doi.org/10.1145/543613.543633

Reverse Data Management: Example

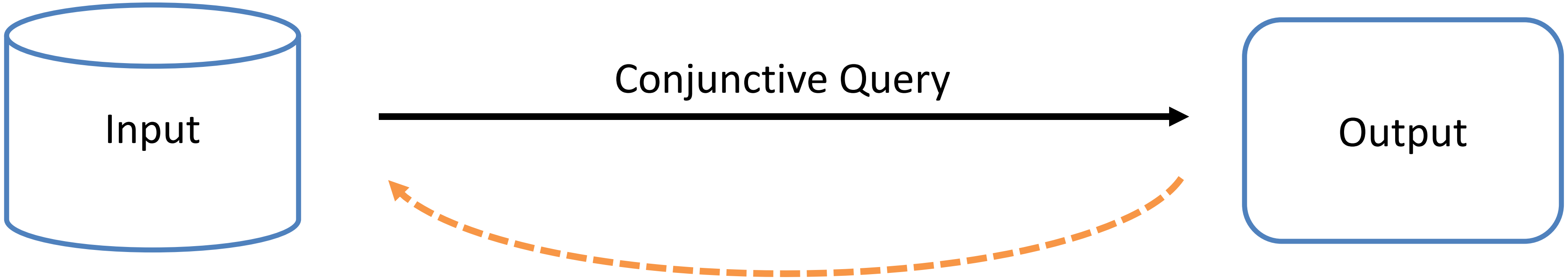


"What minimum change would it take to **ensure output is "fair"**?"

Algorithmic Fairness

Salimi, Rodriguez, Howe, Suciu. Interventional fairness: Causal database repair for algorithmic fairness, SIGMOD 2019 https://doi.org/10.1145/3299869.3319901

Reverse Data Management: Example

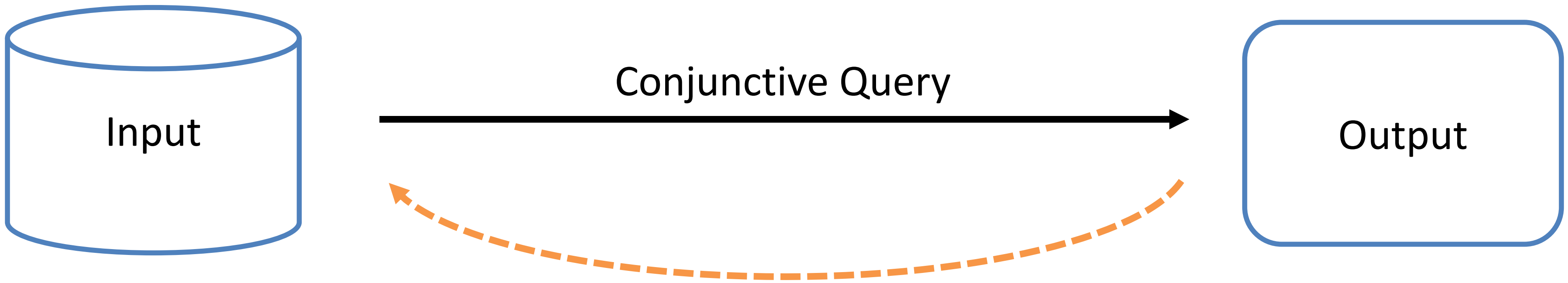


“What change would it take to
minimize the generated provenance expression?”

Minimal Factorization

Makhija, Gatterbauer. Towards a Dichotomy for Minimally Factorizing the Provenance of Self-Join Free Conjunctive Queries, Arxiv 2021 https://arxiv.org/abs/2105.14307

Reverse Data Management: Example



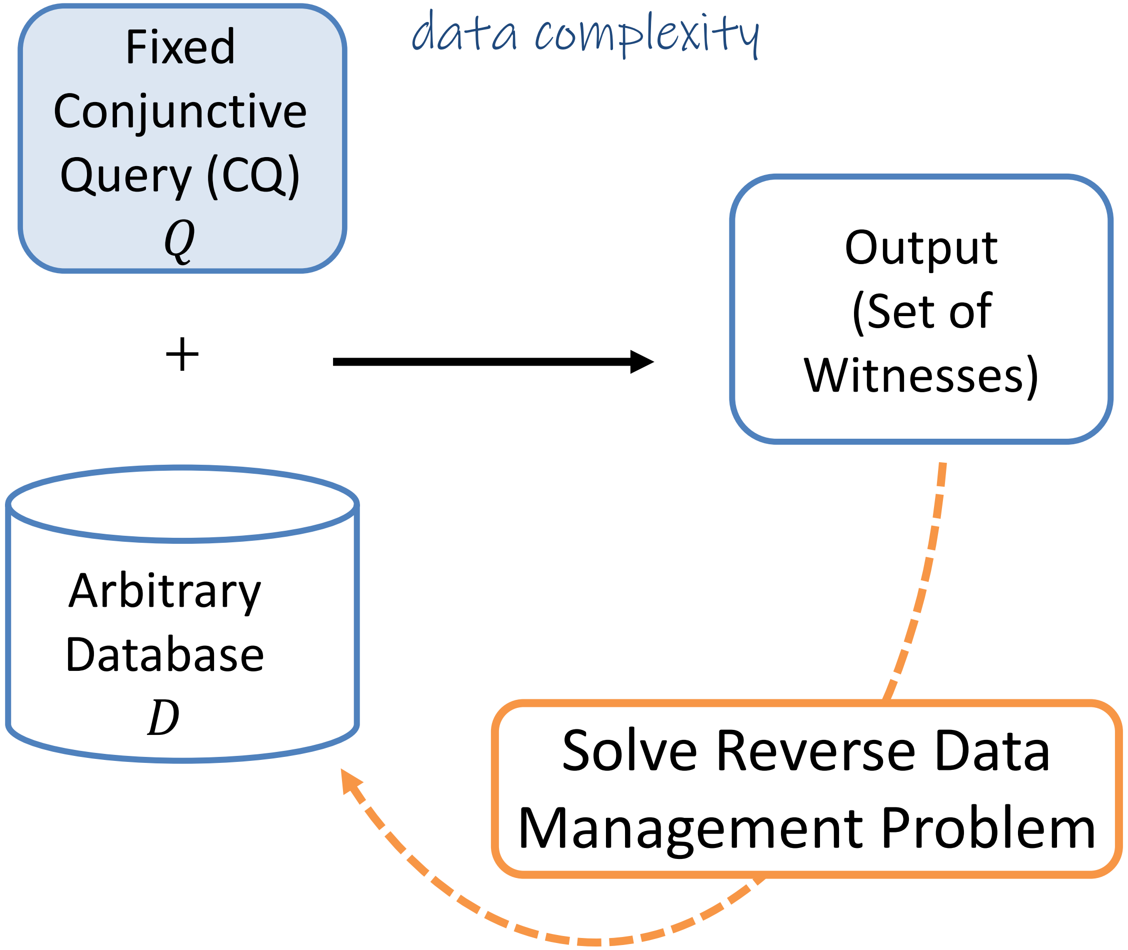
"What minimum change would it take to **make tuple X** **counterfactual**?"

Causal Responsibility

Meliou, Gatterbauer, Halpern, Koch, Moore, Suciu. Causality in Databases, IEEE Special Bulletin 2010 http://sites.computer.org/debull/A10sept/p59.pdf
Meliou, Gatterbauer, Moore, Suciu. The complexity of causality and responsibility for query answers and non-answers, PVLDB 2010 https://doi.org/10.14778/1880172.1880176

Our Goal

In other words,
data complexity



If hard

- Solve exactly
- Approximate in PTIME

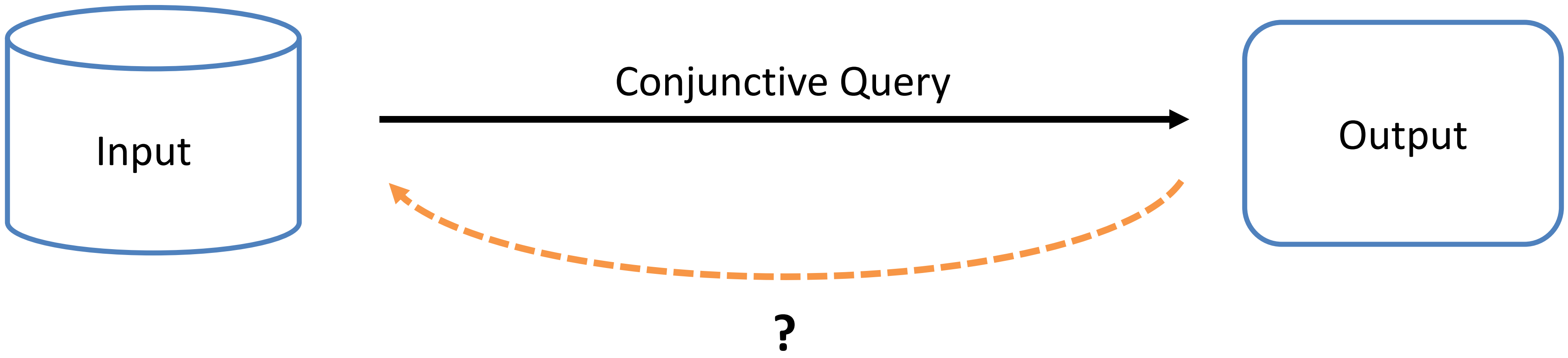
Find tractability border

Exploit structure in data

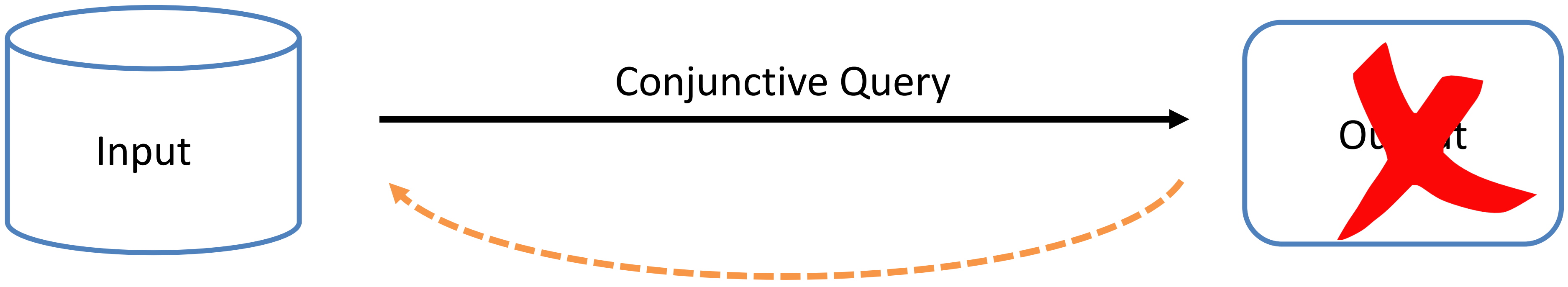
- Solve easy subclasses in PTIME

One unified algorithm

Simplest Reverse Data Management Problem?



Simplest Reverse Data Management Problem?



"What minimal change would it take to **delete the output**?"

Resilience

- Diagnose Points of Failure
- Equivalent to Deletion Propagation with Source Side-Effects

Freire, Gatterbauer, Immerman, Meliou. The complexity of resilience and responsibility for self-join-free conjunctive queries, PVLDB 2015 https://dl.acm.org/doi/10.14778/2850583.2850592

- Reverse Data Management Problems
- Our Focus: Resilience
- Results
- Our Unified Approach
- Takeaways + Open Questions

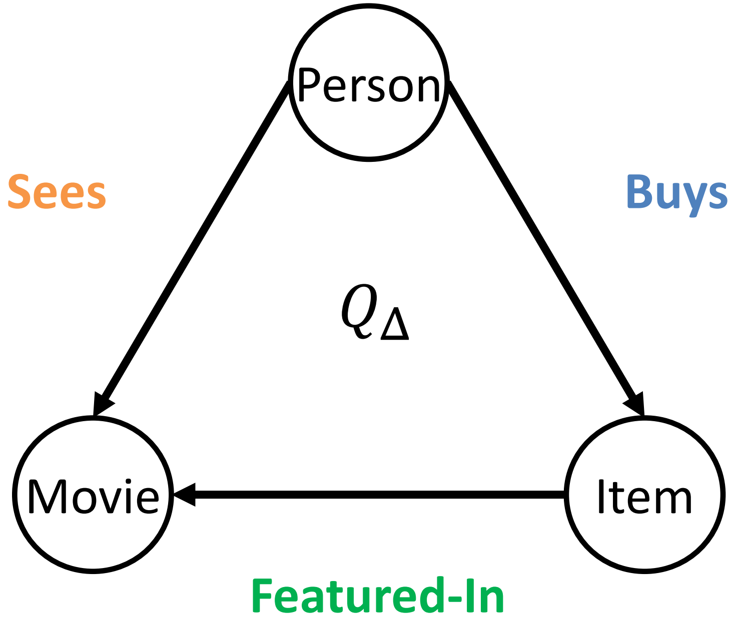
Resilience: Example

Sees(person, movie) **Buys**(person, item) **Featured-In**(item, movie)

Resilience: Example

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)



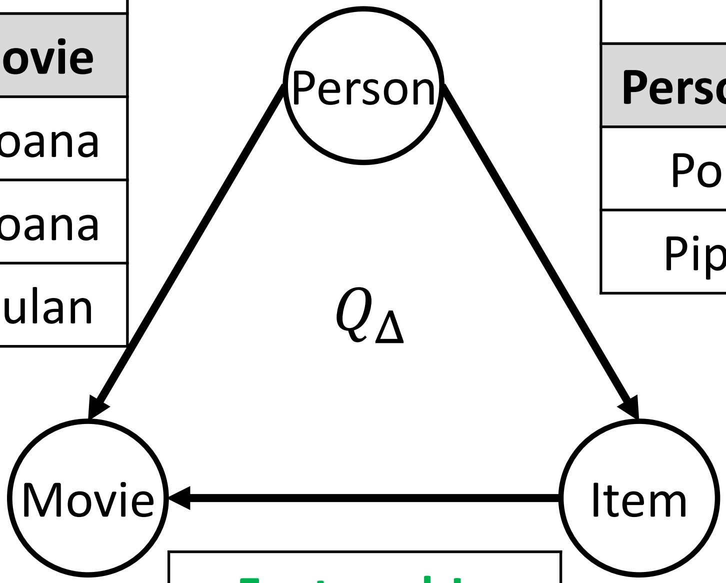
Resilience: Example

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

Sees

	Person	Movie
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan



Buys

Person	Item	
Po	Iron	b_1
Pip	Iron	b_2

Featured-In

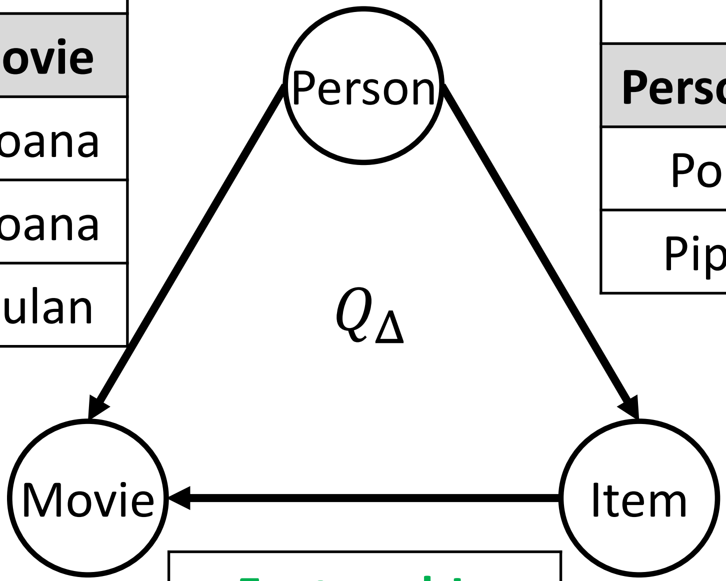
Movie	Item	
Moana	Iron	f_1
Mulan	Iron	f_2

Resilience: Example

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

	Sees	
	Person	**Movie**
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan



Buys		
Person	**Item**	
Po	Iron	b_1
Pip	Iron	b_2

Featured-In		
Movie	**Item**	
Moana	Iron	f_1
Mulan	Iron	f_2

Output			
Person	**Movie**	**Item**	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

Domain values

Tuples

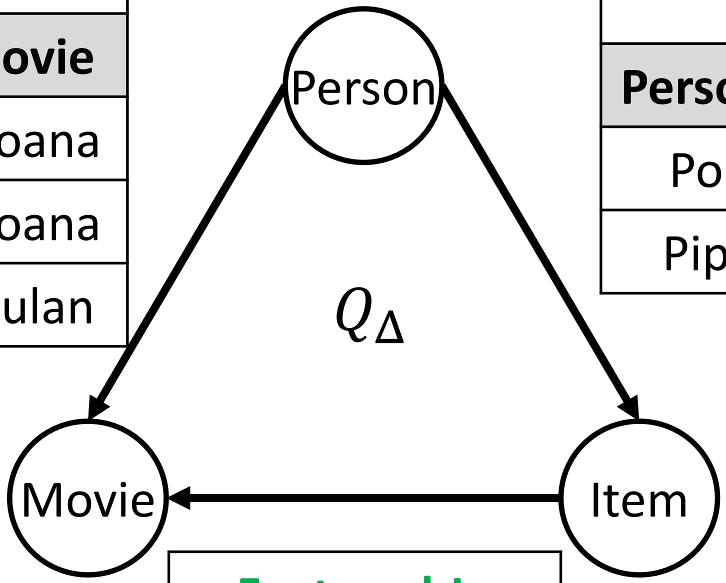
Resilience

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

Sees

	Person	Movie
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan

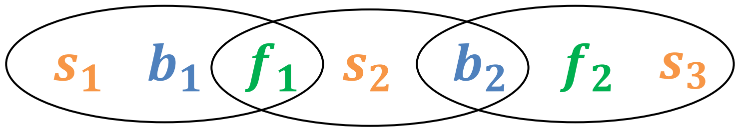


Buys

Person	Item	
Po	Iron	b_1
Pip	Iron	b_2

Featured-In

Movie	Item	
Moana	Iron	f_1
Mulan	Iron	f_2



Output

Person	Movie	Item	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

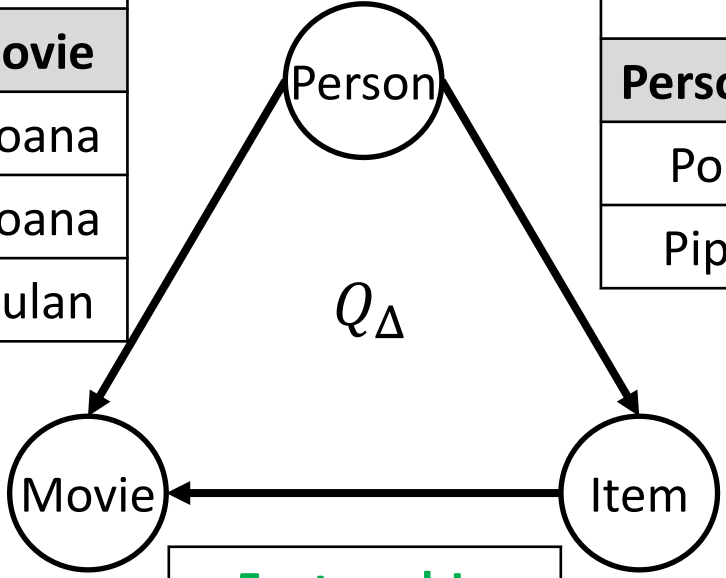
Resilience

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

Sees

	Person	Movie
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan

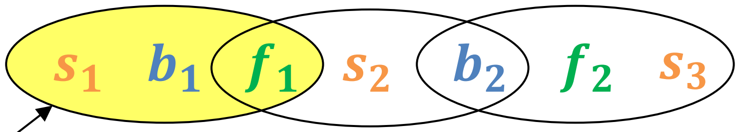


Buys

Person	Item	
Po	Iron	b_1
Pip	Iron	b_2

Featured-In

Movie	Item	
Moana	Iron	f_1
Mulan	Iron	f_2



witness

Output

Person	Movie	Item	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

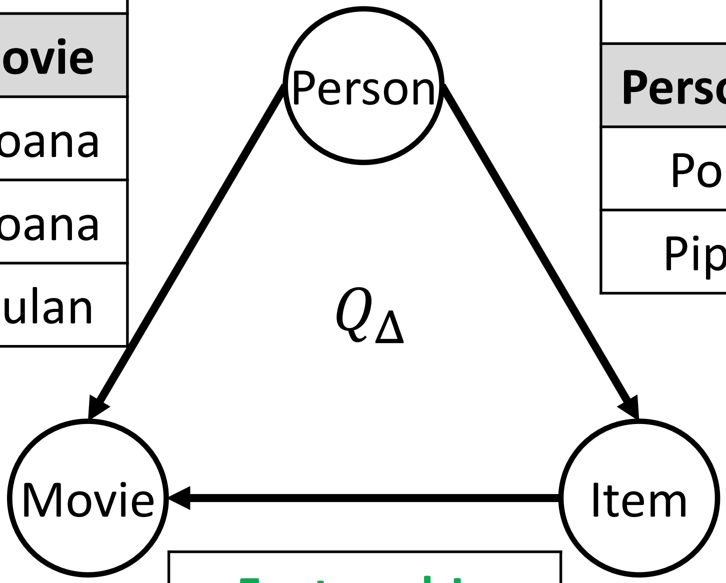
Resilience

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

Sees

	Person	Movie
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan

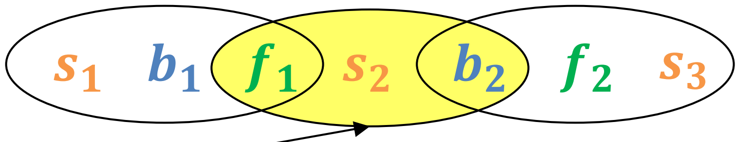


Buys

Person	Item	
Po	Iron	b_1
Pip	Iron	b_2

Featured-In

Movie	Item	
Moana	Iron	f_1
Mulan	Iron	f_2



witness

Output

Person	Movie	Item	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_3 f_2$

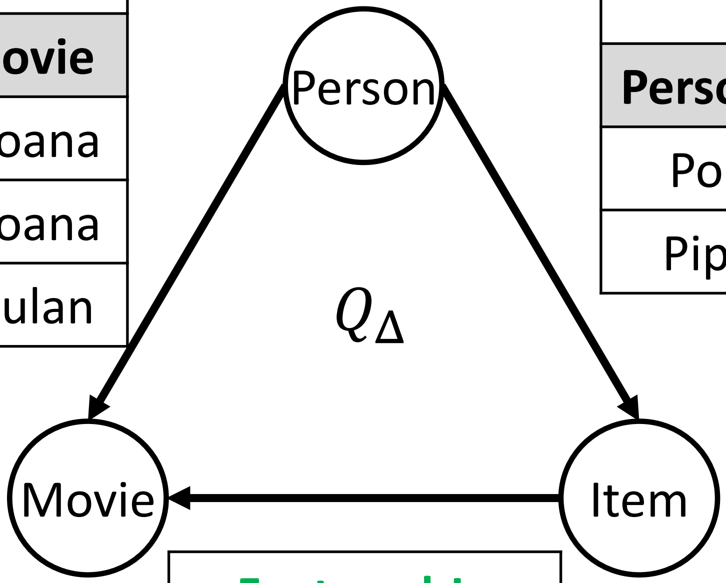
Resilience

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

Sees

	Person	Movie
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan

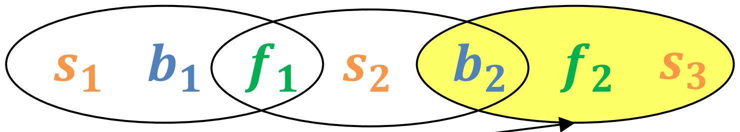


Buys

Person	Item	
Po	Iron	b_1
Pip	Iron	b_2

Featured-In

Movie	Item	
Moana	Iron	f_1
Mulan	Iron	f_2



witness

Output

Person	Movie	Item	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

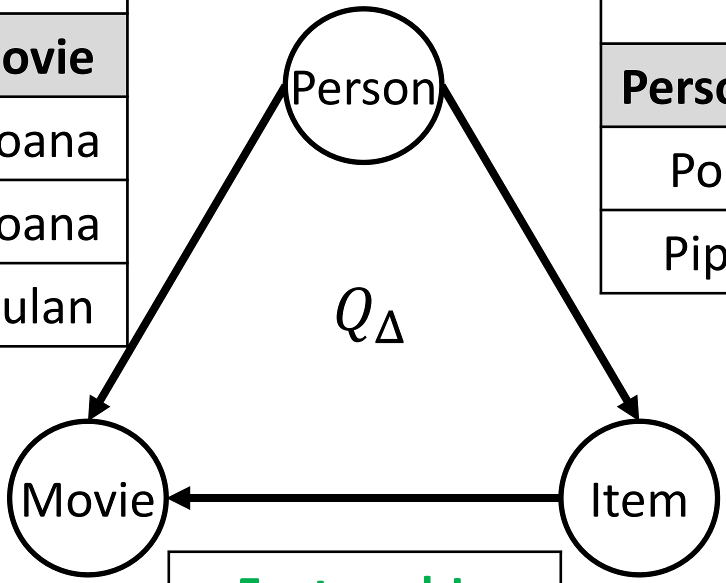
Resilience

Query:- What person ***sees*** a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

Sees

	Person	Movie
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan

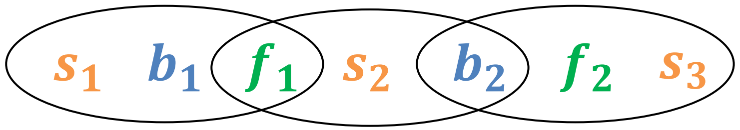


Buys

Person	Item	
Po	Iron	b_1
Pip	Iron	b_2

Featured-In

Movie	Item	
Moana	Iron	f_1
Mulan	Iron	f_2



Output

Person	Movie	Item	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

Recall: Resilience = What minimal change would it take to **delete the output**?”

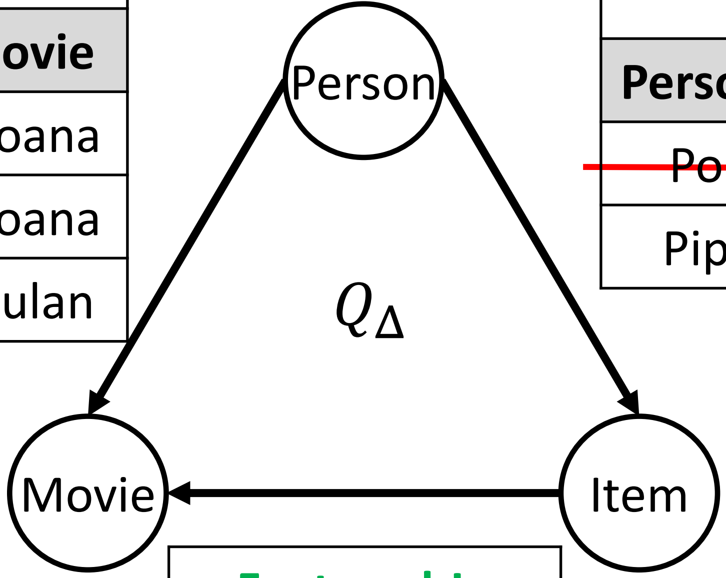
Resilience

Query:- What person *sees* a movie and ***buys*** an item ***featured in*** the movie?

Q(person, movie, item):- **Sees**(person, movie), **Buys**(person, item), **Featured-In**(item, movie)

Sees

	Person	Movie
s_1	Po	Moana
s_2	Pip	Moana
s_3	Pip	Mulan

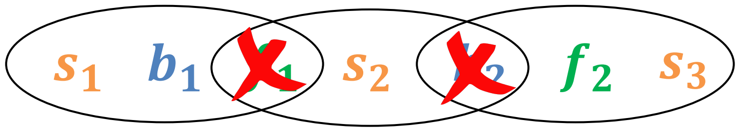


Buys

Person	Item	
~~Po~~	~~Iron~~	b_1
Pip	Iron	b_2

Featured-In

Movie	Item	
~~Moana~~	~~Iron~~	f_1
Mulan	Iron	f_2



Output

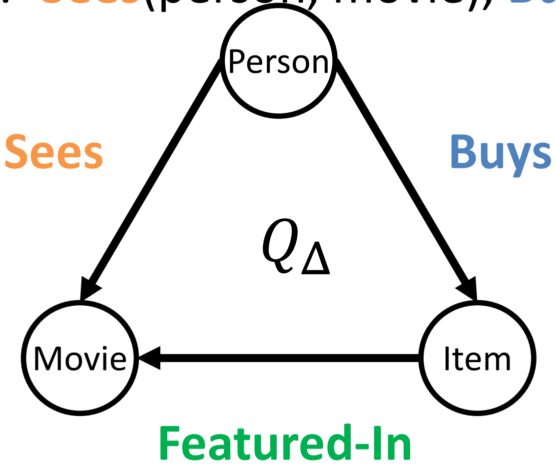
Person	Movie	Item	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

Recall: Resilience = What minimal change would it take to **delete the output**?"

Resilience Complexity

Query:- What person *sees* a movie and *buys* an item *featured in* the movie?

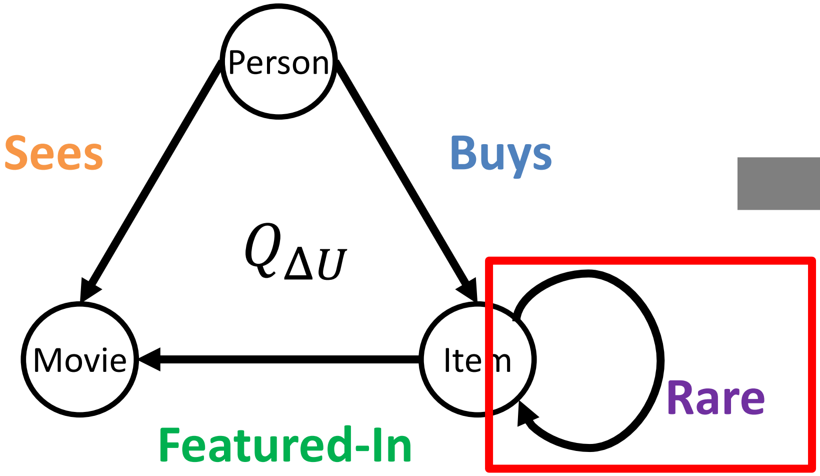
Q(person, movie, item):- Sees(person, movie), Buys(person, item), Featured-In(item, movie)



NP-Complete

Query:- What person *sees* a movie and *buys* an item *featured in* the movie that is rare?

Q(person, movie, item):- Sees(person, movie), Buys(person, item), Featured-In(item, movie), Rare(Item)

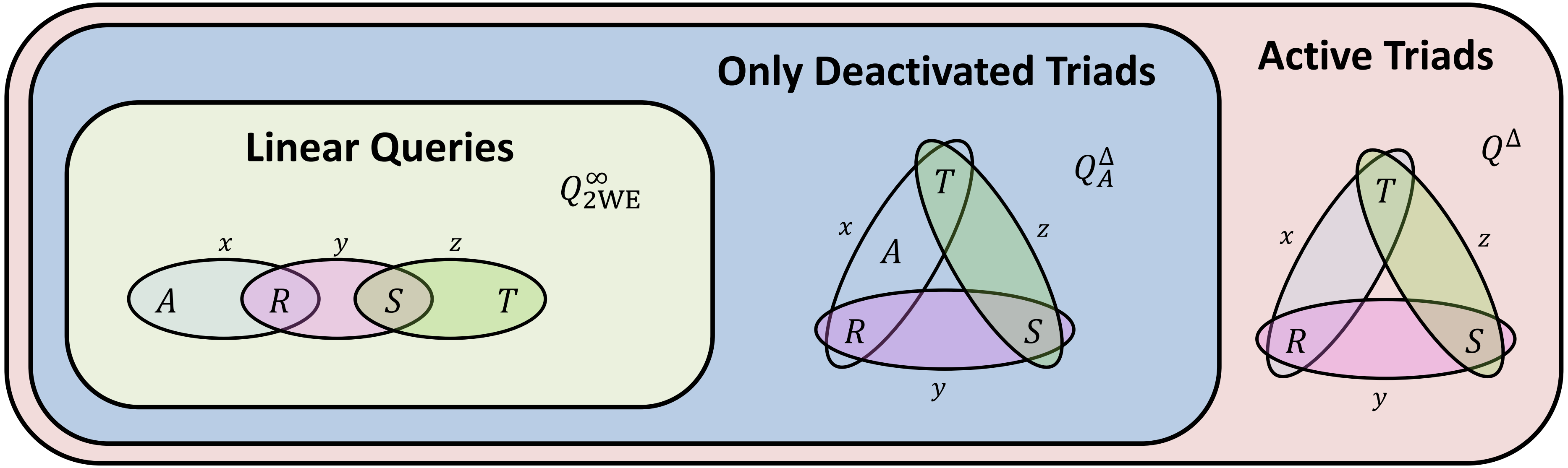


PTIME

Freire, Gatterbauer, Immerman, Meliou. The complexity of resilience and responsibility for self-join-free conjunctive queries, PVLDB 2015 https://dl.acm.org/doi/10.14778/2850583.2850592

- Reverse Data Management Problems
- Our Focus: Resilience
- Results
- Our Unified Approach
- Takeaways + Open Questions

Resilience Complexity: Self-Join Free Queries



Resilience Set Semantics: {

Project-Join Queries are NP-Hard			Buneman+02
PTIME	PTIME	NPC	Freire+15

Resilience Bag Semantics:

PTIME	NPC	NPC	Makhija+24

Complexity Results for *Self-Join Free* Queries

Buneman, Khanna, Tan. On propagation of deletions and annotations through views, PODS 2002 https://doi.org/10.1145/543613.543633
Freire, Gatterbauer, Immerman, Meliou. The complexity of resilience and responsibility for self-join-free conjunctive queries, PVLDB 2015 https://dl.acm.org/doi/10.14778/2850583.2850592
Makhija, Gatterbauer. A Unified Approach for Resilience and Causal Responsibility with Integer Linear Programming (ILP) and LP Relaxations, *SIGMOD 2024* https://arxiv.org/abs/2212.08898

Unified Algorithm (Easy or Hard)

	Prior Work (RES+RESP)	Our Approach (RES+RESP)
Self-Join Free PTIME Cases	Flow based encodings specific to query [Freire+15]	One **Unified** Algorithm - Solves all cases - All known PTIME cases terminate in PTIME
Self-Join Free NPC Cases: Exact Evaluation	No prior algorithm: brute force trivially	
Self-Join Free NPC Cases: Approximations	No prior algorithm	
Queries with Self-Joins	Flow algorithms for some known PTIME queries [Freire+20]	

Freire, Gatterbauer, Immerman, Meliou. The complexity of resilience and responsibility for self-join-free conjunctive queries, PVLDB 2015 https://dl.acm.org/doi/10.14778/2850583.2850592

Freire, Gatterbauer, Immerman, Meliou. New results for the complexity of resilience for binary conjunctive queries with self-joins, PODS 2020 https://doi.org/10.1145/3375395.3387647

Makhija, Gatterbauer. A Unified Approach for Resilience and Causal Responsibility with Integer Linear Programming (ILP) and LP Relaxations, *SIGMOD 2024* https://arxiv.org/abs/2212.08898

Unified Algorithm (Across Different Settings)

	Prior Work	Our Work
Semantics	Set Semantics	Set + **Bag** Semantics
Queries	Self-Join Free Queries + Some Restricted SJ Queries	**All** UCQs
Functional Dependencies	Can leverage known FDs	Take advantage of **unspecified FDs** in data

- Reverse Data Management Problems
- Our Focus: Resilience
- Results
- Our Unified Approach
  - Unified Algorithm
  - Unified Hardness Criterion
- Takeaways + Open Questions

Resilience as an Integer Linear Program (ILP)

Weights if applied can go in the objective

$$\min \ \mathbf{c}^{\mathrm{T}}\mathbf{x}$$

objective vector

$$\mathbf{A}\mathbf{x} \geq \mathbf{b}$$

constraint vector

constraint matrix

$$\mathbf{x} \in \{0,1\}^n$$

$$\min[\, x[s_1] + x[b_1] + x[f_1] + x[s_2] + x[b_2] + x[f_2] + x[s_3] \,]$$

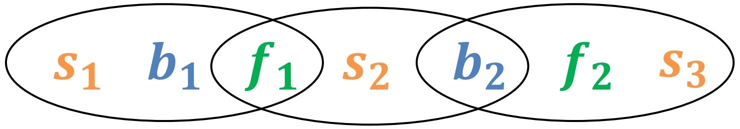
$$x[s_1] + x[b_1] + x[f_1] \geq 1$$

$$x[f_1] + x[s_2] + x[b_2] \geq 1$$

$$x[b_2] + x[f_2] + x[s_3] \geq 1$$

$\forall$ tuples t: $\mathbf{x}[\mathbf{t}] \in \{0,1\}$

Q_Δ Example:



Output			
Person	**Movie**	**Item**	
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

Queries with self-joins are modelled the same way. Not all constraints must have the same arity.

Setting $X[f_1] = X[b_2] = 1$, and all other variables to 0 satisfies the ILP.

Optimal value = 2

Resilience as a Linear Program (LP)

$$\min \ \mathbf{c}^{\mathbf{T}}\mathbf{x}$$

objective vector

constraint vector

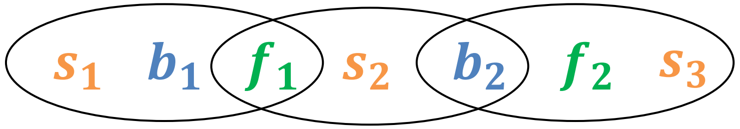
$$\mathbf{A}\mathbf{x} \geq \mathbf{b}$$

constraint matrix

~~$\mathbf{x} \in \{0,1\}^n$~~

$$\mathbf{x} = [0,1]^n$$

Q_Δ Example:



Output			
Po	Moana	Iron	$s_1 b_1 f_1$
Pip	Moana	Iron	$s_2 b_2 f_1$
Pip	Mulan	Iron	$s_3 b_2 f_2$

$$\min[\, x[s_1] + x[b_1] + x[f_1] + x[s_2] + x[b_2] + x[f_2] + x[s_3] \,]$$

$$x[s_1] + x[b_1] + x[f_1] \geq 1$$

$$x[f_1] + x[s_2] + x[b_2] \geq 1$$

$$x[b_2] + x[f_2] + x[s_3] \geq 1$$

~~$\forall$ tuples t: $\mathbf{x}[\mathbf{t}] \in \{0,1\}$~~

$\forall$ tuples t: $0 \leq \mathbf{x}[\mathbf{t}] \leq 1$

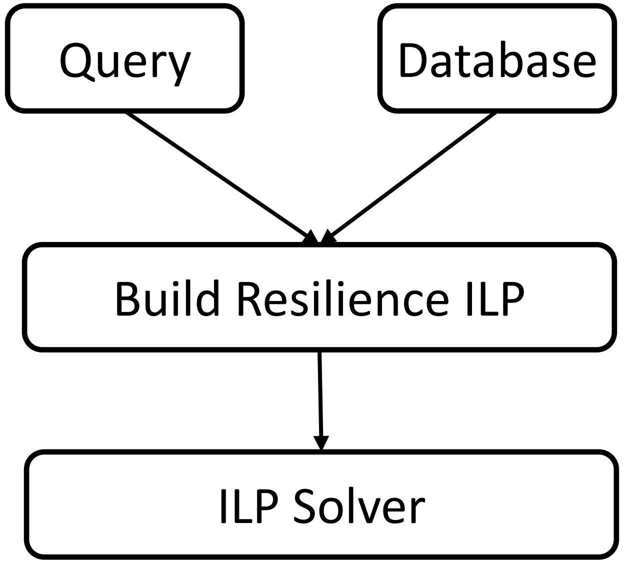
Linear programs allow fractional solutions

We can get a lower bound in PTIME.

Unified Algorithm for Resilience

Theorem.

For all known PTIME cases of resilience, the objective of the LP relaxation is identical to the ILP.

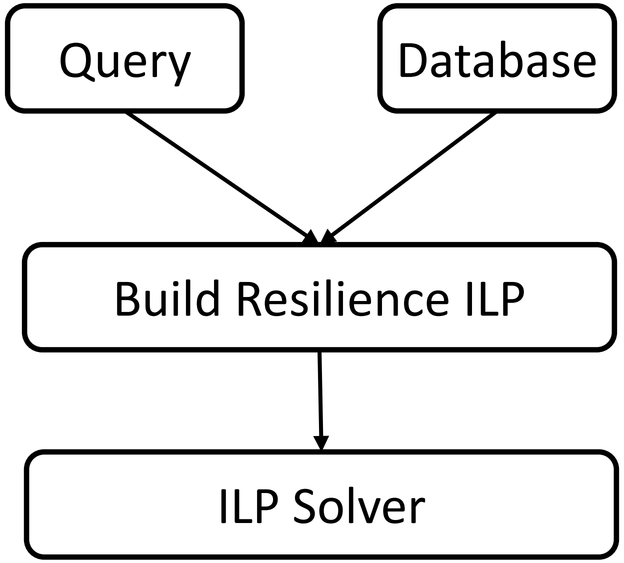


But ILPs are NP-Hard!

Unified Algorithm for Resilience

Theorem.

For all known PTIME cases of Resilience, **LP=ILP**

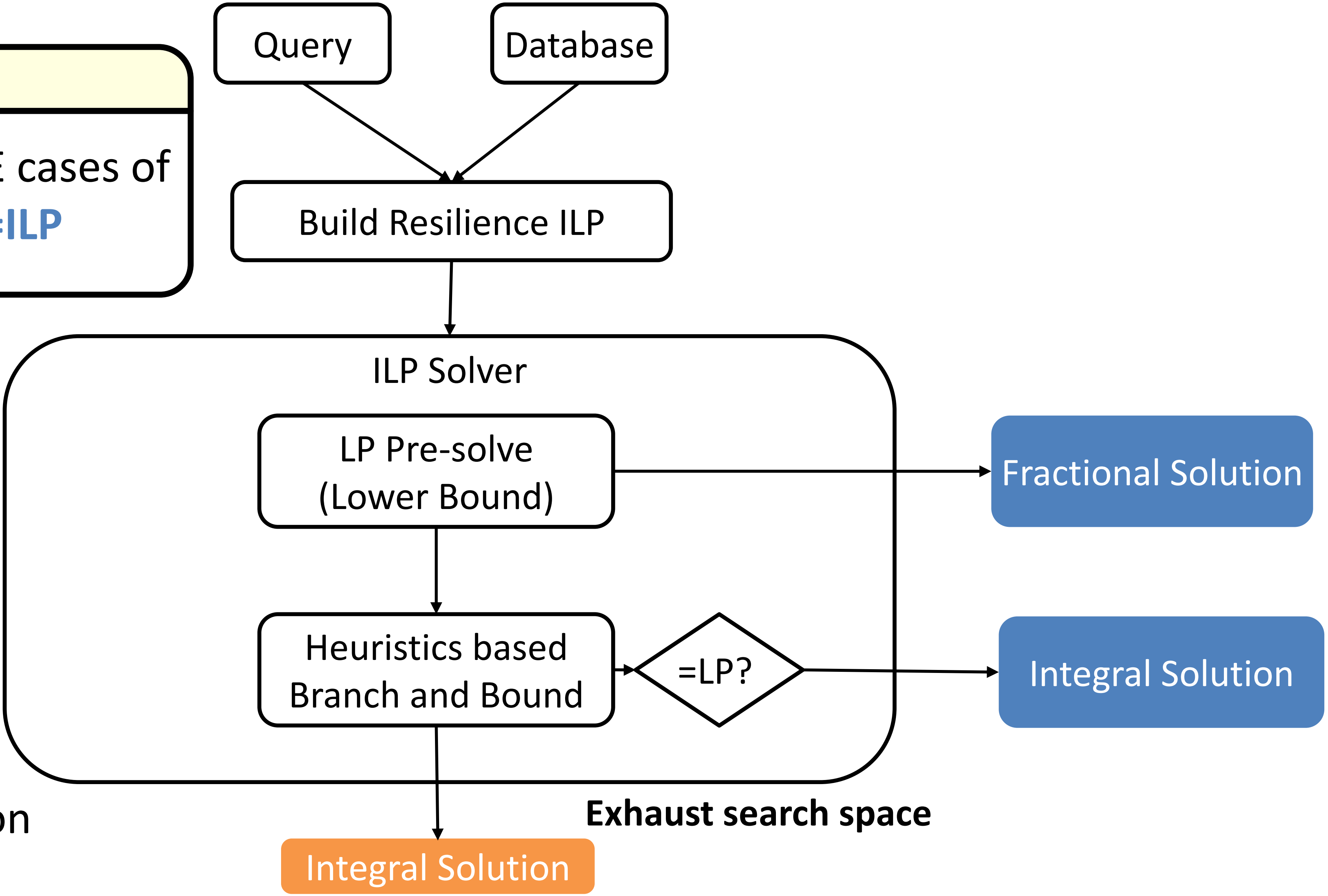


But ILPs are NP-Hard!

Unified Algorithm for Resilience

Theorem.

For all known PTIME cases of Resilience, **LP=ILP**



If **LP=ILP**, solvers can recover a solution efficiently!

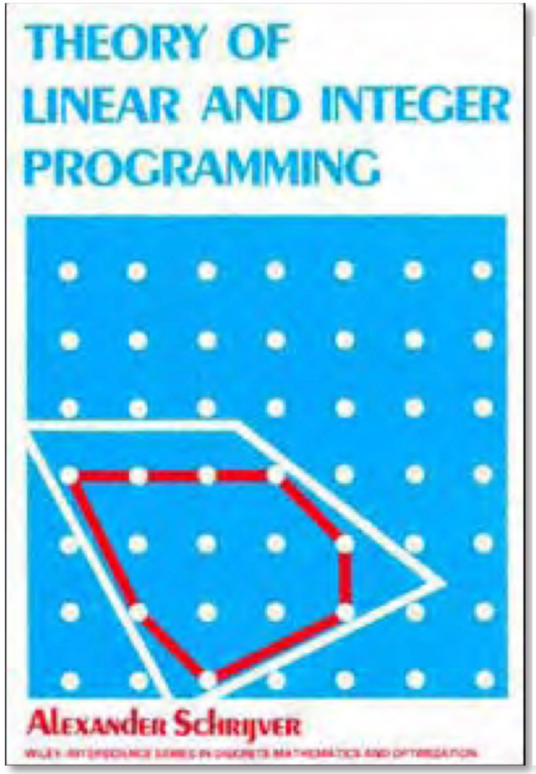
When does LP = ILP?

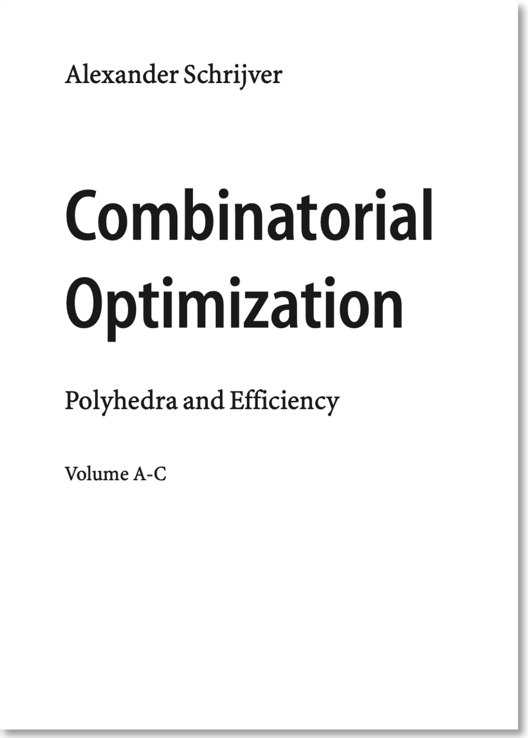
Theorem.

For all known PTIME cases of Resilience, **LP=ILP**

Our PTIME constraint matrixes need not be balanced or Totally Unimodular

The PTIME cases go beyond these known criterion!

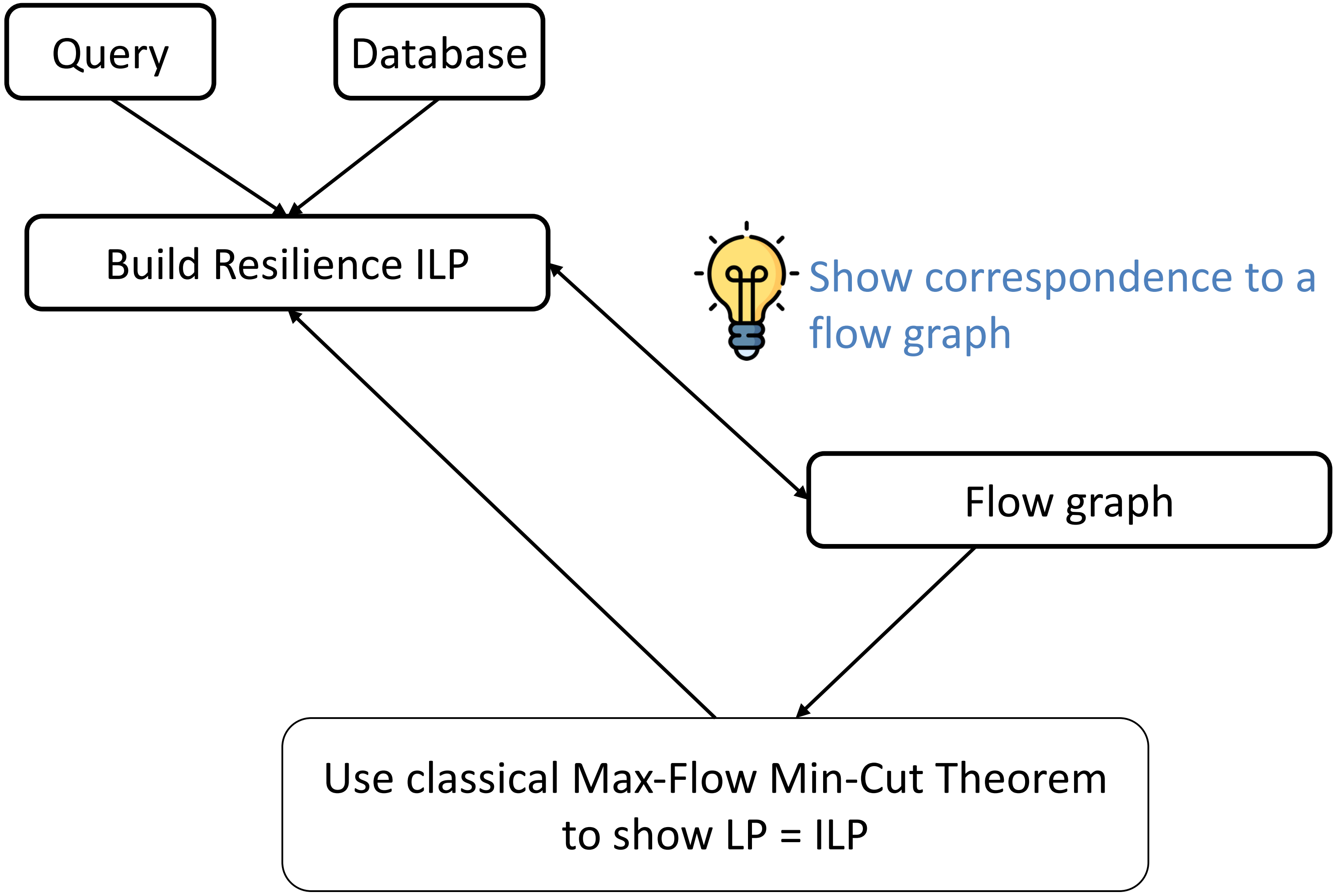




When does LP = ILP?

Theorem.

For all known PTIME cases of Resilience, **LP=ILP**



- Reverse Data Management Problems
- Our Focus: Resilience
- Results
- Our Unified Approach
  - Unified Algorithm
  - Unified Hardness Criterion
- Takeaways + Open Questions

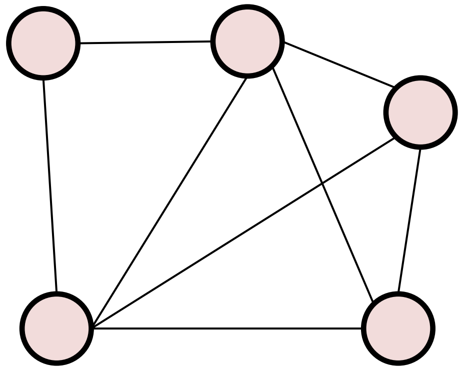
Unified Approach to prove Hardness

Vertex Cover → Resilience(Q)

Reduction Gadget ?

Arbitrary graph G → **Corresponding database D**

?



$\mathrm{VC(G) = k}$

	Sees	
s_1	...	...
s_2	...	...
s_3	...	...

	Featured-In	
f_1	...	...
f_2	...	...
f_3	...	...

	Buys	
b_1	...	...
b_2	...	...
b_3	...	...

$\mathrm{Resilience(Q, D) = f(k)}$

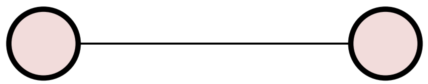
Unified Approach to prove Hardness

Vertex Cover → Resilience(Q)

Reduction Gadget ?

Single Edge → **Corresponding database *D***

Edge Gadget



	Sees	
s_1	…	…
s_2	…	…
s_3	…	…

	Featured-In	
f_1	…	…
f_2	…	…
f_3	…	…

	Buys	
b_1	…	…
b_2	…	…
b_3	…	…

$\mathrm{VC(G) = k}$

$\mathrm{Resilience(Q, D) = f(k)}$

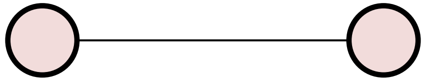
Unified Approach to prove Hardness

Vertex Cover → Resilience(Q)

Reduction Gadget ?

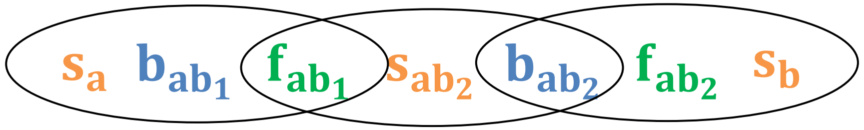
Single Edge → **Corresponding database D**

Edge Gadget



Independent Join Path

We prove: If a query forms an "Independent Join Path" → it is NPC (conjecture Freire+20)



Data Hypergraph

$\mathrm{VC}(G) = k$

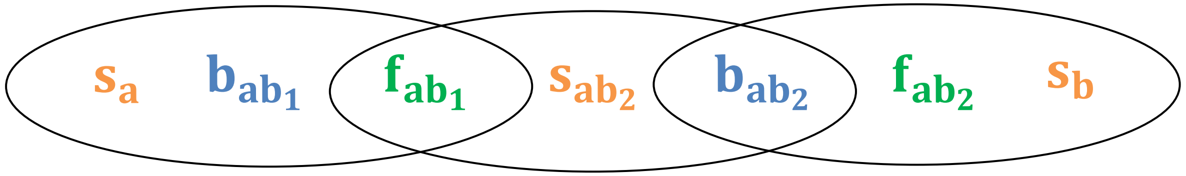
$\mathrm{Resilience}(Q, D) = f(k)$

Unified Approach to prove Hardness

What is an Independent Join Path?

Database under query Q, with endpoints, with 5 ***testable*** properties:

1. Data hypergraph is connected



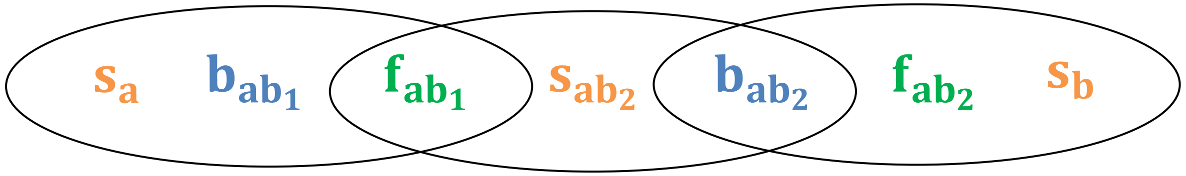
Data Hypergraph

Unified Approach to prove Hardness

What is an Independent Join Path?

Database under query Q, with endpoints, with 5 ***testable*** properties:

1. Data hypergraph is connected
2. Database is reduced



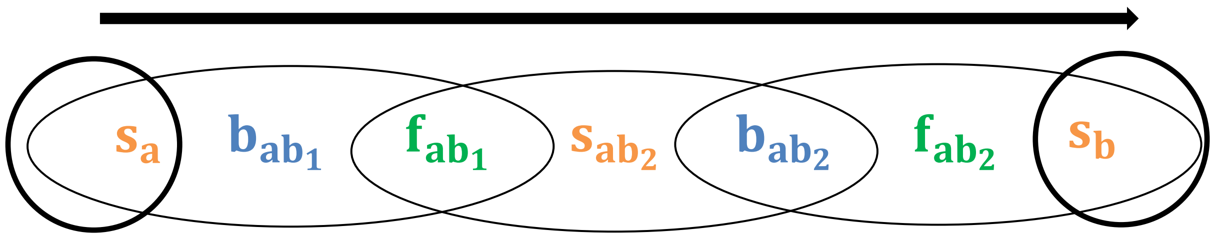
Data Hypergraph

Unified Approach to prove Hardness

What is an Independent Join Path?

Database under query Q, with endpoints, with 5 ***testable*** properties:

1. Data hypergraph is connected
2. Database is reduced
3. Endpoints are "valid"



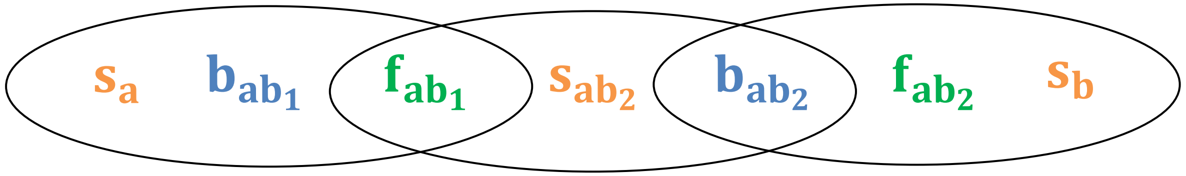
Data Hypergraph

Unified Approach to prove Hardness

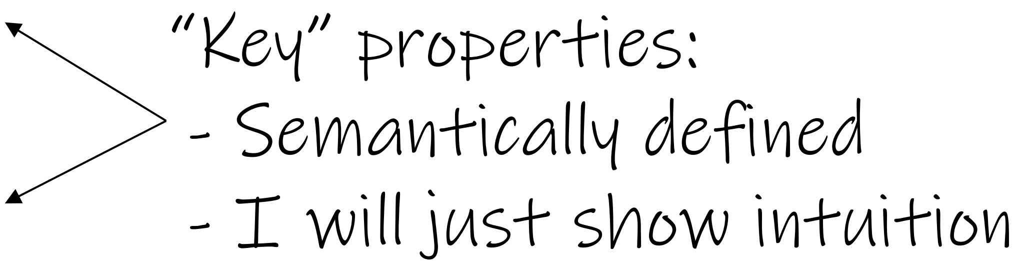
What is an Independent Join Path?

Database under query Q, with endpoints, with 5 ***testable*** properties:

1. Data hypergraph is connected
2. Database is reduced
3. Endpoints are "valid"
4. OR – property
5. Composability

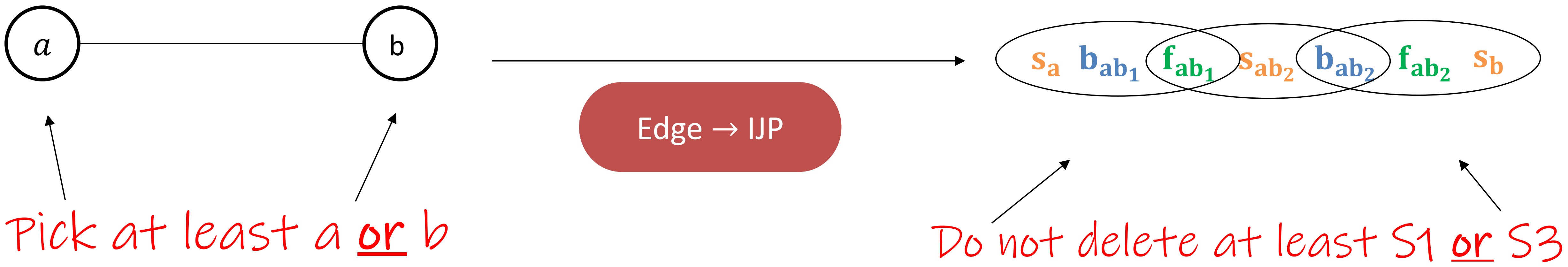


Data Hypergraph



Unified Approach to prove Hardness

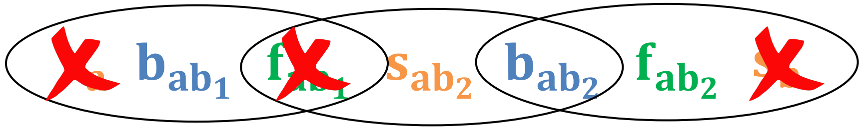
Key Property #1: OR property



Unified Approach to prove Hardness

Key Property #1: OR property

Edge → IJP



Pick at least a **or** b

Do not delete at least S1 **or** S3

VC = ??

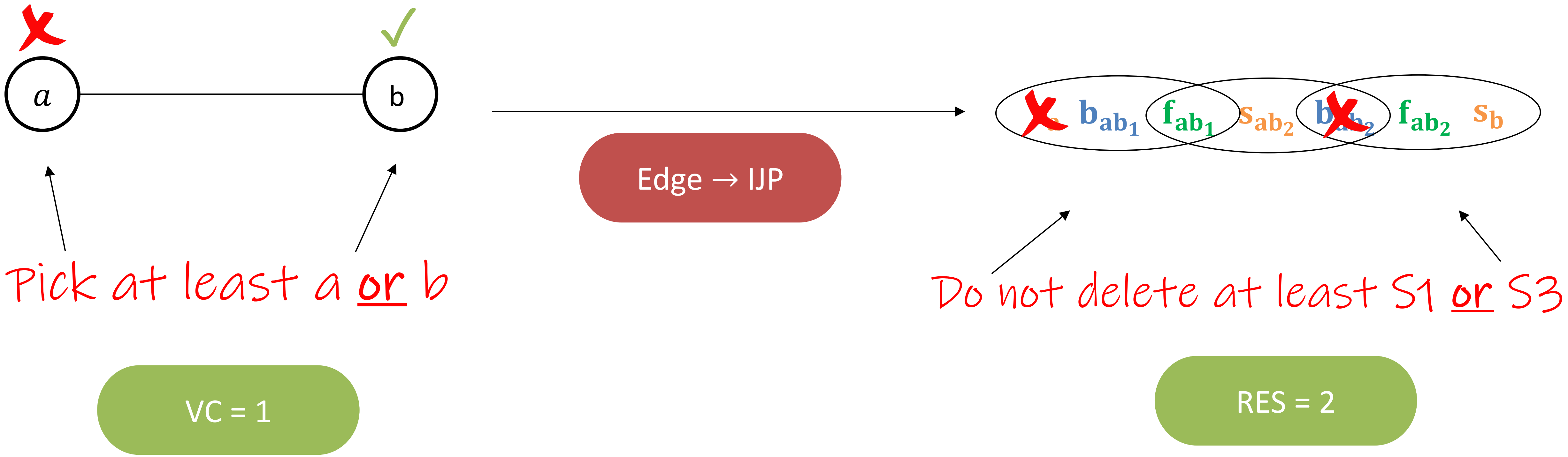
Violates constraints

RES = 3

Violates minimality

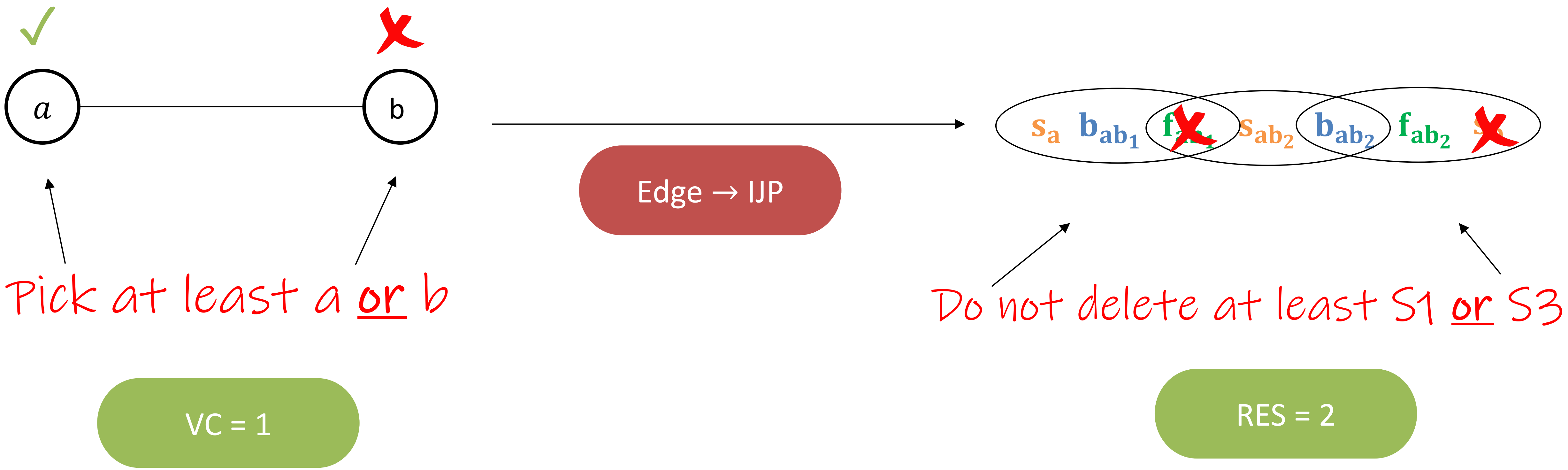
Unified Approach to prove Hardness

Key Property #1: OR property



Unified Approach to prove Hardness

Key Property #1: OR property



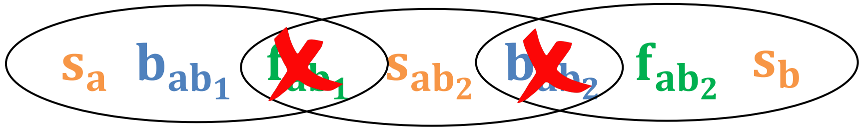
Unified Approach to prove Hardness

Key Property #1: OR property

Pick at least a **or** b

VC = 2

Edge → IJP

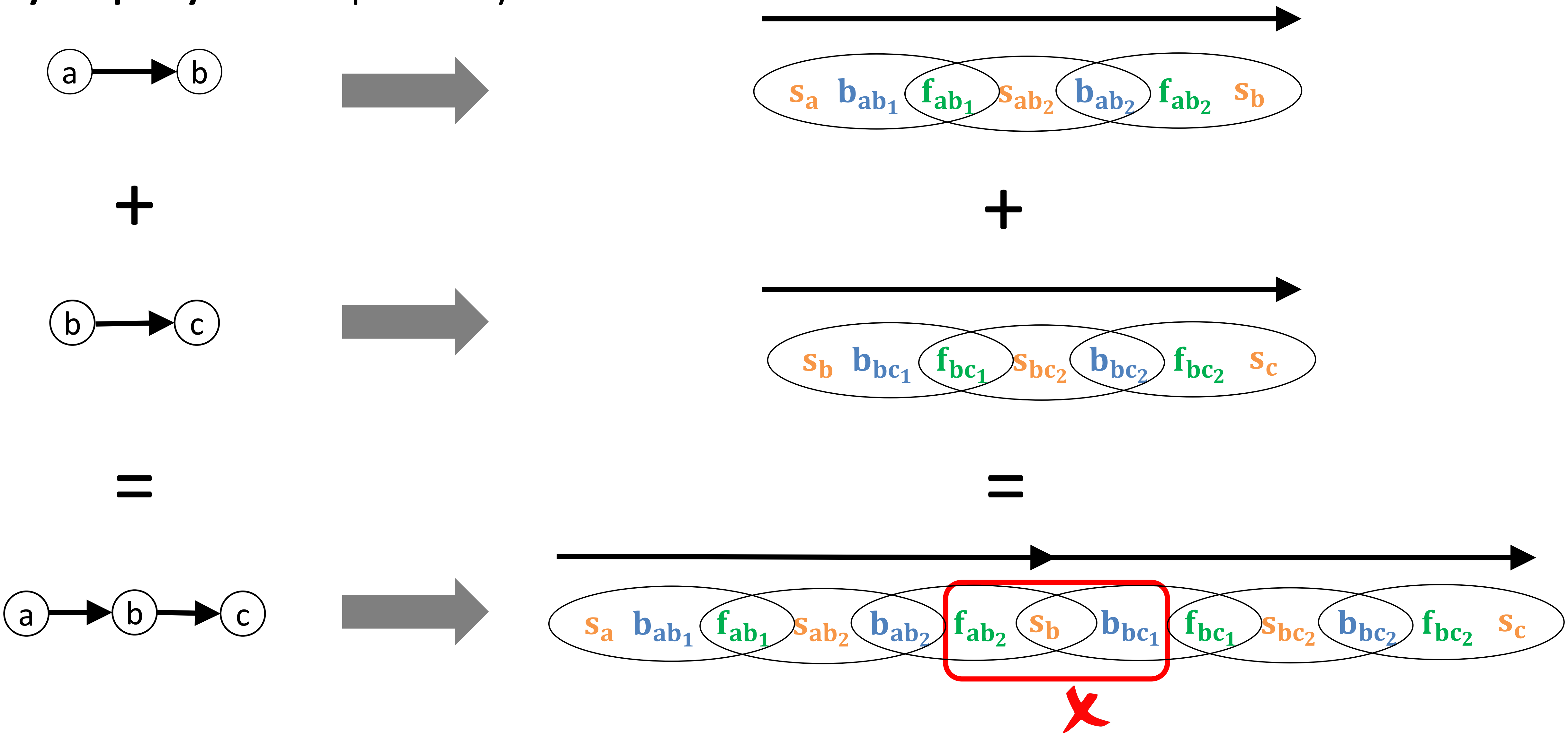


Do not delete at least S1 **or** S3

RES = 2

Unified Approach to prove Hardness

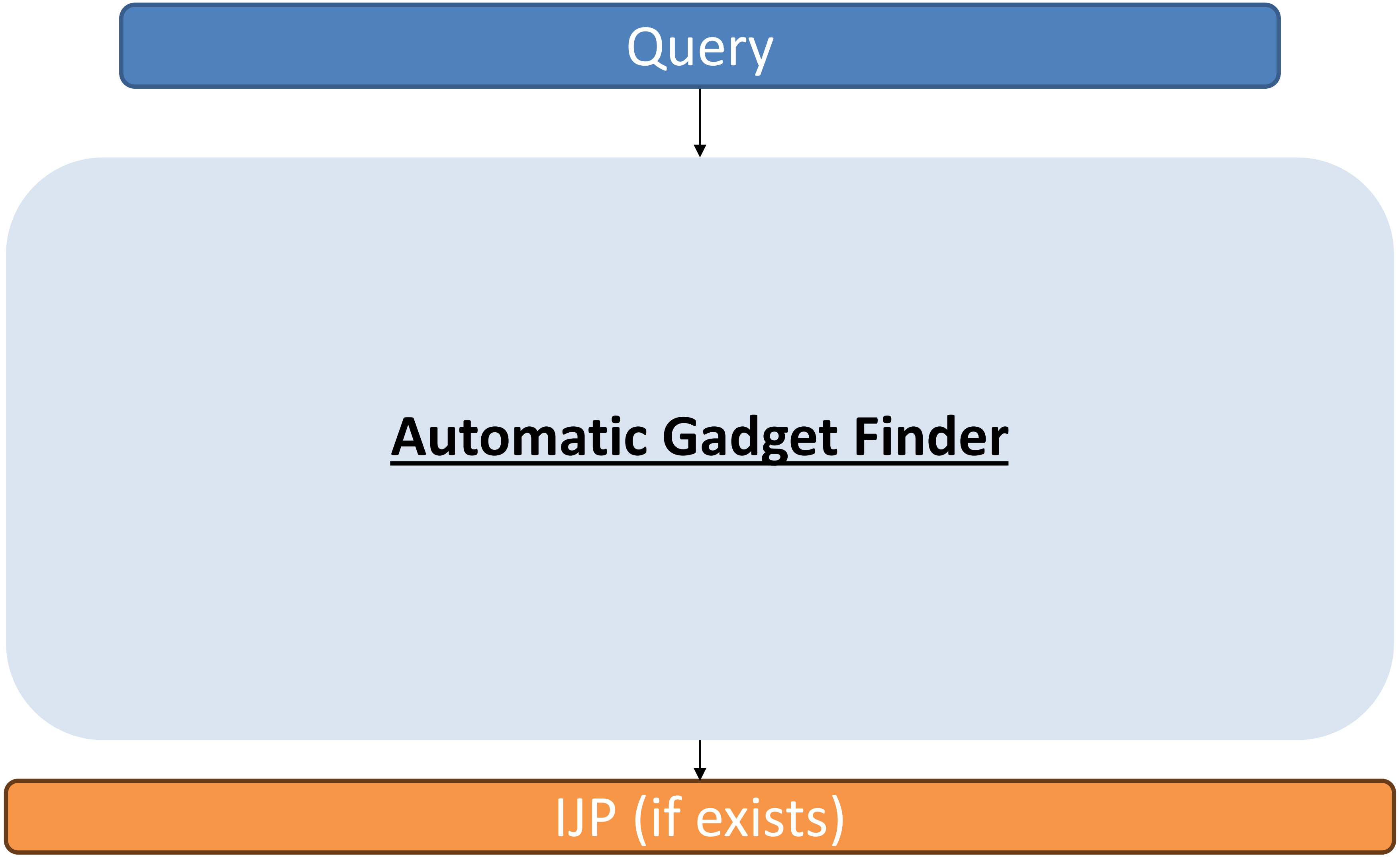
Key Property #2: Composability of IJPs



Composing two IJPs should not lead to additional witnesses

Our Goal

Using the complete criterion for IJP, can we build a principled way to find IJPs?



Unified Approach to prove Hardness

Using the complete criterion for IJP, we can build a principled way to find IJPs

Query, Domain Size of IJP

More on size in just a bit

Automatic Gadget Finder

Search through an exponential space – “guess” an IJP

Verify IJP:

Check IJP has OR property

Check IJP composes

Certificate of Hardness

Unified Approach to prove Hardness

Using the complete criterion for IJP, we can build a principled way to find IJPs

Query, Domain Size of IJP

More on size in just a bit

Automatic Gadget Finder

Search through an exponential space – "guess" an IJP

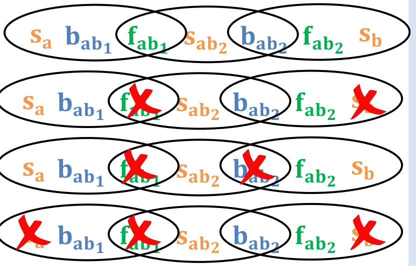
Verify IJP:

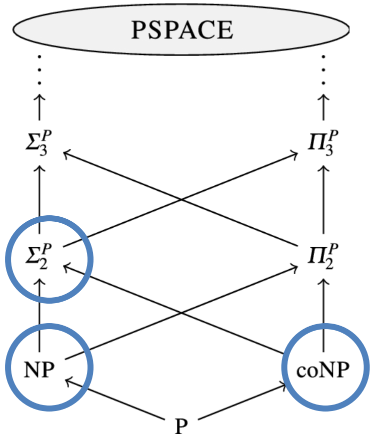
Check IJP has OR property

- Find size k resilience set ***(NP)***
- Find no size k-1 resilience set ***(Co-NP)***

Check IJP composes

Certificate of Hardness





Unified Approach to prove Hardness

Using the complete criterion for IJP, we can build a principled way to find IJPs

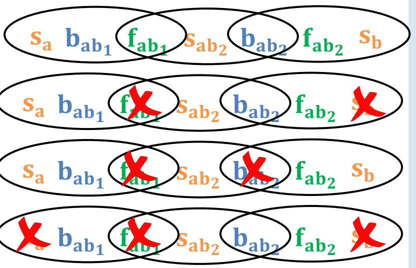
Query, Domain Size of IJP

More on size in just a bit

Automatic Gadget Finder

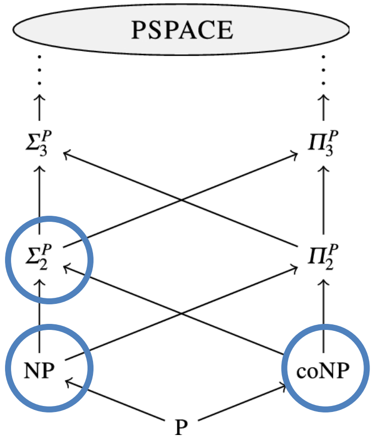
Search through an exponential space – "guess" an IJP

Verify IJP:



Check IJP has OR property

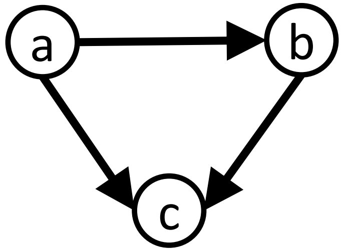
- Find size k resilience set ***(NP)***
- Find no size k-1 resilience set ***(Co-NP)***



Check IJP composes

Theorem.

It suffices to check just 3 join paths compose



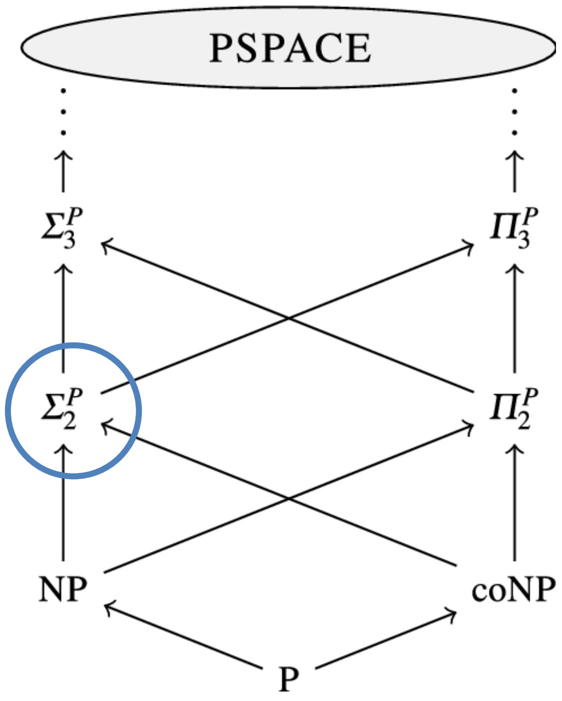
Certificate of Hardness

Unified Approach to prove Hardness

Using the complete criterion for IJP, we can build a principled way to find IJPs

Query, Domain Size of IJP

More on size in just a bit



Automatic Hardness Finder

<u>Disjunctive Logic Program</u>

Σ_p^2

Declarative

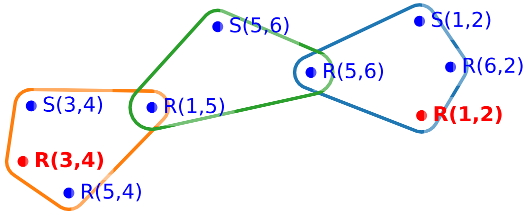
$x \lor y \text{ :- } z$

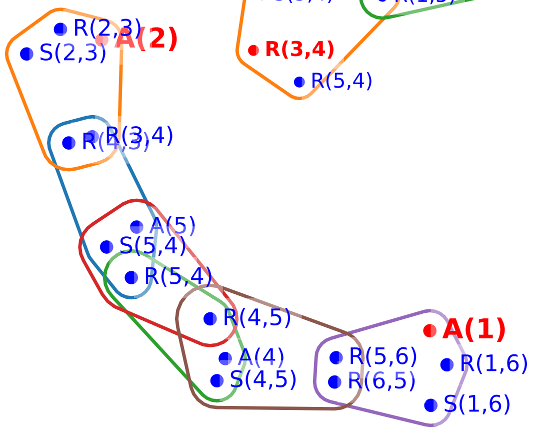
Certificate of Hardness

Finding IJPs with DLP: 5 New Hardness Gadgets

- Using the **Automatic IJP Generator** we proved 5 queries hard (out of 7 previously open from Freire+20)

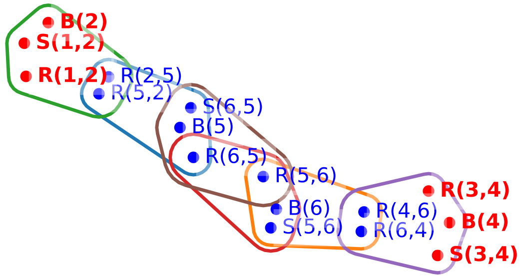
q^{S}_{3cc}:-R(x,y),R(y,z),R(w,z),S(w,z)

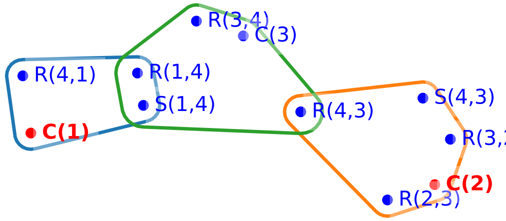




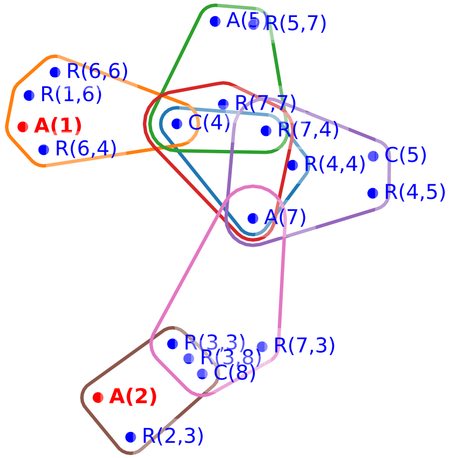
$q^{AS_{xy}}_{3perm-R}$:- A(x),S(x,y),R(x,y),R(y,z),R(z,y)

$q^{S_{xy}B}_{3perm-R}$:- S(x,y),R(x,y),B(y),R(y,z),R(z,y)





$q^{S_{xy}C}_{3perm-R}$:- S(x,y),R(x,y),R(y,z),R(z,y),C(z)



z_6: − A(x),R(x,y),R(y,y),R(y,z),C(z)

- Can recover all previous hardness results + find new ones!

Dichotomy Conjectures for Resilience

Theorem.

IJP → **NPC**

Theorem.

IJP ↔ **NPC** for SJ-Free Queries

Conjecture. [Hardness]

IJP ↔ **NPC**

Conjecture. [Hardness] Corollary

NPC → DLP finds a hardness proof

Conjecture. [Hardness]

IJP → IJP of domain size $\leq 7 * var(Q)$

Conjecture. [PTIME]

∄IJP → LP = ILP

Conjecture. [PTIME]

∄IJP → There is a flow graph that encodes resilience

Takeaways

- One unified algorithm, only need to prove PTIME
- One unified hardness criterion
  - Automatic search

Open Problems

- Which RDM problems can we solve with this unified approach?
  - Resilience
  - Causal Responsibility
  - Minimal Factorization of Provenance of CQs
  - …… **Claim: many more**

Many more details, proofs, experiments, approximations:

- https://northeastern-datalab.github.io/unified-reverse-data-management/
- Makhija, Gatterbauer. A Unified Approach for Resilience and Causal Responsibility with Integer Linear Programming (ILP) and LP Relaxations, *To appear SIGMOD 2024* (https://arxiv.org/abs/2212.08898)
- Makhija, Gatterbauer. Towards a Dichotomy for Minimally Factorizing the Provenance of Self-Join Free Conjunctive Queries, Arxiv 2021 (https://arxiv.org/abs/2105.14307)