Thanks to the Simons Institute
for the Theory of Computing

SIMONS INSTITUTE
for the Theory of Computing

https://simons.berkeley.edu/programs/logic-algorithms-database-theory-ai

DATALAB
@Northeastern

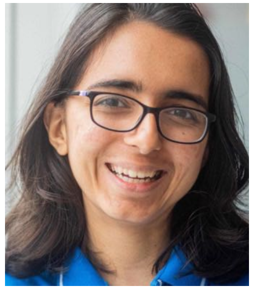
Is Integer Linear Programming All You Need for Deletion Propagation?

A unified and practical approach to Generalized Deletion Propagation

VLDB 2025 (London, Sept 4, 2025)

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

Neha Makhija
Northeastern University
(UMass Amherst from Sept 2025)

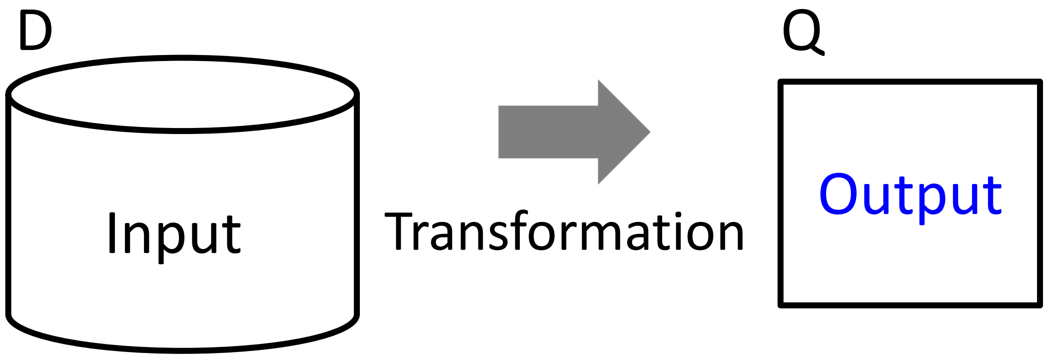


Wolfgang Gatterbauer
Northeastern University

Data Management

QUERY EVALUATION:

A transformation of the input to the output



D

Input

Transformation

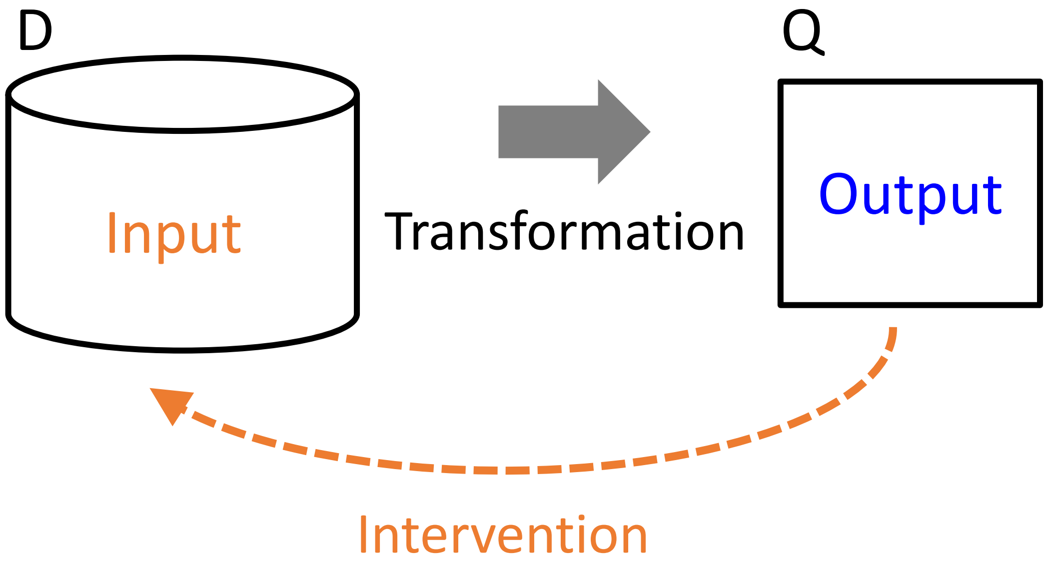
Q

Output

Reverse Data Management (RDM)

QUERY EVALUATION:

A transformation of the input to the output



REVERSE DATA MANAGEMENT (RDM):

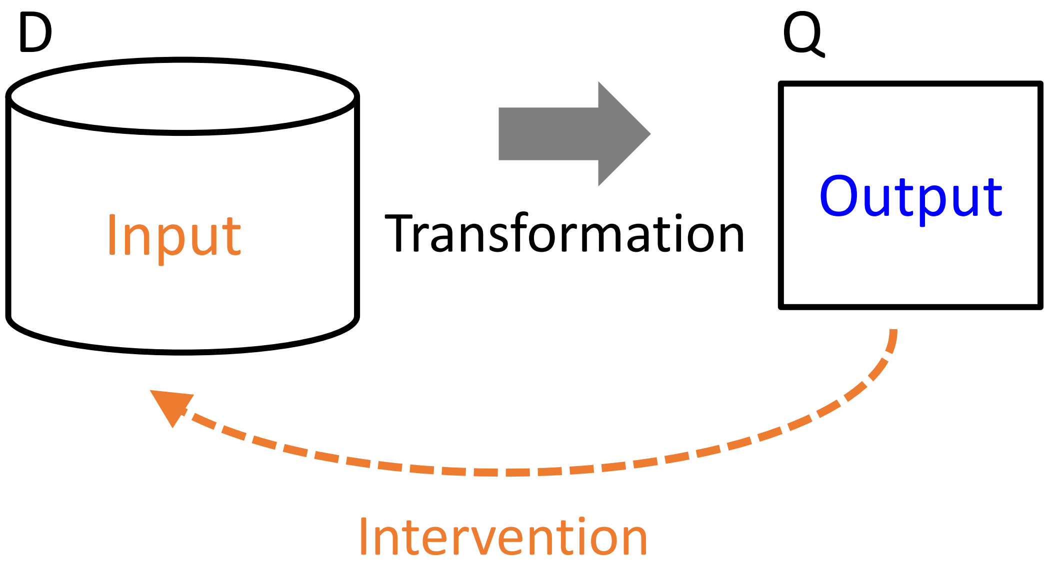
What are the required changes to the input, in order to achieve a desired output?

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

Reverse Data Management (RDM): 2 types of Explanations

QUERY EVALUATION:

A transformation of the input to the output



REVERSE DATA MANAGEMENT (RDM):

What are the required changes to the input, in order to achieve a desired output?

Two types of explanations in RDM, naming used from Explainable AI (XAI):

1. Contrastive (counterfactual):

2. Abductive (factual):

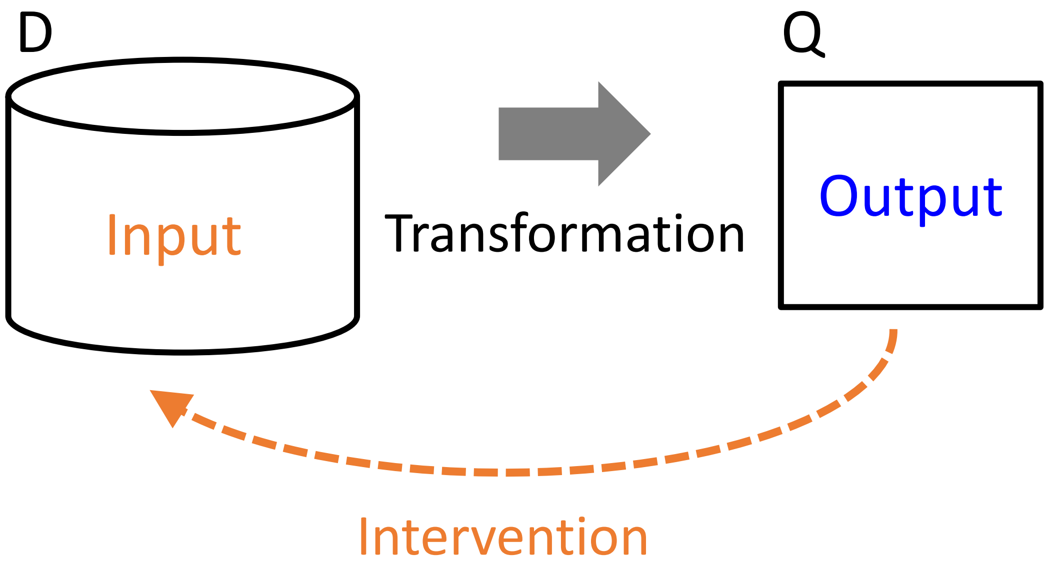
Meliou, Gatterbauer, Suciu. Reverse Data Management. VLDB 2011 Vision track. https://doi.org/10.14778/3402755.3402803

Marques-Silva. Logic-Based Explainability in Machine Learning. Reasoning Web. Causality, Explanations and Declarative Knowledge. 2023. https://doi.org/10.1007/978-3-031-31414-8_2

Reverse Data Management (RDM): 2 types of Explanations

QUERY EVALUATION:

A transformation of the input to the output



REVERSE DATA MANAGEMENT (RDM):

What are the required changes to the input, in order to achieve a desired output?

Two types of explanations in RDM, naming used from Explainable AI (XAI):

1. Contrastive (counterfactual): set of tuples (features) of min size that are sufficient to change an output (prediction)

2. Abductive (factual):

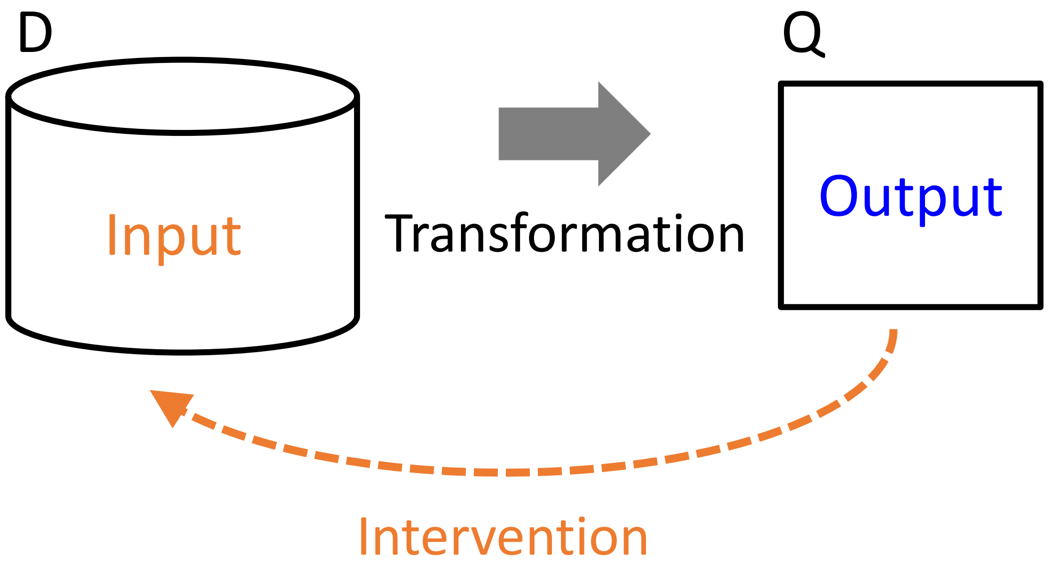
Meliou, Gatterbauer, Suciu. Reverse Data Management. VLDB 2011 Vision track. https://doi.org/10.14778/3402755.3402803

Marques-Silva. Logic-Based Explainability in Machine Learning. Reasoning Web. Causality, Explanations and Declarative Knowledge. 2023. https://doi.org/10.1007/978-3-031-31414-8_2

Reverse Data Management (RDM): 2 types of Explanations

QUERY EVALUATION:

A transformation of the input to the output



REVERSE DATA MANAGEMENT (RDM):

What are the required changes to the input, in order to achieve a desired output?

Two types of explanations in RDM, naming used from Explainable AI (XAI):

1. Contrastive (counterfactual): set of tuples (features) of min size that are sufficient to change an output (prediction)

= min deletions of tuples

(e.g., resilience, deletion propagation)

2. Abductive (factual): set of tuples (features) of min size that are sufficient for ensuring a certain output (prediction)

= max deletions of tuples

(e.g., smallest witness problem)

= DELETION PROPAGATION

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

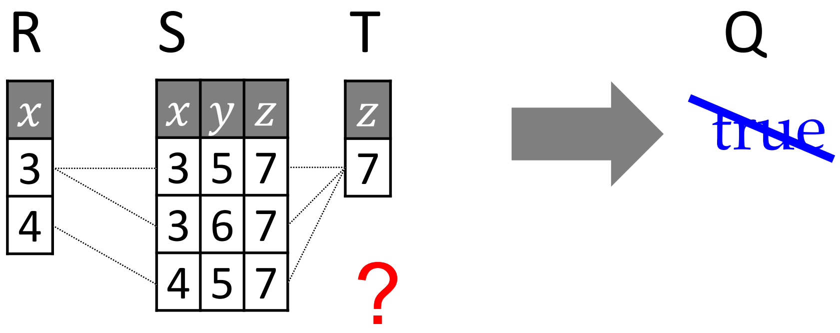
Marques-Silva. Logic-Based Explainability in Machine Learning. Reasoning Web. Causality, Explanations and Declarative Knowledge. 2023. https://doi.org/10.1007/978-3-031-31414-8_2

Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
7

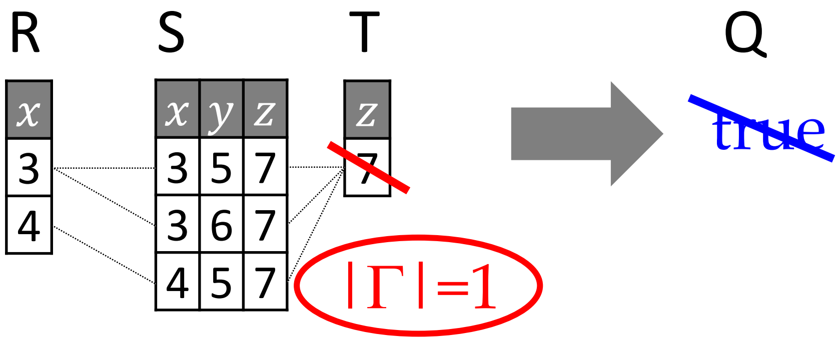
```
select exists(
  select 1
  from R, S, T
  where R.x=S.x
  and S.z=T.z)
```

Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

Q

~~true~~

$|\Gamma|=1$

```
select exists(
  select 1
  from R, S, T
  where R.x=S.x
  and S.z=T.z)
```

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

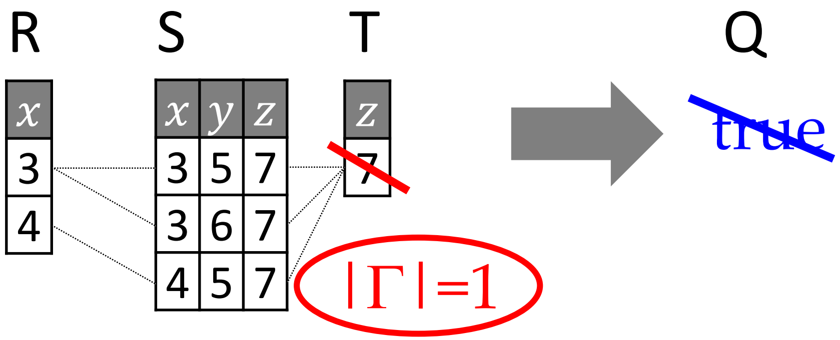
Makhija, Gatterbauer. A Unified Approach for Resilience and Causal Responsibility with Integer Linear Programming (ILP) and LP Relaxations, SIGMOD 2024. https://doi.org/10.1145/3626715

Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

Q

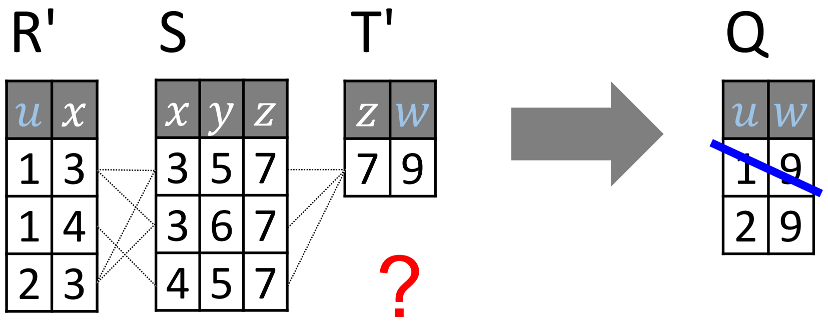
~~true~~

```
select exists(
  select 1
  from R, S, T
  where R.x=S.x
  and S.z=T.z)
```

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

2. Source side-effects (deletion propagation)

Delete min number of tuples to delete an output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

Q

u	w
~~1~~	~~9~~
2	9

```
select R'.u, T'.w
from R', S, T'
where R'.x=S.x
and S.z=T'.z
```

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

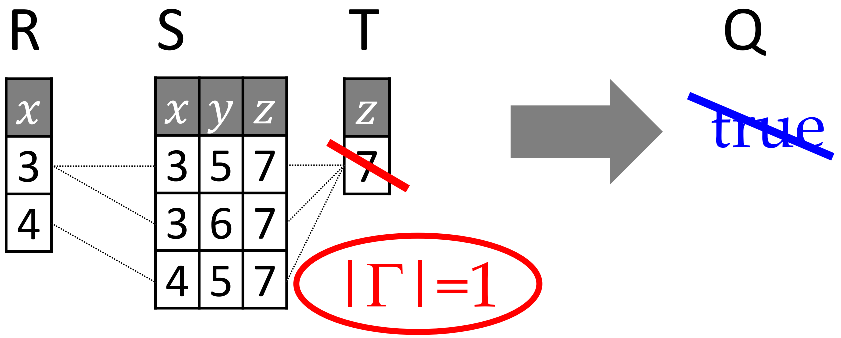
Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

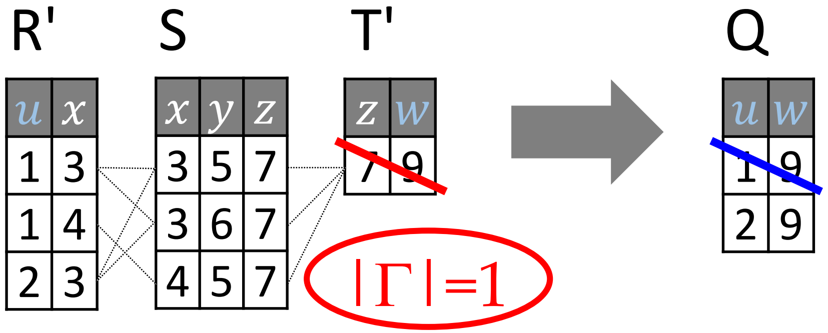
Q

~~true~~

$|\Gamma|=1$

2. Source side-effects (deletion propagation)

Delete min number of tuples to delete an output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

Q

u	w
~~1~~	~~9~~
2	9

$|\Gamma|=1$

basically identical

A long open problem: for which conjunctive queries is resilience in PTIME? (and for which NP-complete?). Only a partial classification known today

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

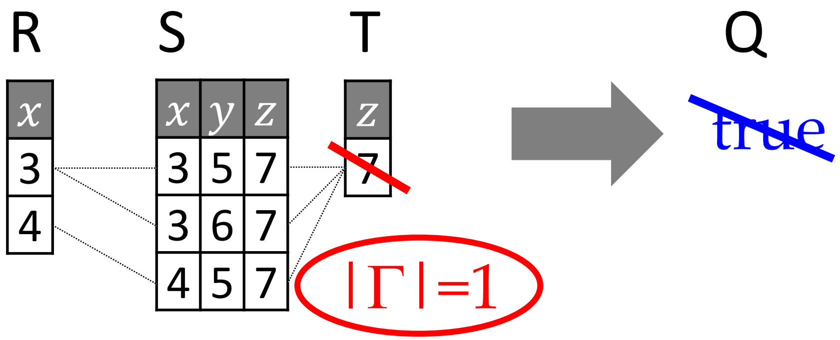
Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

1. Resilience

Delete min number of tuples to make Boolean Q false



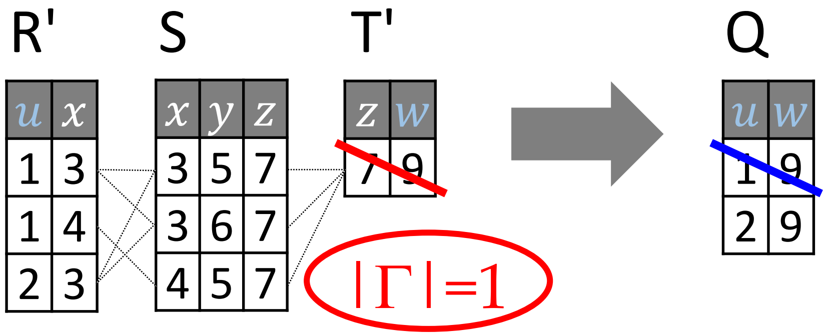
x
3
4

x	y	z
3	5	7
3	6	7
4	5	7

z
~~7~~

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



u	x
1	3
1	4
2	3

x	y	z
3	5	7
3	6	7
4	5	7

z	w
~~7~~	~~9~~

u	w
~~1~~	~~9~~
2	9

Different tractability results!

Hu, Sun, Patwa, Panigrahi, Roy. Aggregated Deletion Propagation for Counting Conjunctive Query Answers. PVLDB 2020. https://doi.org/10.14778/3425879.3425892

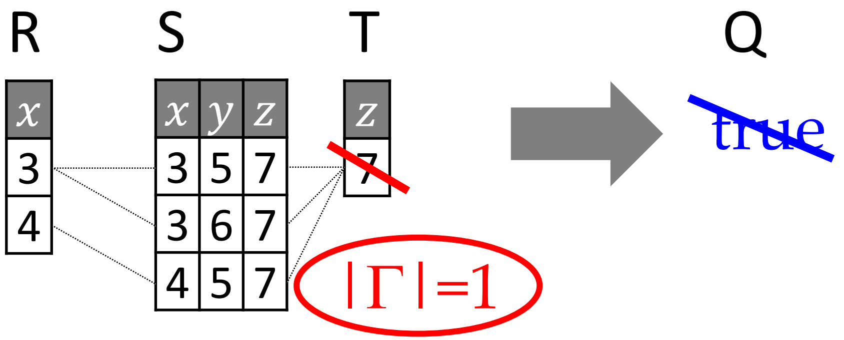
Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

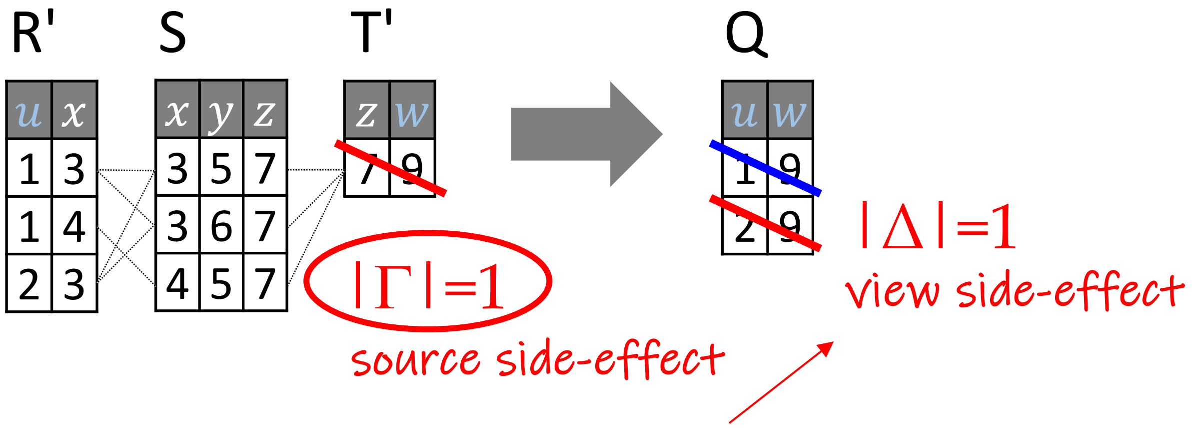
Q

~~true~~

$|\Gamma|=1$

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

Q

u	w
~~1~~	~~9~~
~~2~~	~~9~~

$|\Delta|=1$

view side-effect

$|\Gamma|=1$

source side-effect

Oops, we deleted more output tuples than needed.

That is not a concern when minimizing source-side effects.
But when minimizing view-side effects!

Hu, Sun, Patwa, Panigrahi, Roy. Aggregated Deletion Propagation for Counting Conjunctive Query Answers. PVLDB 2020. https://doi.org/10.14778/3425879.3425892

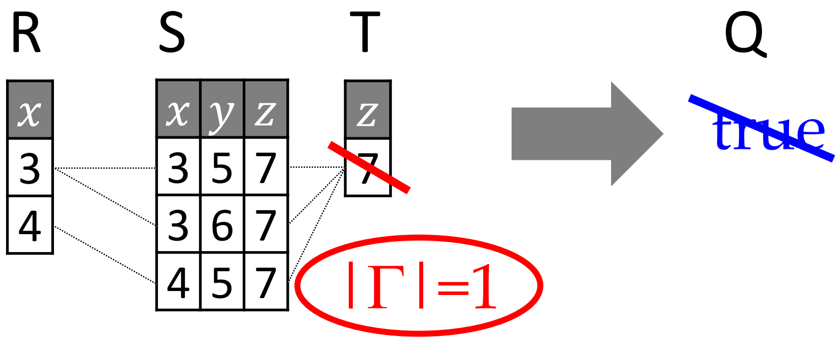
Example Reverse Data Management Problems

$Q\text{:-}R(x),S(x,y,z),T(z)$

$Q(u,w)\text{:-}R'(u,x),S(x,y,z),T'(z,w)$

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

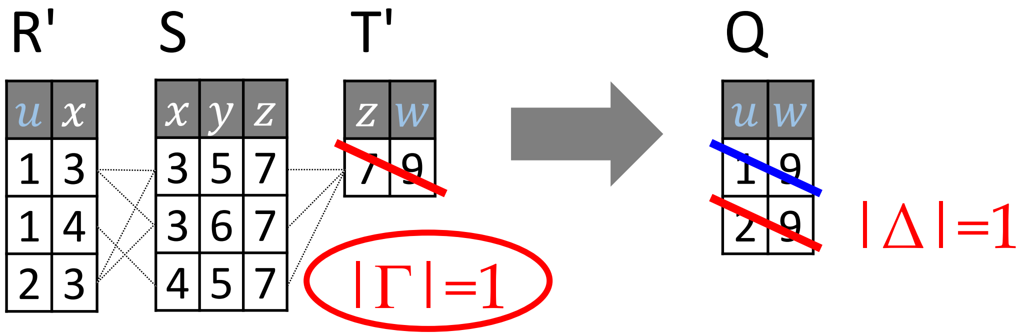
x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

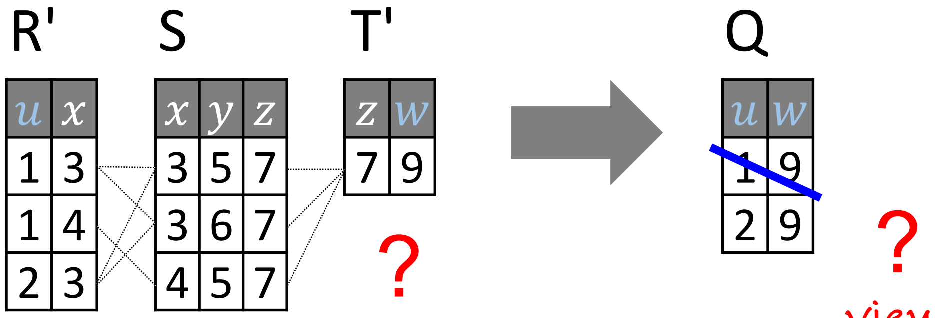
Q

u	w
~~1~~	~~9~~
~~2~~	~~9~~

source side-effect

4. View side-effects (deletion propagation)

Delete tuples in order to delete an output tuple, while minimizing the other output tuples deleted



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

Q

u	w
~~1~~	~~9~~
2	9

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
Kimelfeld, Vondrak, Williams. Maximizing conjunctive views in deletion propagation, PODS 2011, https://doi.org/10.1145/1989284.1989308

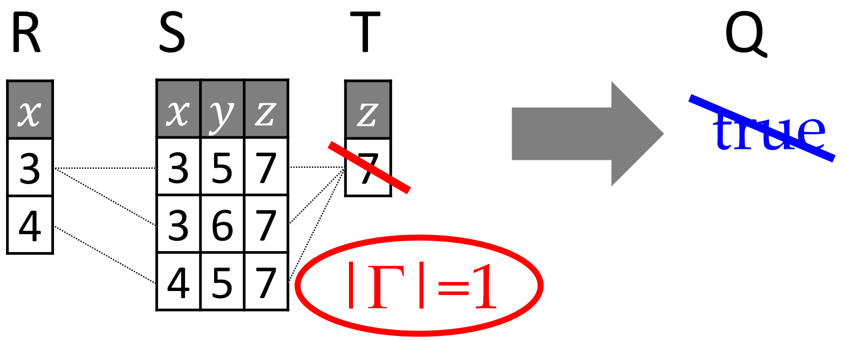
Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

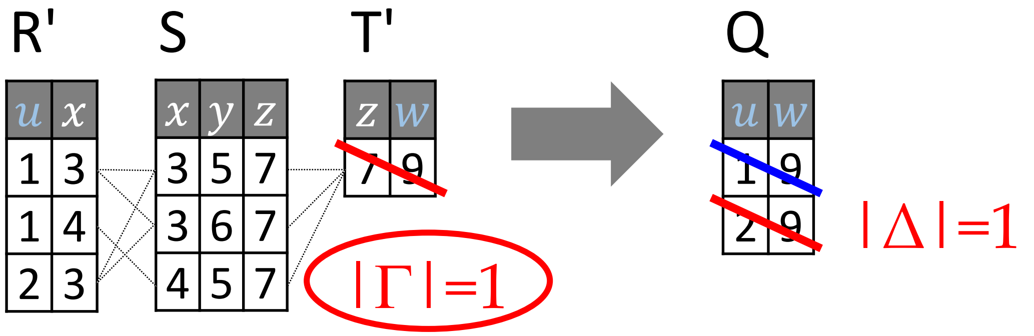
z
~~7~~

Q

~~true~~

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

Q

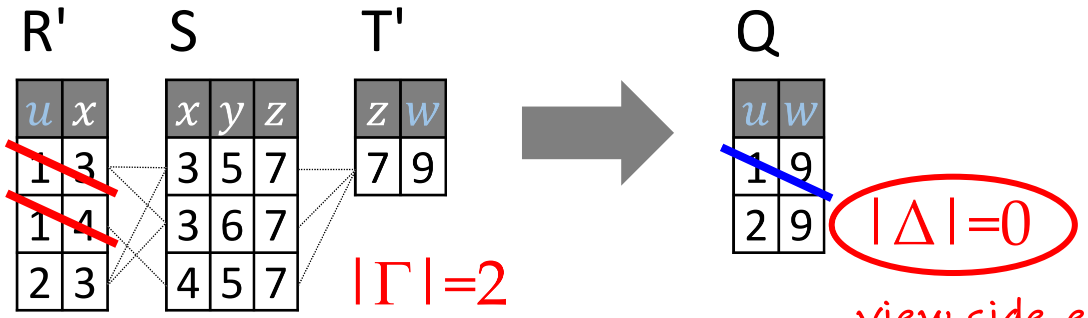
u	w
~~1~~	~~9~~
~~2~~	~~9~~

source side-effect

4. View side-effects (deletion propagation)

Delete tuples in order to delete an output tuple, while minimizing the other output tuples deleted

Again: Different tractability results!



R'

u	x
~~1~~	~~3~~
~~1~~	~~4~~
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

Q

u	w
~~1~~	~~9~~
2	9

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
Kimelfeld, Vondrak, Williams. Maximizing conjunctive views in deletion propagation, PODS 2011, https://doi.org/10.1145/1989284.1989308

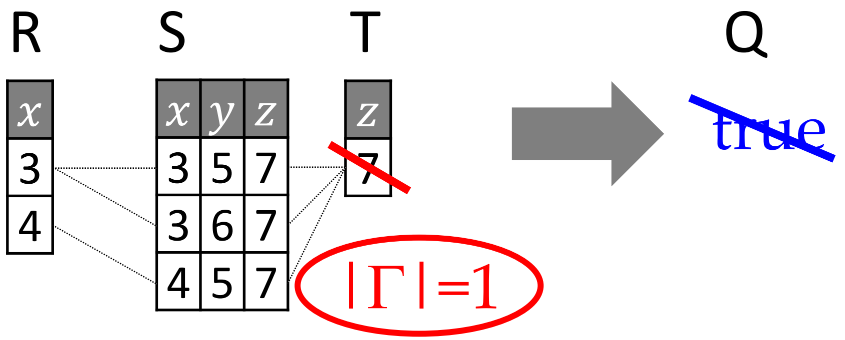
Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

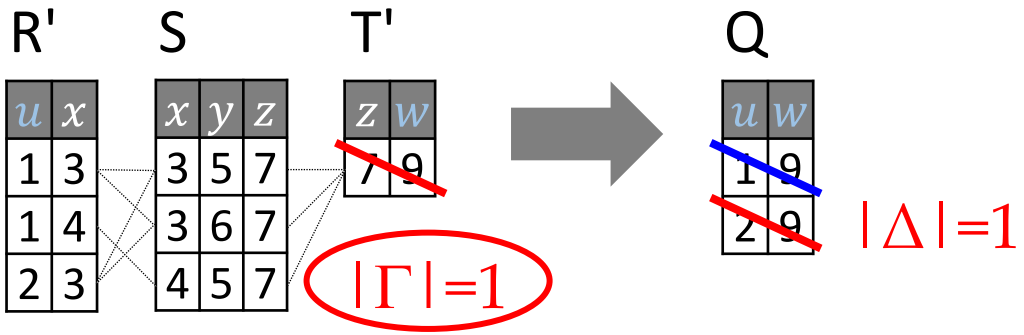
Q

~~true~~

$|\Gamma|=1$

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

Q

u	w
~~1~~	~~9~~
~~2~~	~~9~~

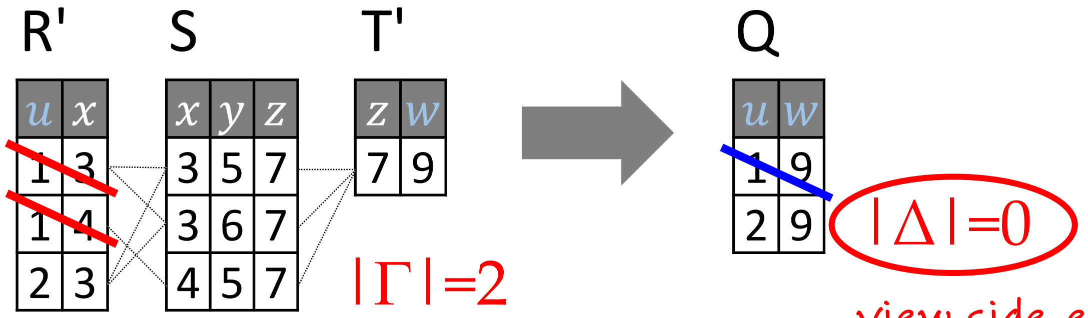
$|\Gamma|=1$

$|\Delta|=1$

source side-effect

4/5. (Aggregated) View side-effects (d.p.)

Delete tuples in order to delete ≥k output tuple,
while minimizing the other output tuples deleted



R'

u	x
~~1~~	~~3~~
~~1~~	~~4~~
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

Q

u	w
~~1~~	~~9~~
2	9

$|\Gamma|=2$

$|\Delta|=0$

view side-effect

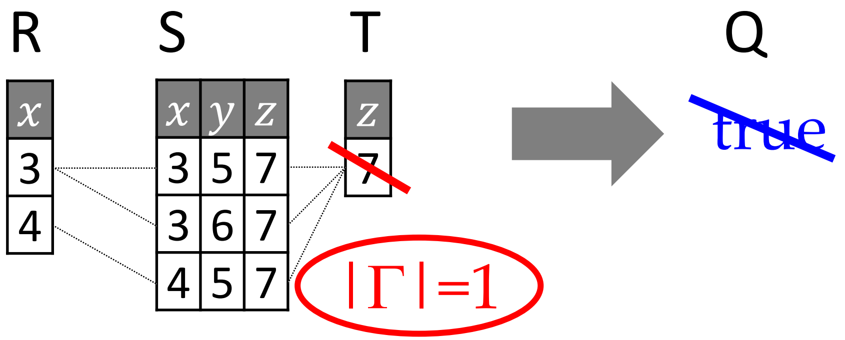
No prior work yet

Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

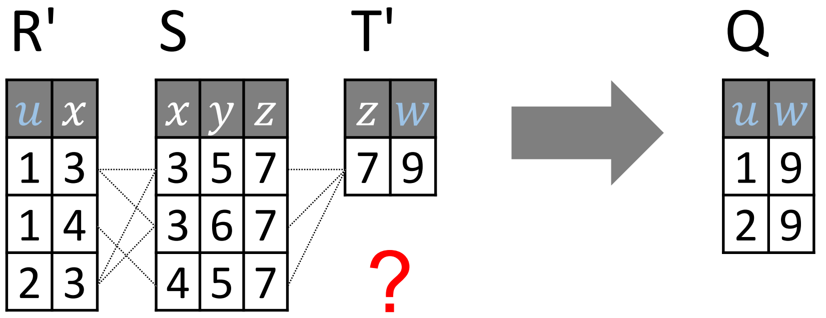
Q

~~true~~

|Γ|=1

5. Smallest witness problem

Delete max num. of tuples while keeping all output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

?

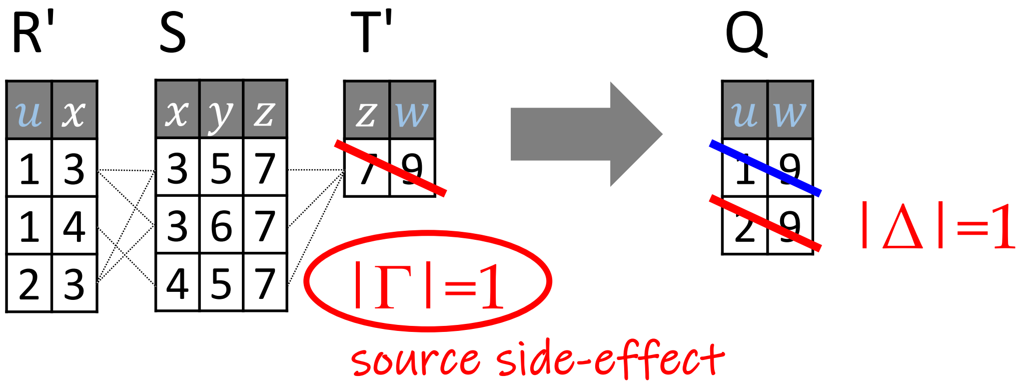
Q

u	w
1	9
2	9

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

Q

u	w
~~1~~	~~9~~
~~2~~	~~9~~

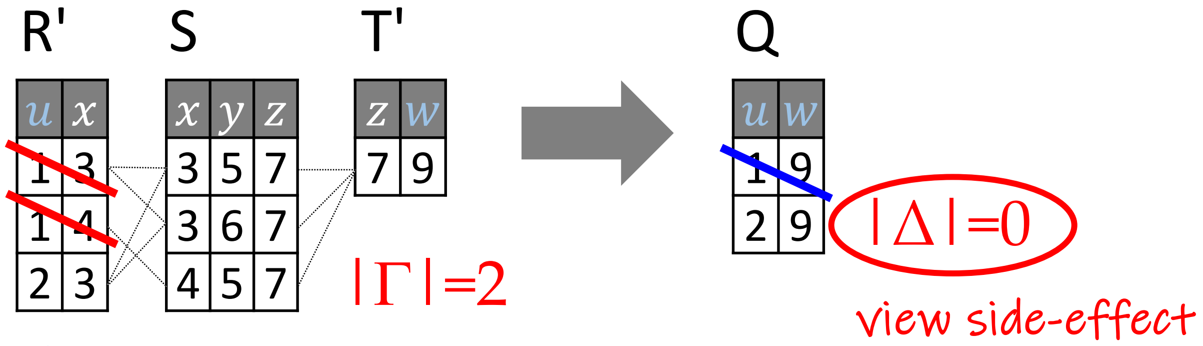
|Δ|=1

|Γ|=1

source side-effect

4/5. (Aggregated) View side-effects (d.p.)

Delete tuples in order to delete ≥k output tuple, while minimizing the other output tuples deleted



R'

u	x
~~1~~	~~3~~
~~1~~	~~4~~
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

|Γ|=2

Q

u	w
~~1~~	~~9~~
2	9

|Δ|=0

view side-effect

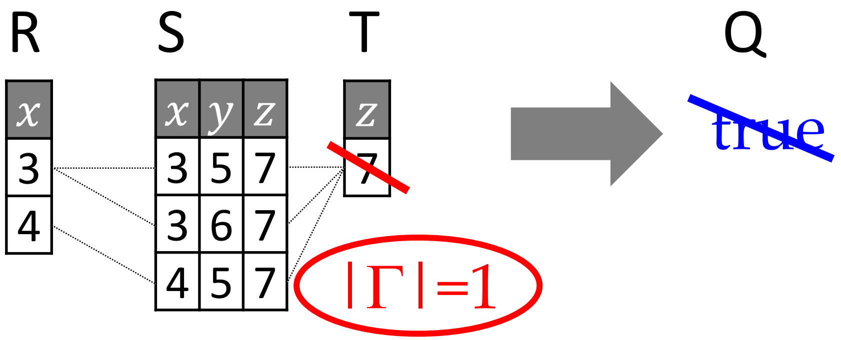
Miao, Roy, Yang. Explaining wrong queries using small examples. SIGMOD, 2019. https://doi.org/10.1145/3299869.3319866
Hu, Sintos. Finding Smallest Witnesses for Conjunctive Queries, ICDT 2024. https://doi.org/10.4230/LIPIcs.ICDT.2024.24

Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

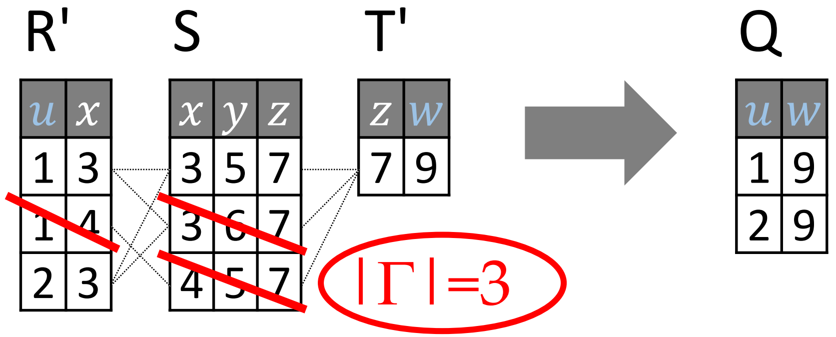
Q

~~true~~

|Γ|=1

5. Smallest witness problem

Delete max num. of tuples while keeping all output tuples



R'

u	x
1	3
~~1~~	~~4~~
2	3

S

x	y	z
3	5	7
~~3~~	~~6~~	~~7~~
~~4~~	~~5~~	~~7~~

T'

z	w
7	9

Q

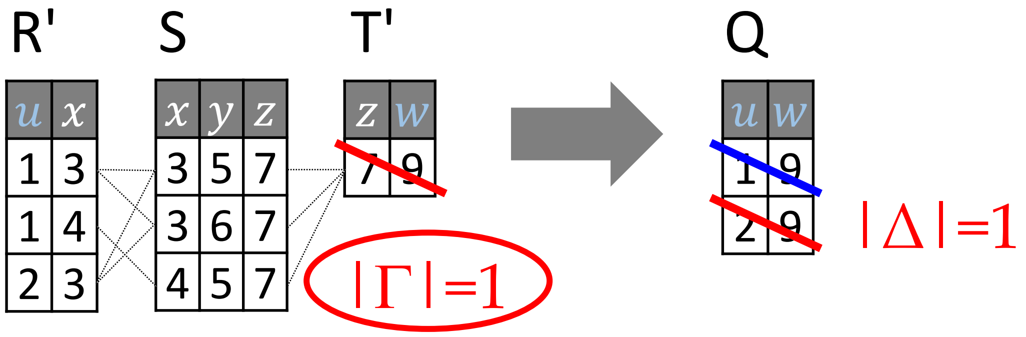
u	w
1	9
2	9

|Γ|=3

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

Q

u	w
~~1~~	~~9~~
~~2~~	~~9~~

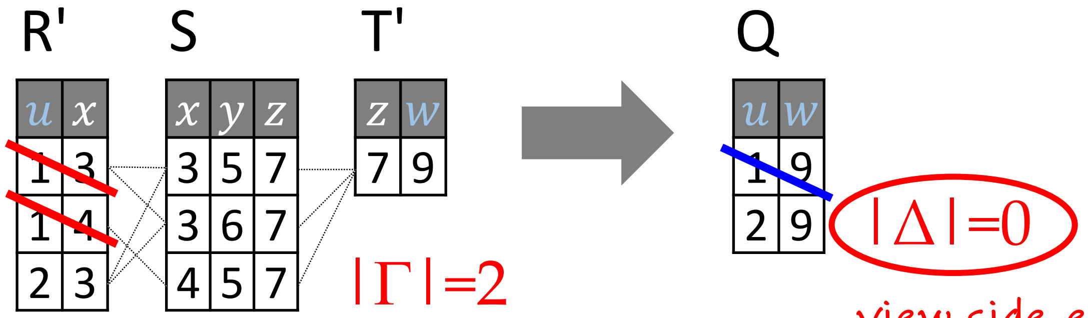
|Γ|=1

|Δ|=1

source side-effect

4/5. (Aggregated) View side-effects (d.p.)

Delete tuples in order to delete ≥k output tuple, while minimizing the other output tuples deleted



R'

u	x
~~1~~	~~3~~
~~1~~	~~4~~
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

Q

u	w
~~1~~	~~9~~
2	9

|Γ|=2

|Δ|=0

view side-effect

Miao, Roy, Yang. Explaining wrong queries using small examples. SIGMOD, 2019. https://doi.org/10.1145/3299869.3319866

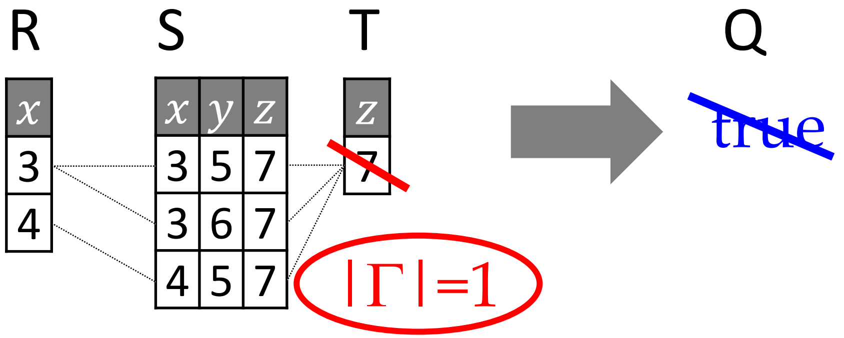
Hu, Sintos. Finding Smallest Witnesses for Conjunctive Queries, ICDT 2024. https://doi.org/10.4230/LIPIcs.ICDT.2024.24

Example Reverse Data Management Problems

Q:-R(x),S(x,y,z),T(z)

1. Resilience

Delete min number of tuples to make Boolean Q false



R

x
3
4

S

x	y	z
3	5	7
3	6	7
4	5	7

T

z
~~7~~

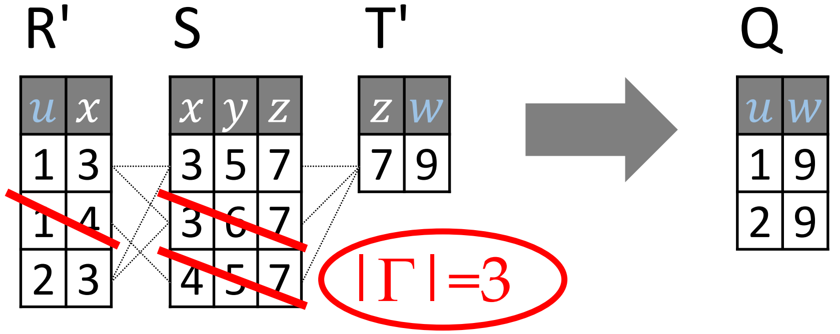
Q

~~true~~

$|\Gamma|=1$

6./7 (Aggregated) Smallest witness problem

Delete max num. of tuples while keeping ≥k output tuples



R'

u	x
1	3
~~1~~	~~4~~
2	3

S

x	y	z
3	5	7
~~3~~	~~6~~	~~7~~
~~4~~	~~5~~	~~7~~

T'

z	w
7	9

Q

u	w
1	9
2	9

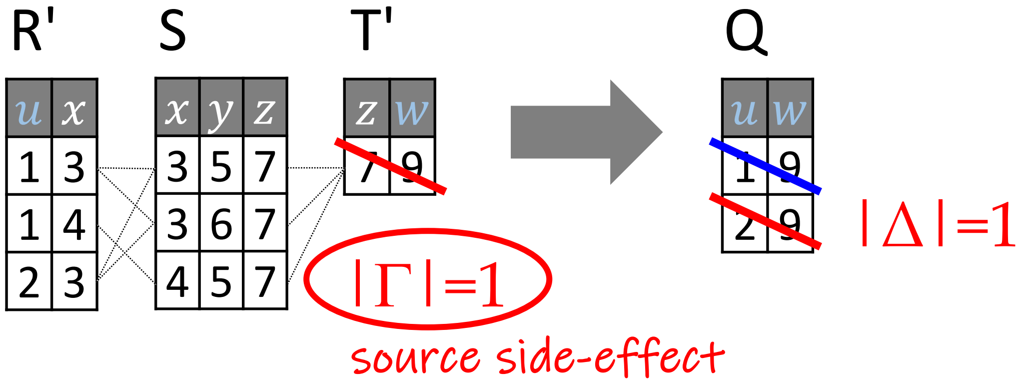
$|\Gamma|=3$

No prior work yet

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



R'

u	x
1	3
1	4
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
~~7~~	~~9~~

Q

u	w
~~1~~	~~9~~
~~2~~	~~9~~

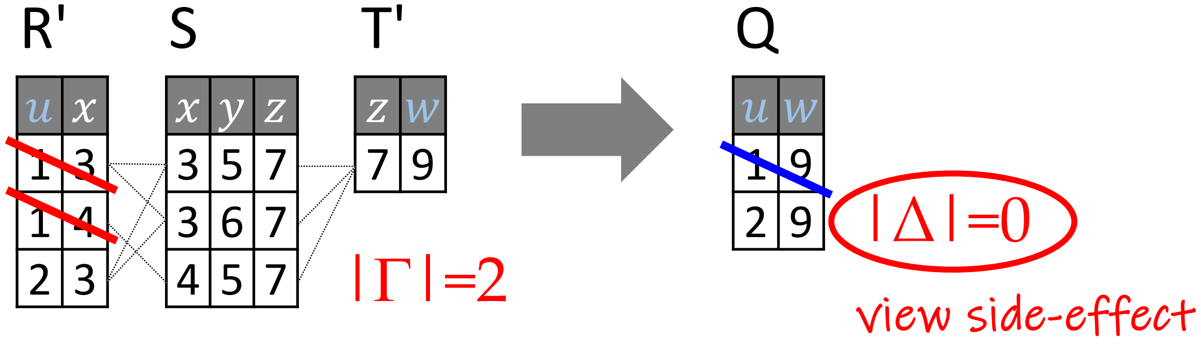
$|\Delta|=1$

$|\Gamma|=1$

source side-effect

4/5. (Aggregated) View side-effects (d.p.)

Delete tuples in order to delete ≥k output tuple, while minimizing the other output tuples deleted



R'

u	x
~~1~~	~~3~~
~~1~~	~~4~~
2	3

S

x	y	z
3	5	7
3	6	7
4	5	7

T'

z	w
7	9

Q

u	w
~~1~~	~~9~~
2	9

$|\Gamma|=2$

$|\Delta|=0$

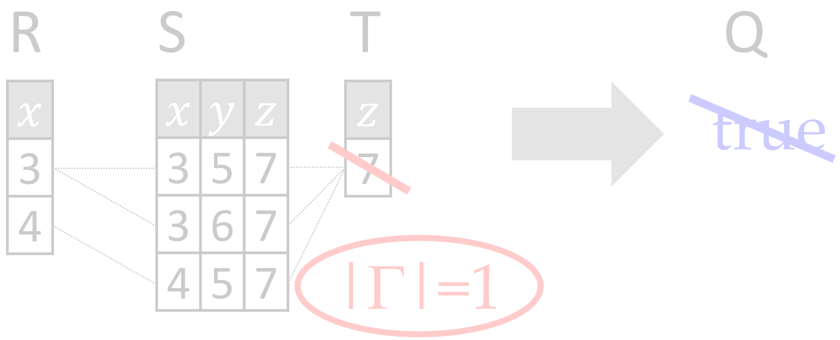
view side-effect

A Plethora of Reverse Data Management Problems ...

Q:-R(x),S(x,y,z),T(z)

1. Resilience

Delete min number of tuples to make Boolean Q false



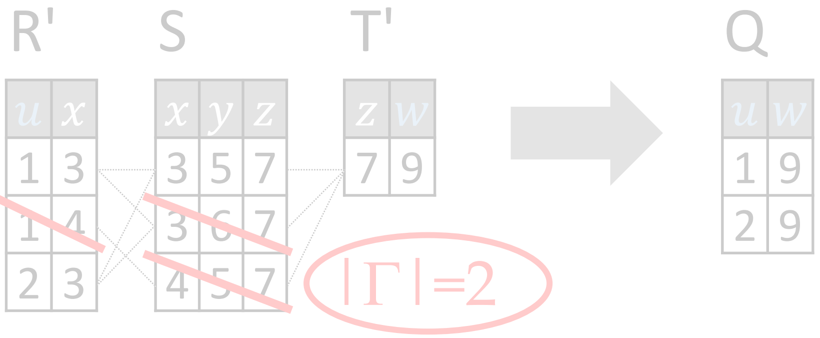
x
3
4

x	y	z
3	5	7
3	6	7
4	5	7

z
~~7~~

6./7 (Aggregated) Smallest witness problem

Delete max num. of tuples while keeping ≥k output tuples



u	x
1	3
~~1~~	~~4~~
2	3

x	y	z
3	5	7
~~3~~	~~6~~	~~7~~
~~4~~	~~5~~	~~7~~

z	w
7	9

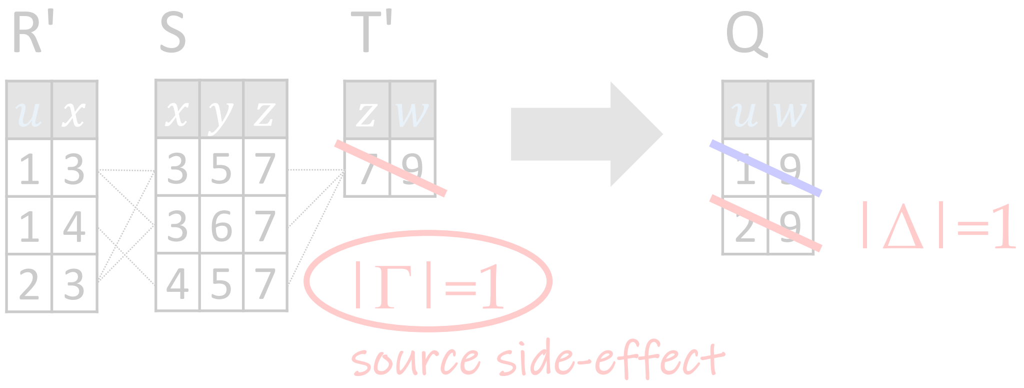
u	w
1	9
2	9

?

Q(u,w):-R'(u,x),S(x,y,z),T'(z,w)

2/3. (Aggregated) Source side-effects (d.p.)

Delete min number of tuples to delete ≥k output tuples



u	x
1	3
1	4
2	3

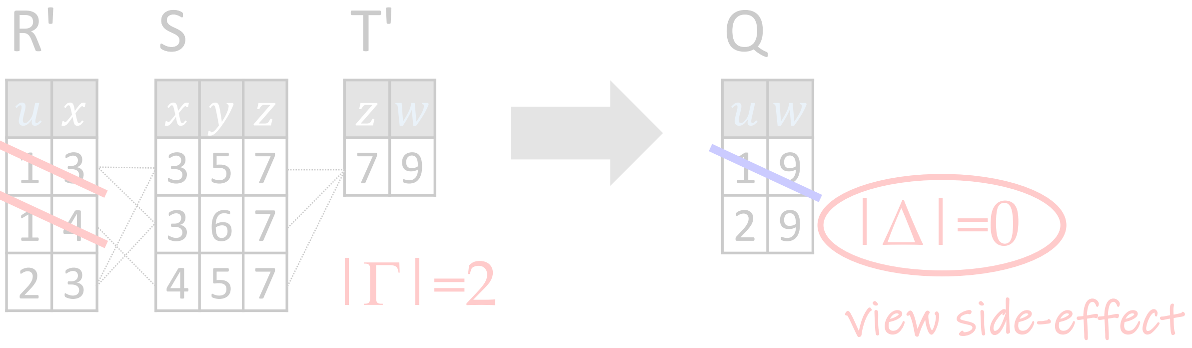
x	y	z
3	5	7
3	6	7
4	5	7

z	w
~~7~~	~~9~~

u	w
~~1~~	~~9~~
~~2~~	~~9~~

4/5. (Aggregated) View side-effects (d.p.)

Delete tuples in order to delete ≥k output tuple, while minimizing the other output tuples deleted



u	x
~~1~~	~~3~~
~~1~~	~~4~~
2	3

x	y	z
3	5	7
3	6	7
4	5	7

z	w
7	9

u	w
~~1~~	~~9~~
2	9

Our topic today: Generalized Deletion Propagation

1. Unify and generalize these problems as instances of "generalized deletion propagation"
    - also allows new problem variants

2. Propose one **ILP encoding** to solve all these problems
    - including difficult cases, such as self-joins, or bag semantics

3. The ILP formulation **is solvable in PTIME for all known PTIME cases**
    - including all known PTIME cases for the problems of: **resilience** [VLDB'15], **aggregated deletion propagation** [VLDB'20], **view-side effects** [PODS'11], **smallest witness problem** [ICDT'24], under functional dependencies, and both set and bag semantics (where results are known)

Outline

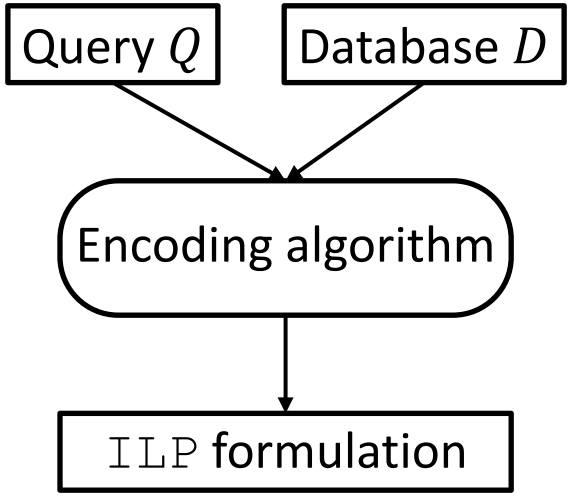
1. Reverse Data Management (RDM)
2. A magical ILP formulation
3. Take-aways

(An illustrated example: only if time remains)

Unified Algorithms for Reverse Data Management

`ILP` formulation:

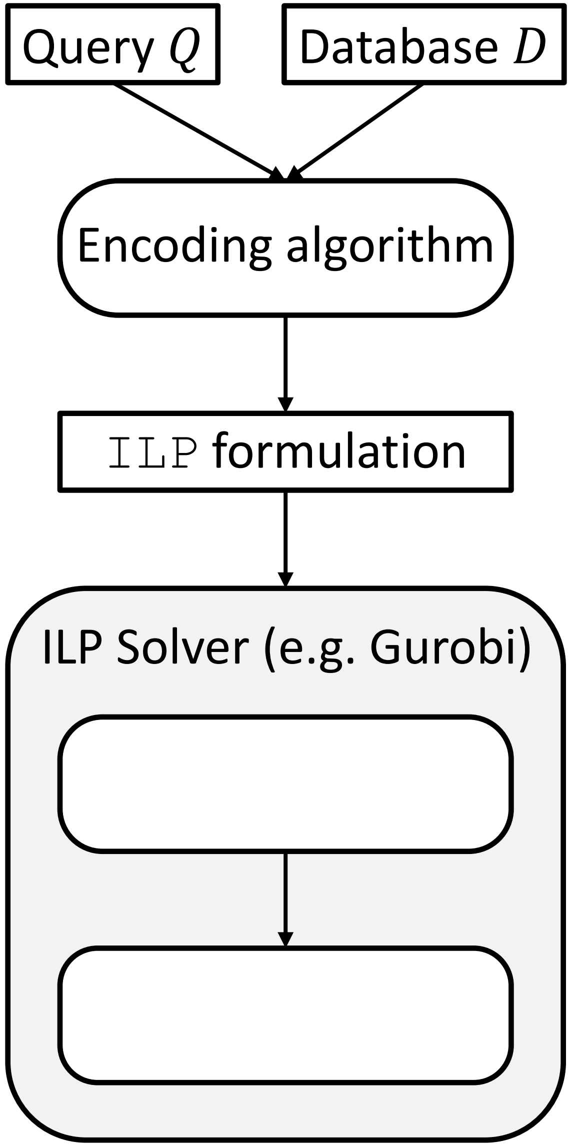
$$
\begin{aligned}
f^* = \min[\mathbf{c} \cdot \mathbf{x}] \\
\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b} \\
\mathbf{x} \in \mathbb{N}^n
\end{aligned}
$$



Unified Algorithms for Reverse Data Management

`ILP` formulation:

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{N}^n$$


Query Q

Database D

Encoding algorithm

`ILP` formulation

ILP Solver (e.g. Gurobi)

Unified Algorithms for Reverse Data Management

`ILP` formulation:

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$

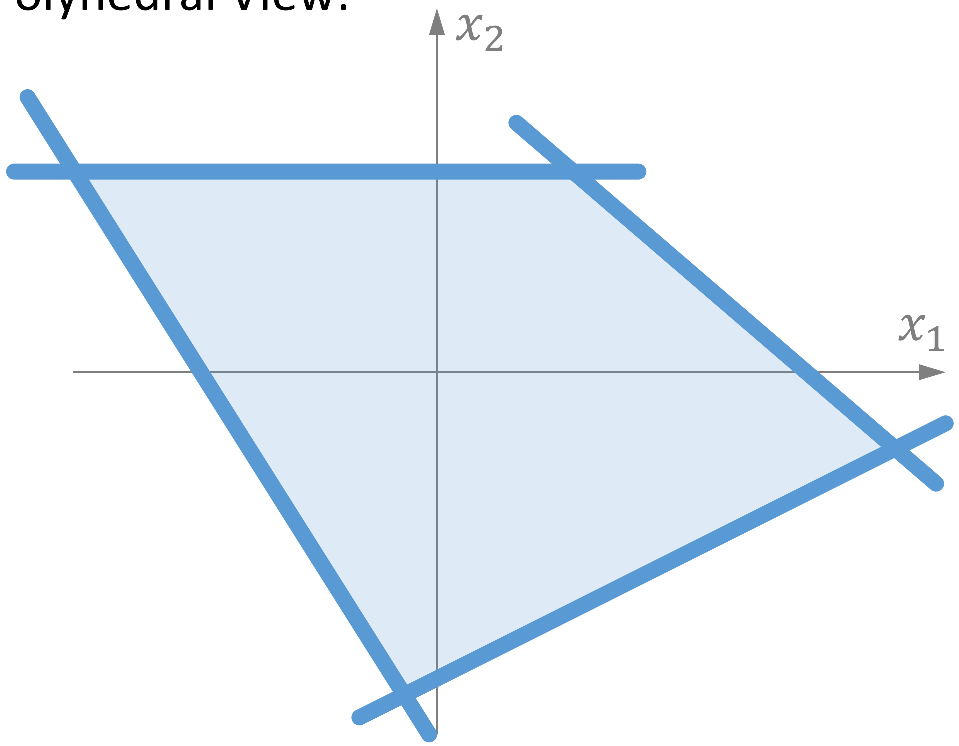
constraint vector

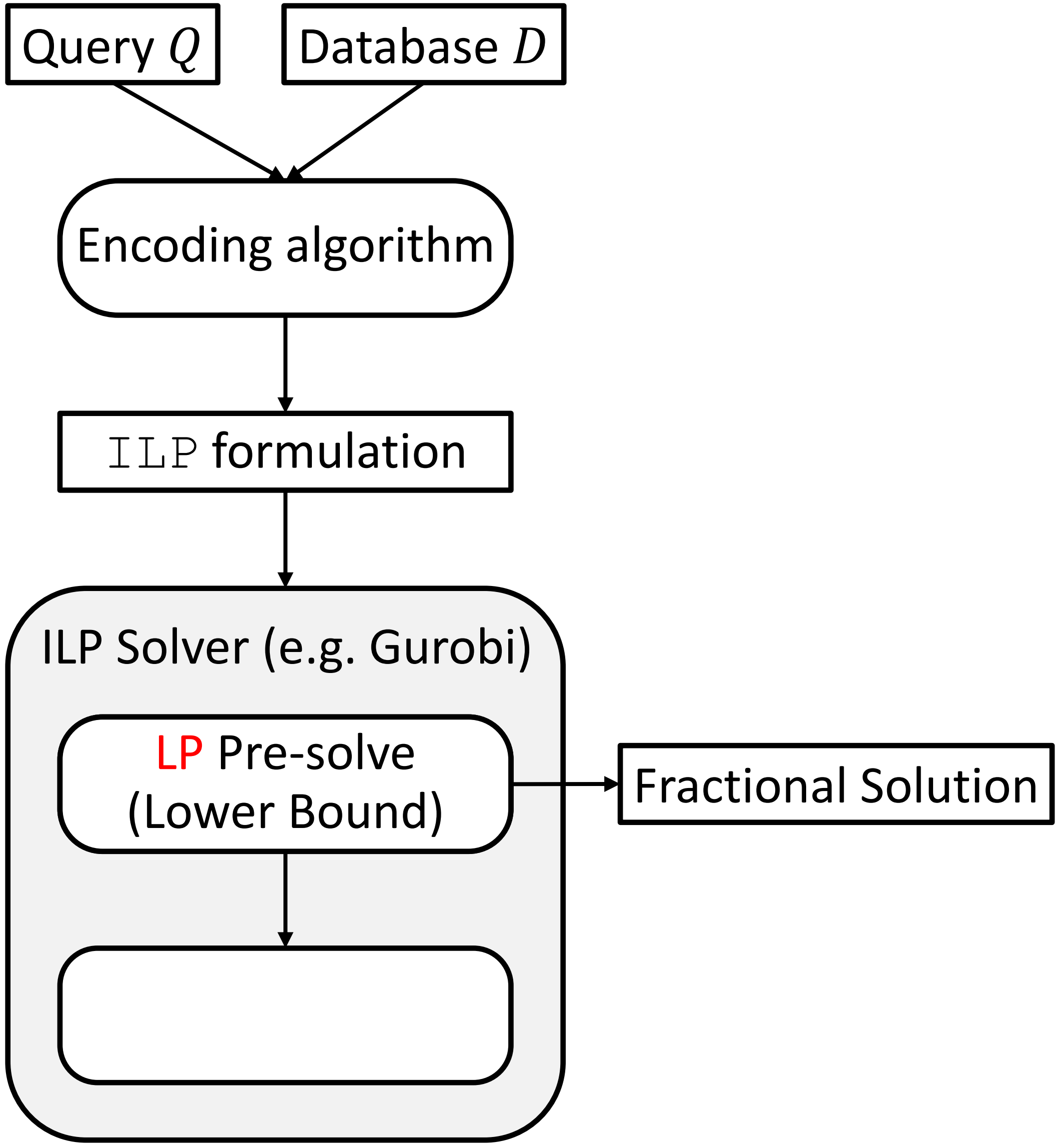
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$

constraint matrix

$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View:





Query Q

Database D

Encoding algorithm

`ILP` formulation

ILP Solver (e.g. Gurobi)

LP Pre-solve
(Lower Bound)

Fractional Solution

Unified Algorithms for Reverse Data Management

`ILP` formulation:

objective vector

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$

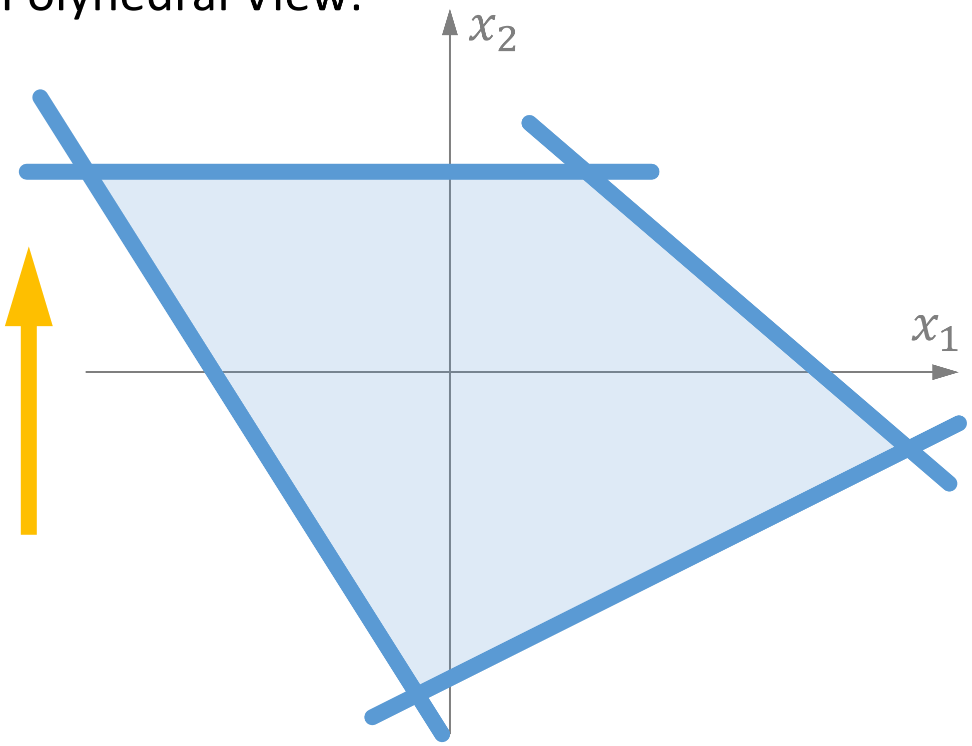
constraint vector

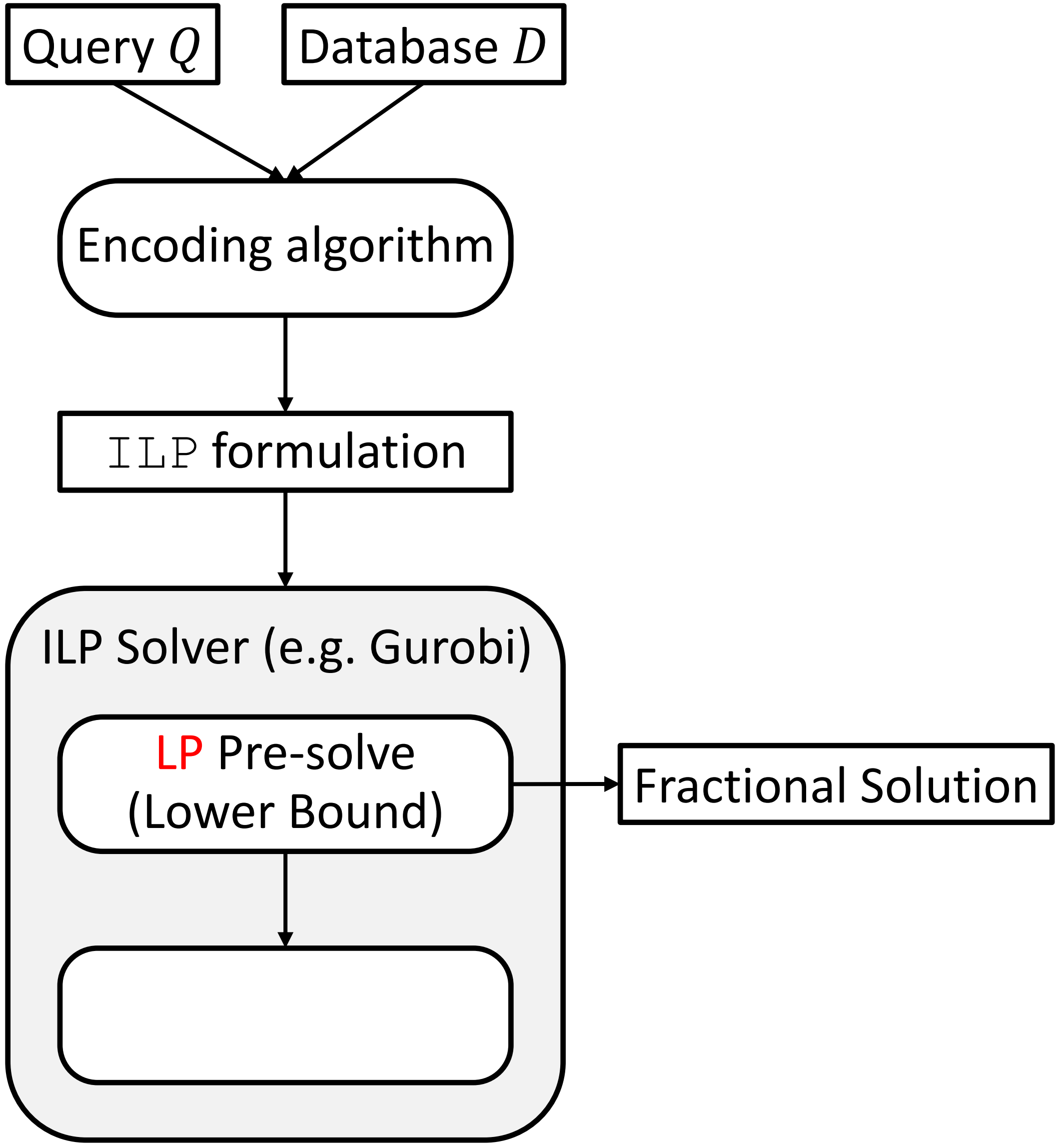
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$

constraint matrix

$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View:





Query Q

Database D

Encoding algorithm

`ILP` formulation

ILP Solver (e.g. Gurobi)

LP Pre-solve
(Lower Bound)

Fractional Solution

Unified Algorithms for Reverse Data Management

`ILP` formulation:

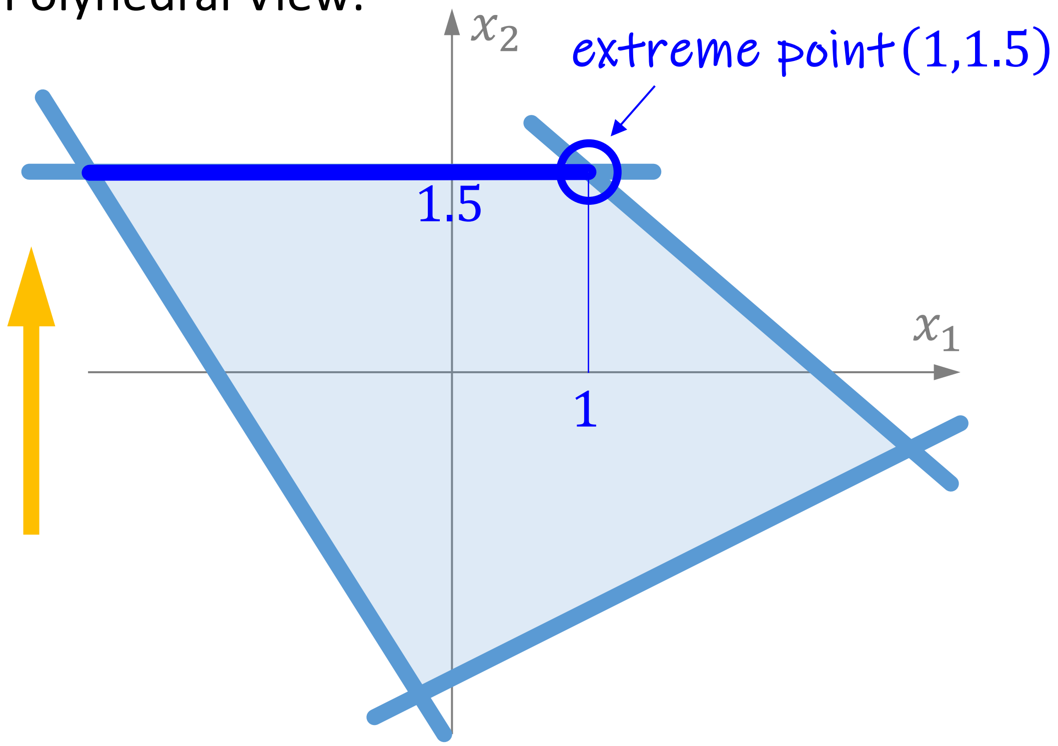
objective vector

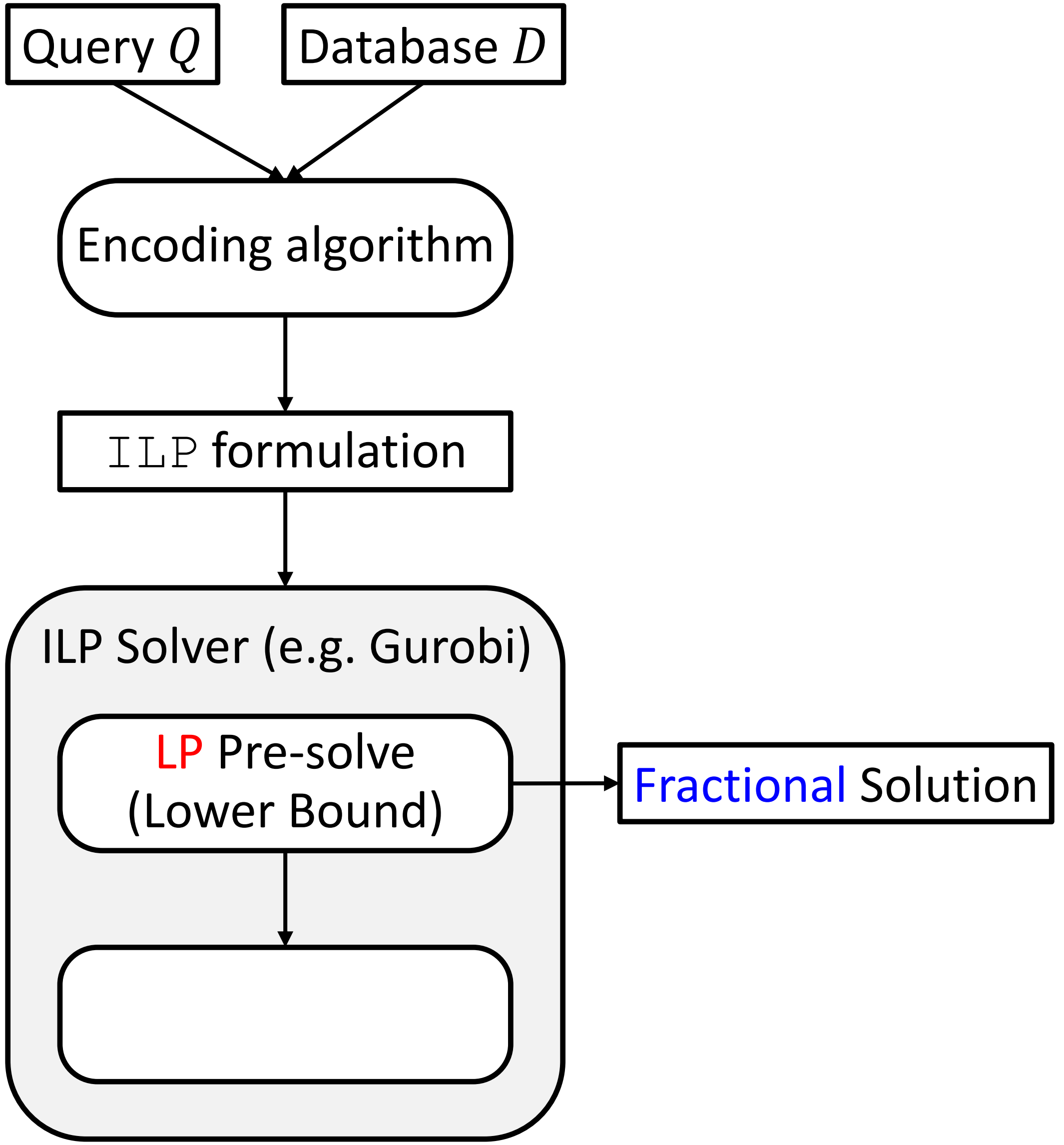
constraint vector

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

constraint matrix

Polyhedral View:




Query Q

Database D

Encoding algorithm

`ILP` formulation

ILP Solver (e.g. Gurobi)

LP Pre-solve
(Lower Bound)

Fractional Solution

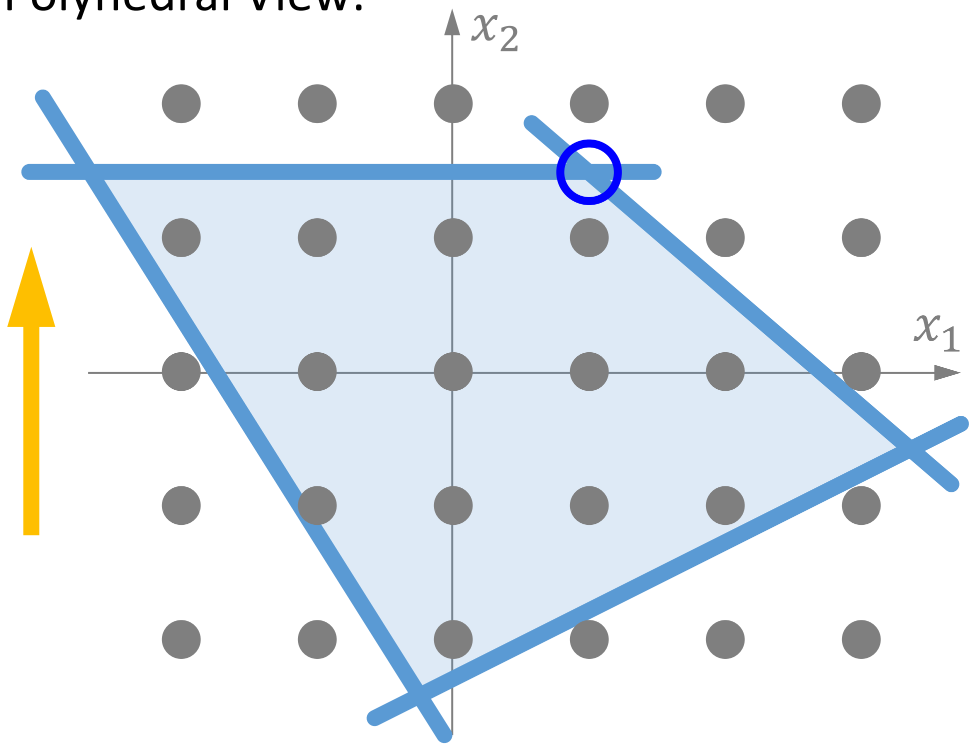
f^*=1.5
e.g. $\mathbf{x}^*$=(1,1.5)

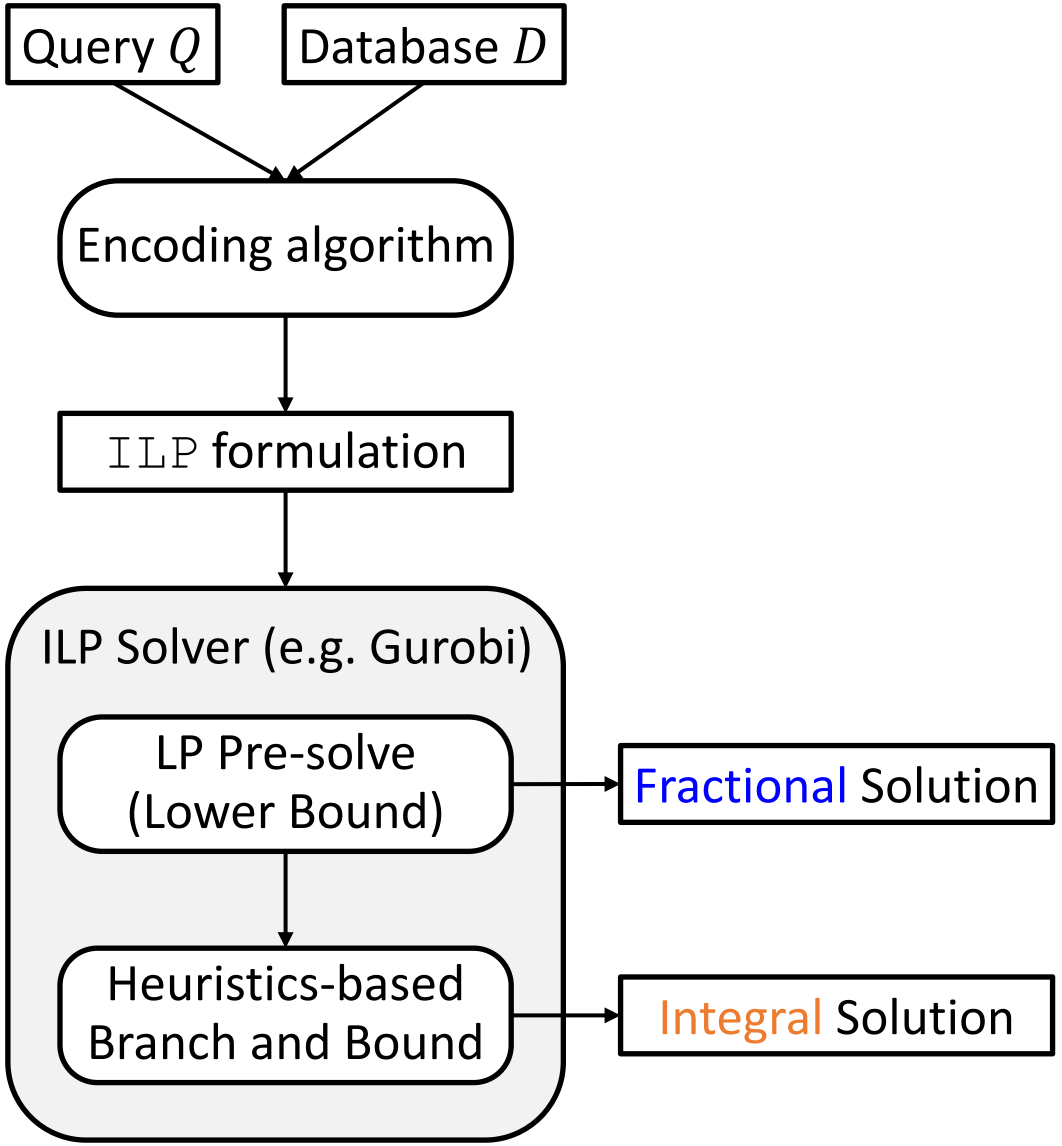
Unified Algorithms for Reverse Data Management

`ILP` formulation:

$$\begin{aligned} f^* &= \min[\mathbf{c}\cdot\mathbf{x}] \\ \text{s.t. } & \mathbf{A}\cdot\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \in \mathbb{N}^n/\mathbb{R}^n \end{aligned}$$

Polyhedral View:





Query Q

Database D

Encoding algorithm

`ILP` formulation

ILP Solver (e.g. Gurobi)

LP Pre-solve
(Lower Bound)

Heuristics-based
Branch and Bound

Fractional Solution

Integral Solution

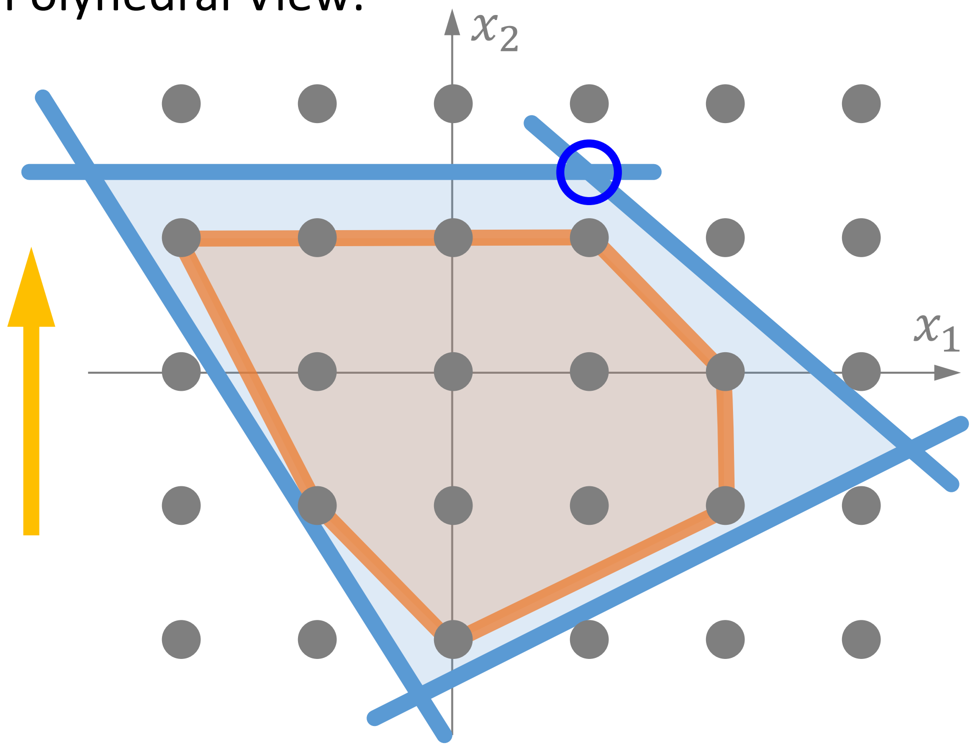
f^*=1.5
e.g. $\mathbf{x}^*$=(1,1.5)

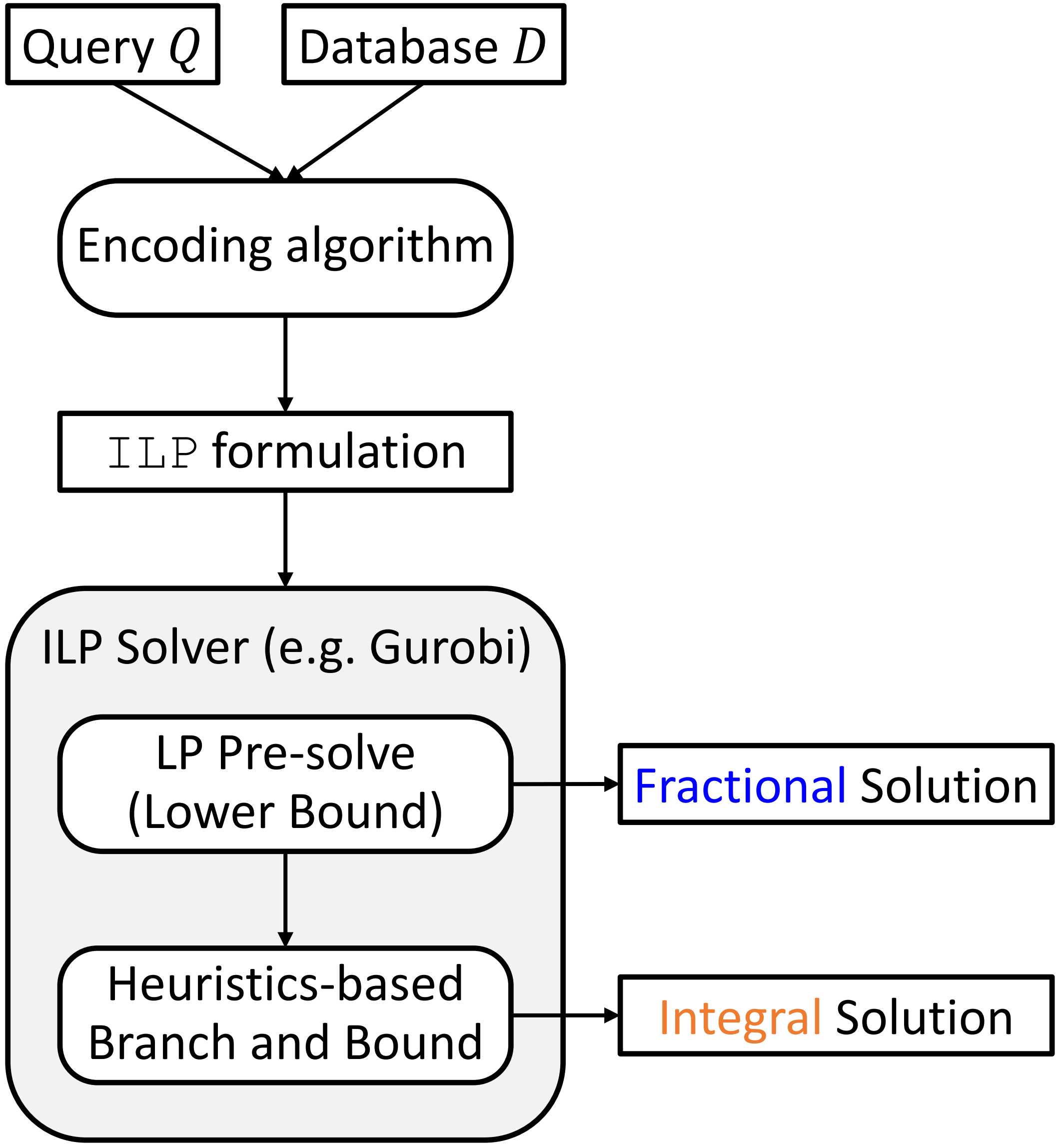
Unified Algorithms for Reverse Data Management

`ILP` formulation:

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View:





Query Q

Database D

Encoding algorithm

`ILP` formulation

ILP Solver (e.g. Gurobi)

LP Pre-solve (Lower Bound)

Heuristics-based Branch and Bound

Fractional Solution

Integral Solution

f^*=1.5
e.g. $\mathbf{x}^*$=(1,1.5)

Unified Algorithms for Reverse Data Management

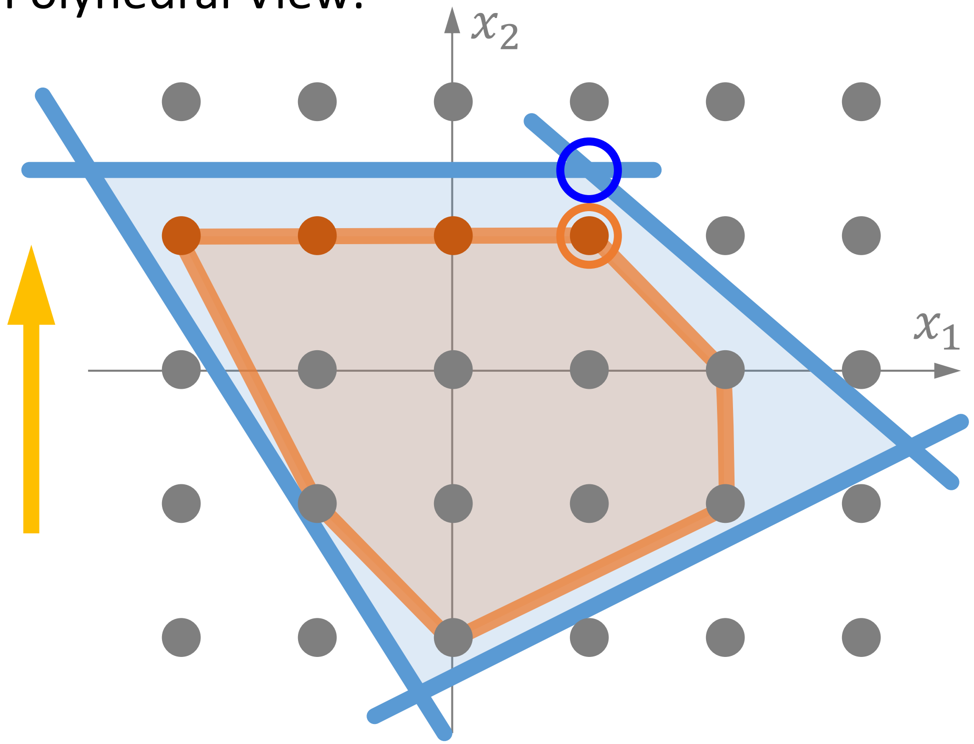
`ILP` formulation:

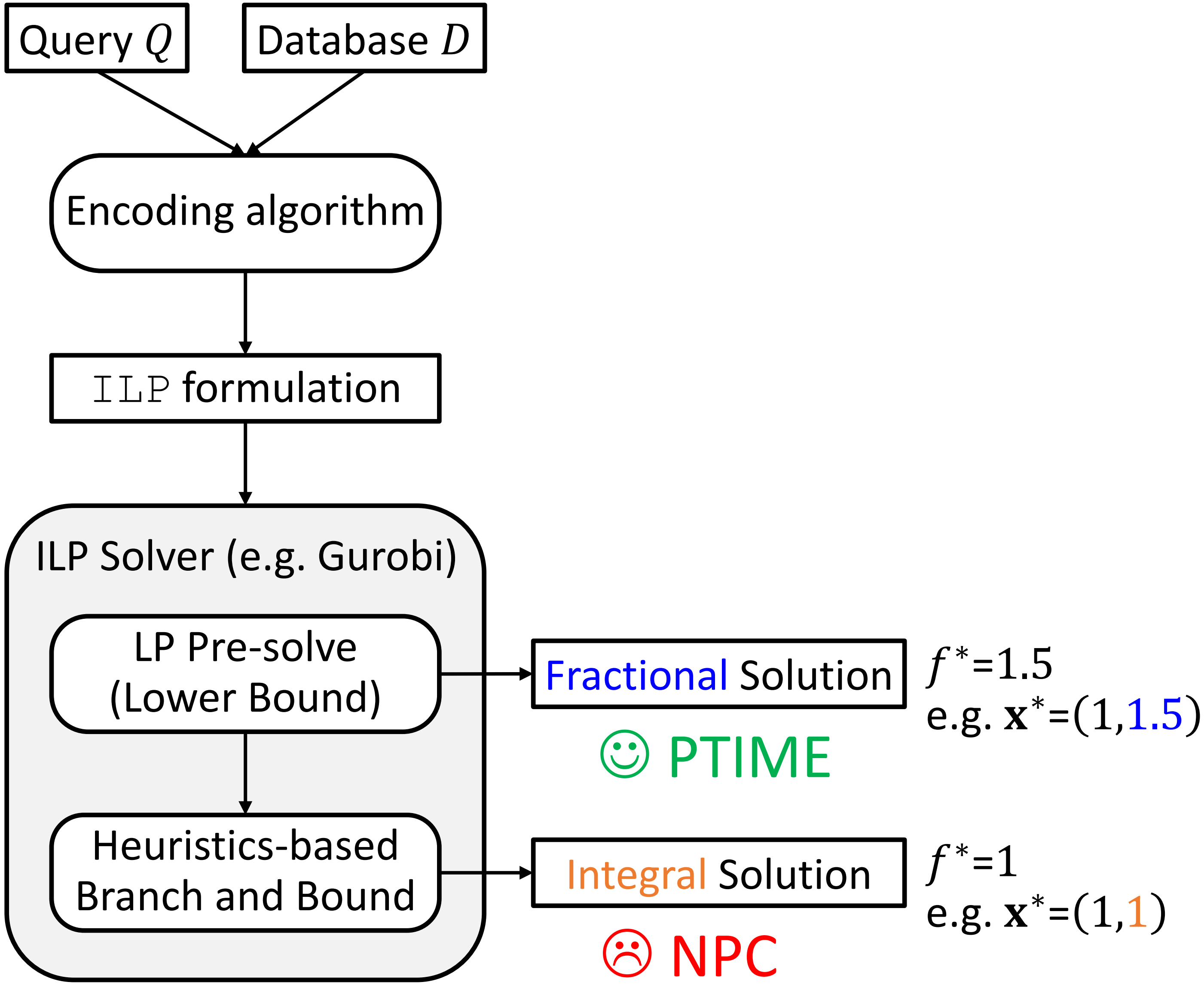
$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$

$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$

$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View:





Query Q

Database D

Encoding algorithm

`ILP` formulation

ILP Solver (e.g. Gurobi)

LP Pre-solve (Lower Bound)

Heuristics-based Branch and Bound

Fractional Solution

f^*=1.5

e.g. $\mathbf{x}^*$=(1,1.5)

☺ PTIME

Integral Solution

f^*=1

e.g. $\mathbf{x}^*$=(1,1)

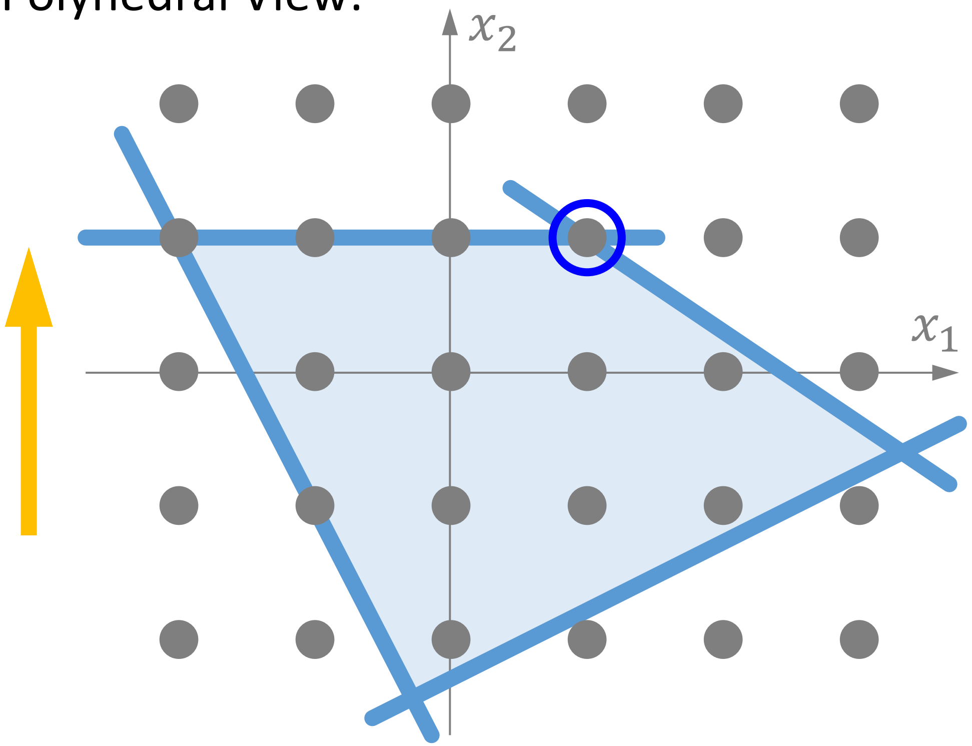
☹ NPC

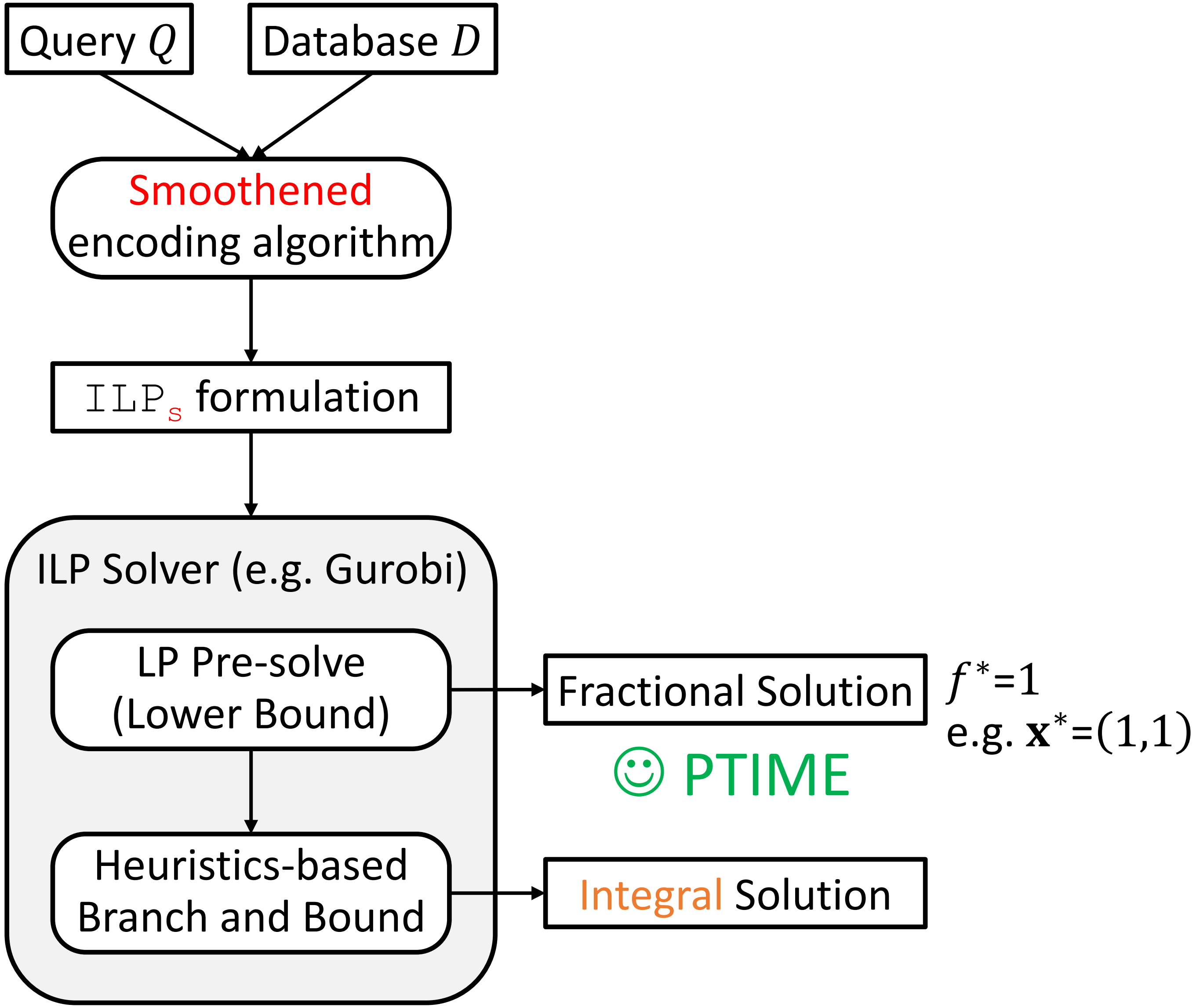
Unified Algorithms for Reverse Data Management

`ILP` formulation:

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View:





Query Q

Database D

Smoothened encoding algorithm

$\texttt{ILP}_s$ formulation

ILP Solver (e.g. Gurobi)

LP Pre-solve (Lower Bound)

Heuristics-based Branch and Bound

Fractional Solution

f^*=1
e.g. $\mathbf{x}^*$=(1,1)

☺ PTIME

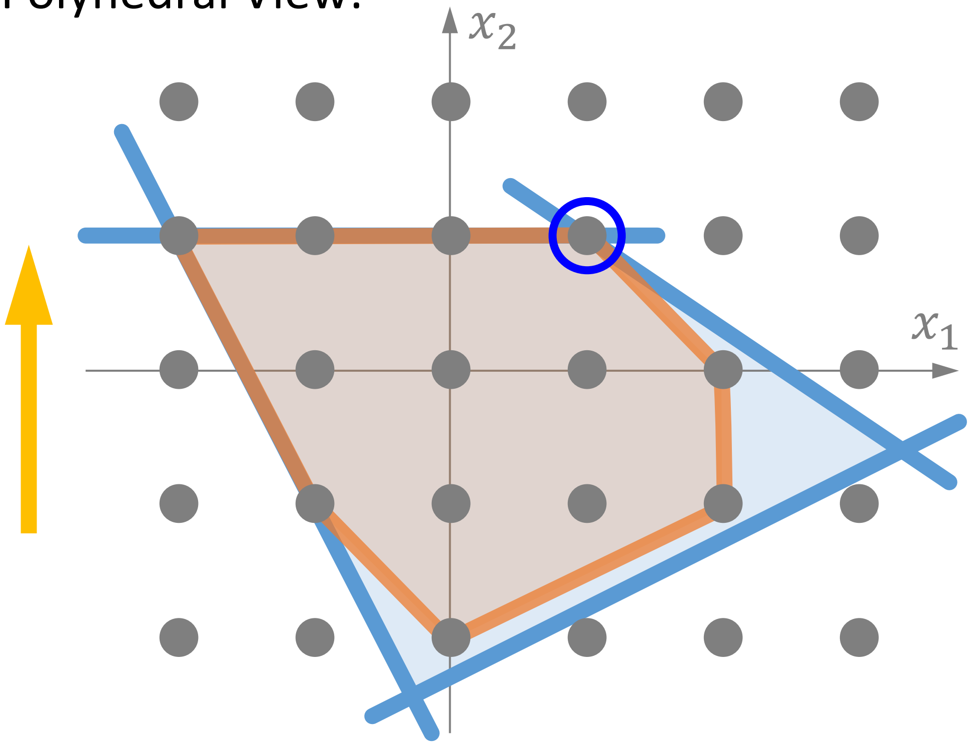
Integral Solution

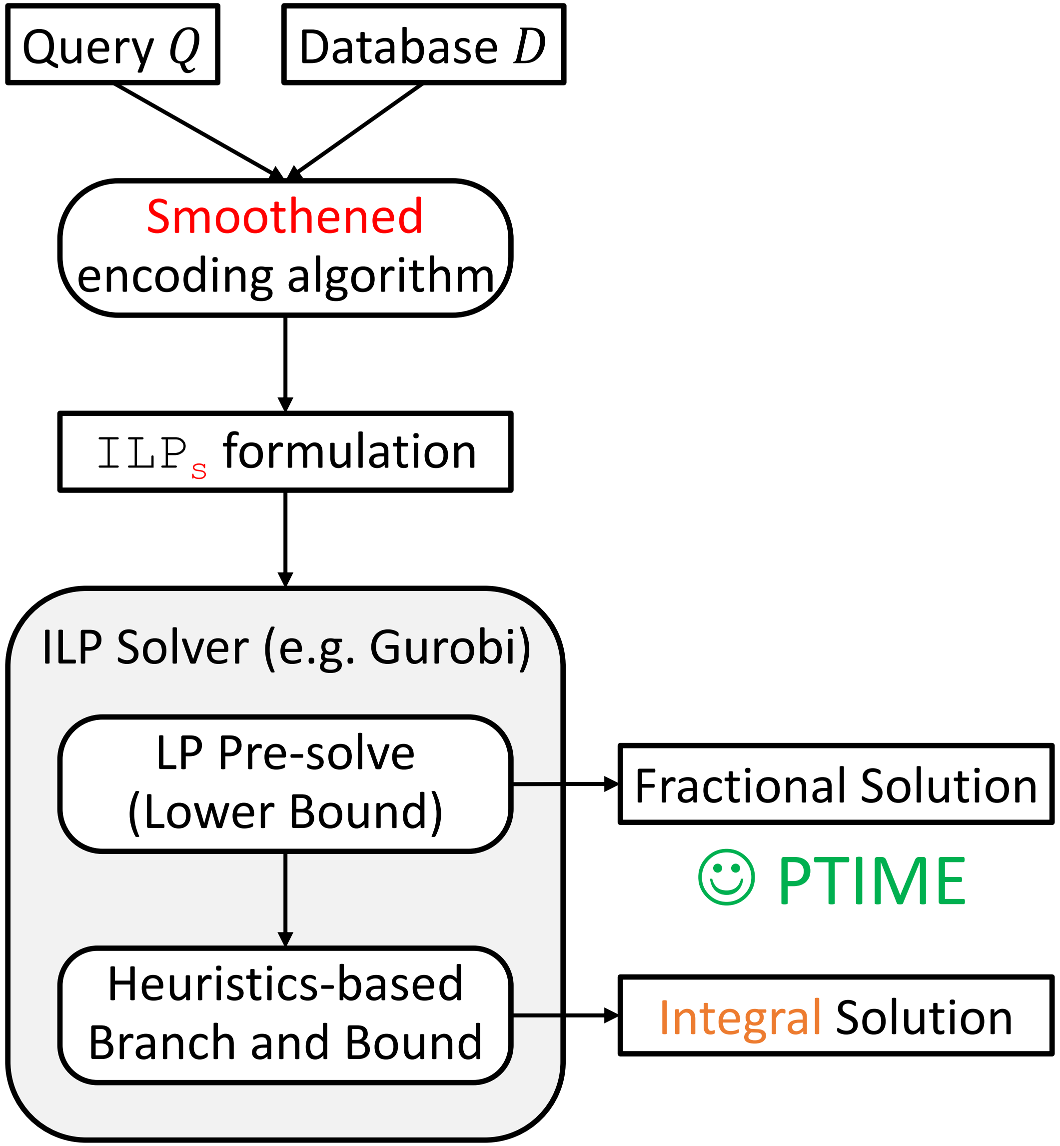
Unified Algorithms for Reverse Data Management

`ILP` formulation:

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View:





f^*=1
e.g. $\mathbf{x}^*$=(1,1)

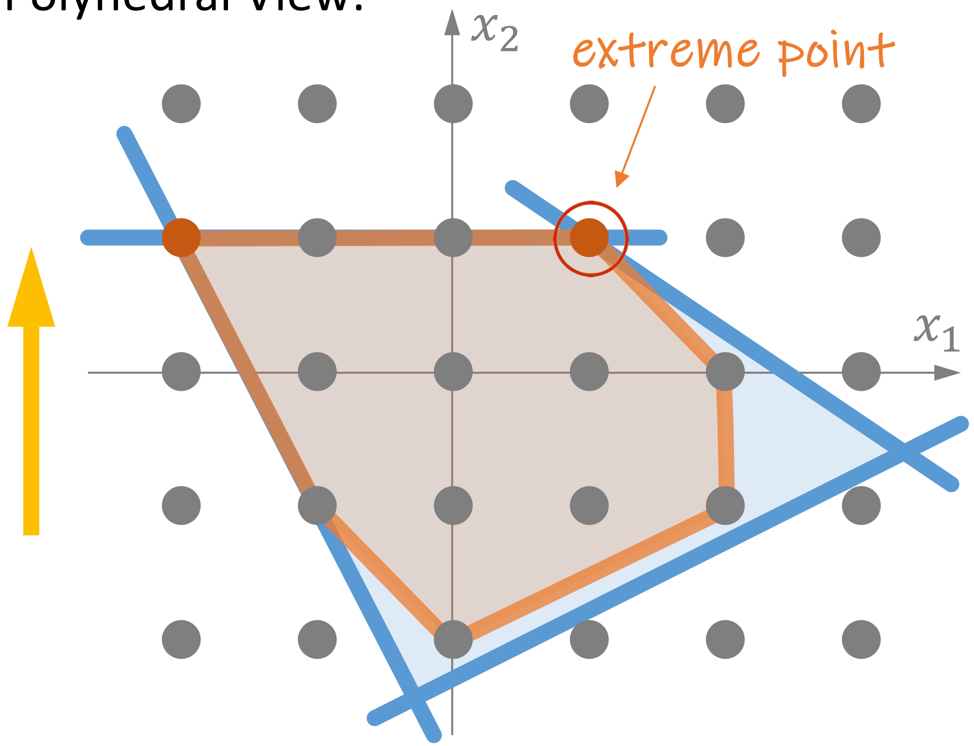
☺ PTIME

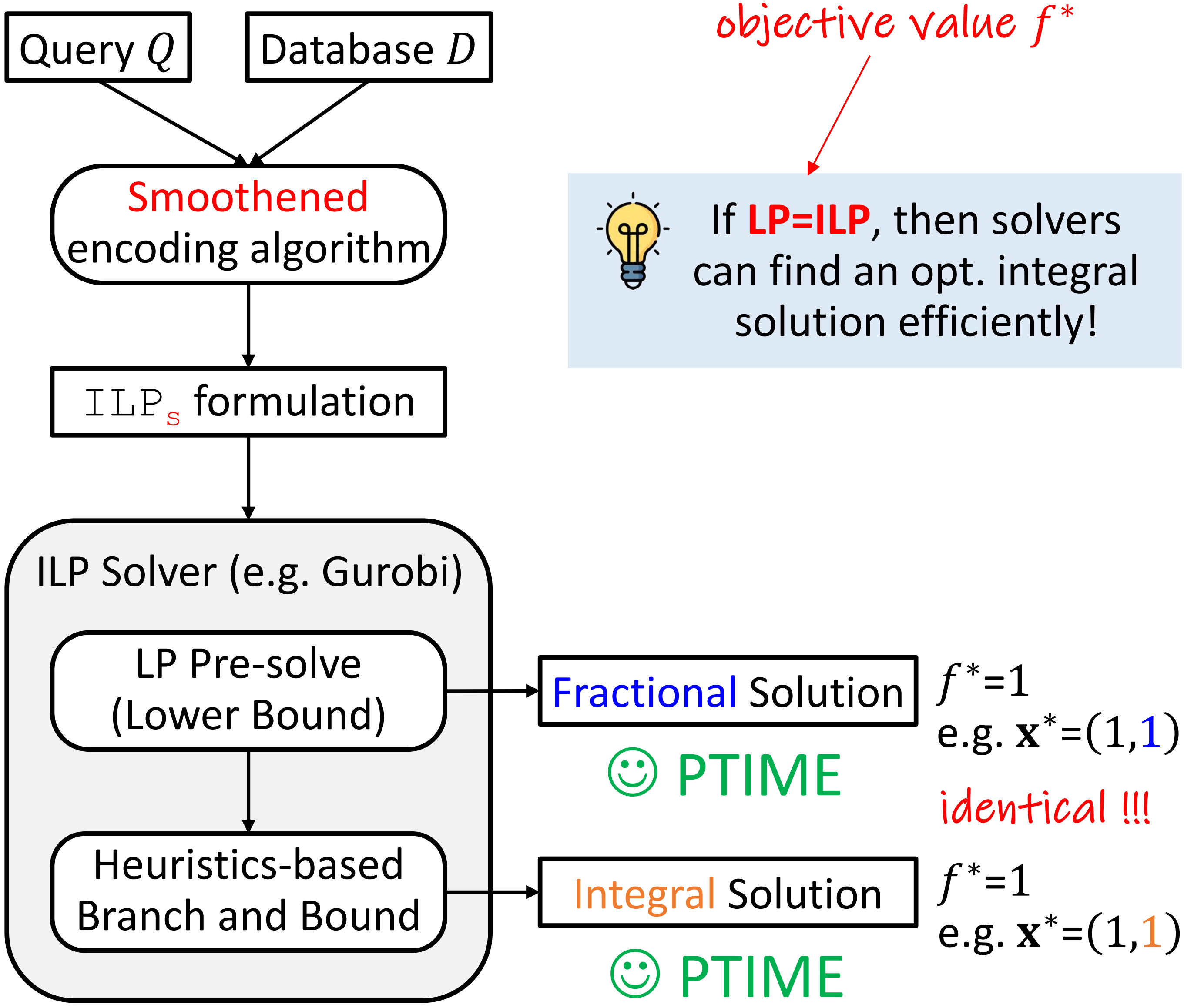
Unified Algorithms for Reverse Data Management

`ILP` formulation:

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View:



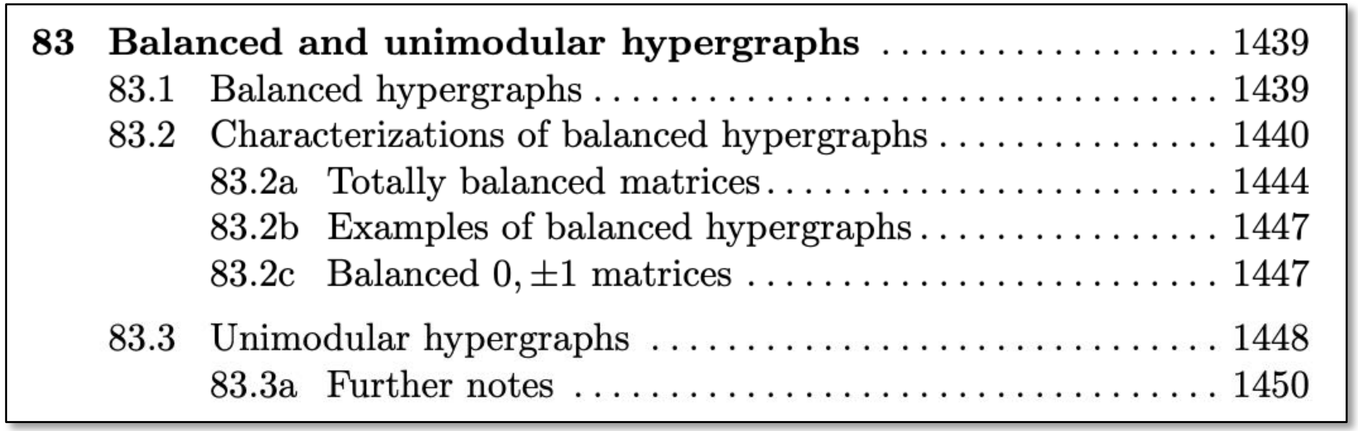


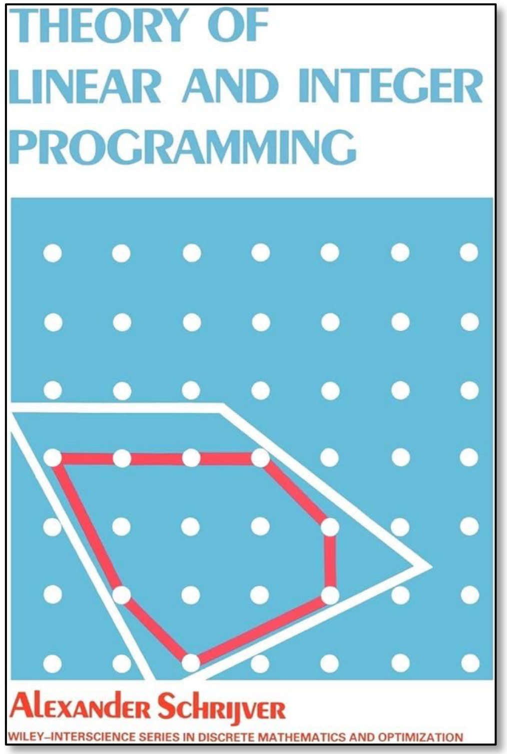
If **LP=ILP**, then solvers can find an opt. integral solution efficiently!

When is LP=ILP according to the literature?

`ILP` formulation:

$$
\begin{aligned}
f^* &= \min[\mathbf{c}\cdot\mathbf{x}] \\
\text{s.t. } &\mathbf{A}\cdot\mathbf{x} \geq \mathbf{b} \\
&\mathbf{x} \in \mathbb{N}^n/\mathbb{R}^n
\end{aligned}
$$

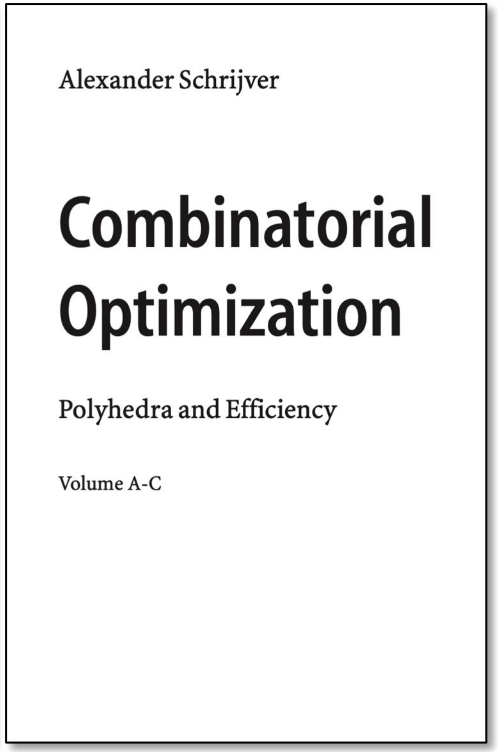




THEORY OF LINEAR AND INTEGER PROGRAMMING

ALEXANDER SCHRIJVER

WILEY–INTERSCIENCE SERIES IN DISCRETE MATHEMATICS AND OPTIMIZATION



Alexander Schrijver

Combinatorial Optimization

Polyhedra and Efficiency

Volume A-C

When is LP=ILP according to the literature: not useful ☹

`ILP` formulation:

objective vector

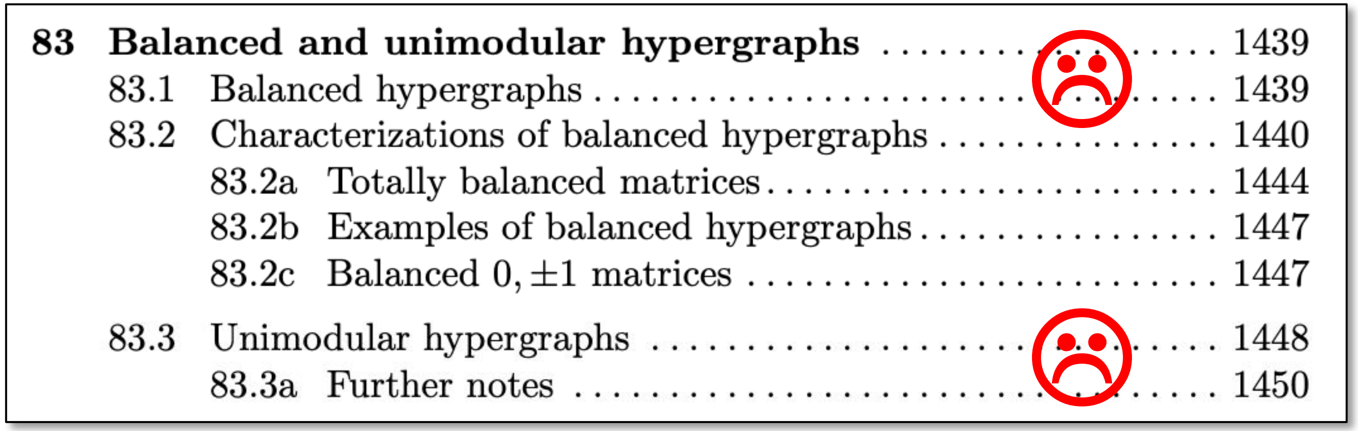
$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$

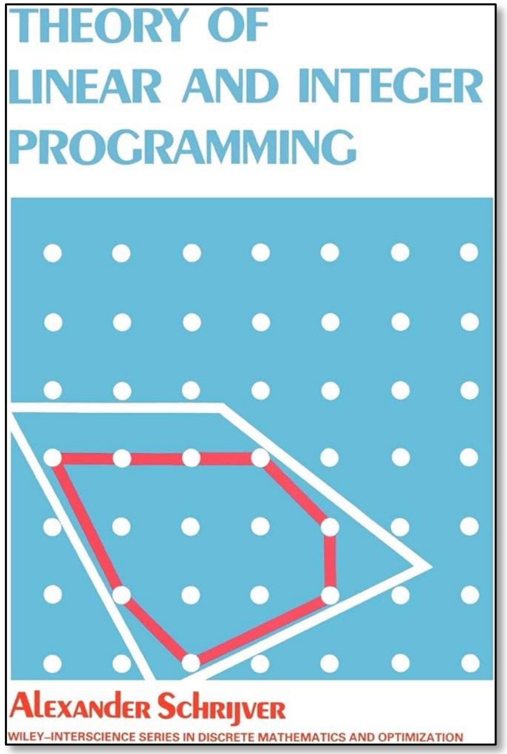
constraint vector

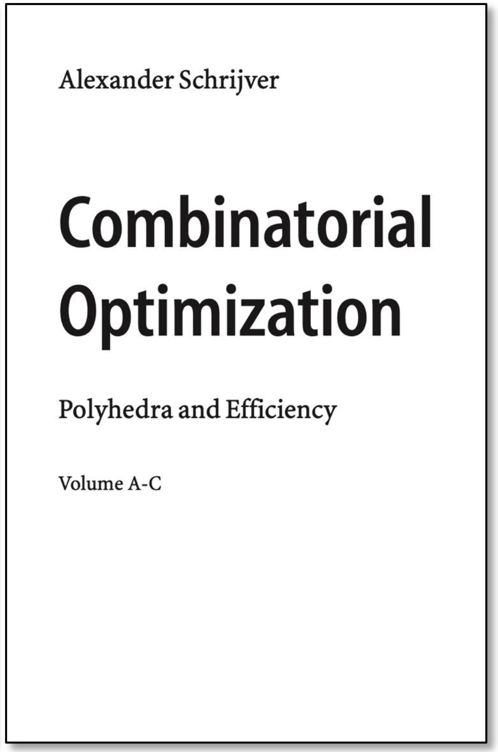
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$

constraint matrix

$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$







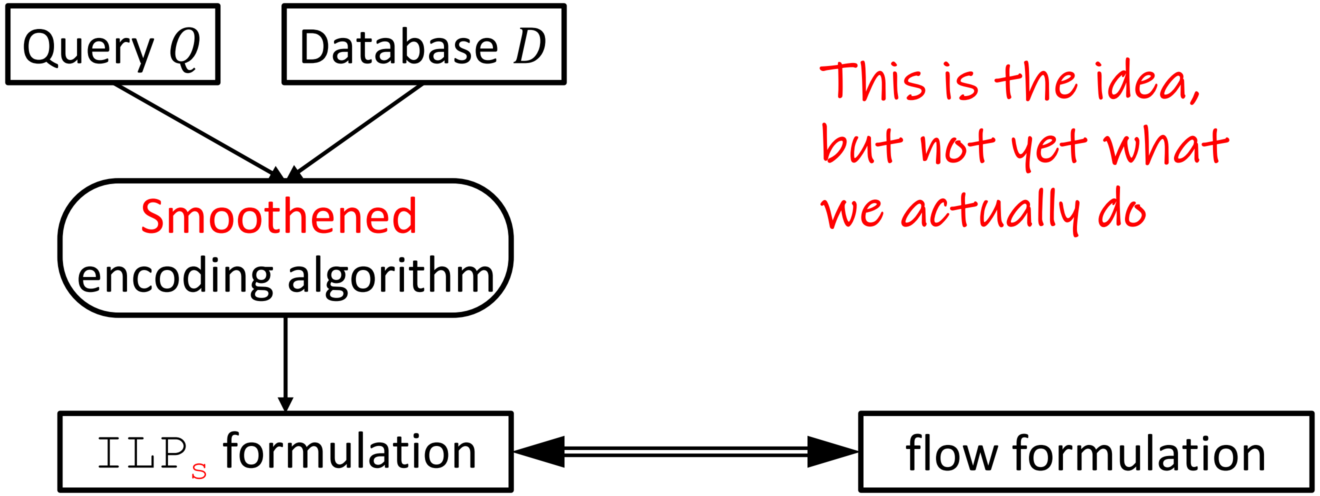
- Focus of polyhedral theory mainly on constraint matrix $\mathbf{A}$. But our PTIME constraint matrixes need not be balanced, nor Totally Unimodular, etc.

1. Our complexity results take into account the objective vector $\mathbf{c}$!
2. This gives us a separation between the problem under set vs. bag semantics!
3. We use an indirect proof via problem-specific MFMC encodings ☺

So what do we do to show ILP=LP for PTIME cases?

`ILP` formulation:

$$\begin{aligned} f^* &= \min[\mathbf{c}\cdot\mathbf{x}] \\ \text{s.t. } \mathbf{A}\cdot\mathbf{x} &\geq \mathbf{b} \\ \mathbf{x} &\in \mathbb{N}^n/\mathbb{R}^n \end{aligned}$$



Query Q

Database D

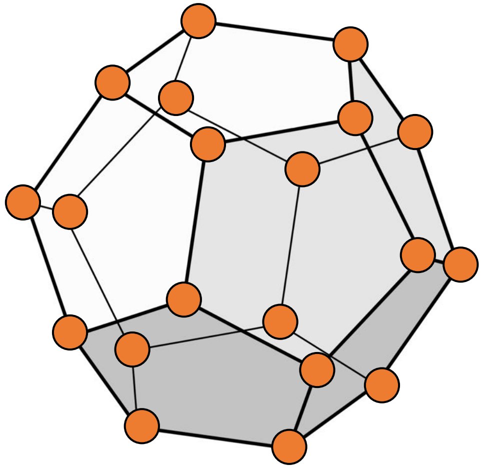
Smoothened encoding algorithm

$\texttt{ILP}_s$ formulation

flow formulation

This is the idea,
but not yet what
we actually do

Polyhedral View



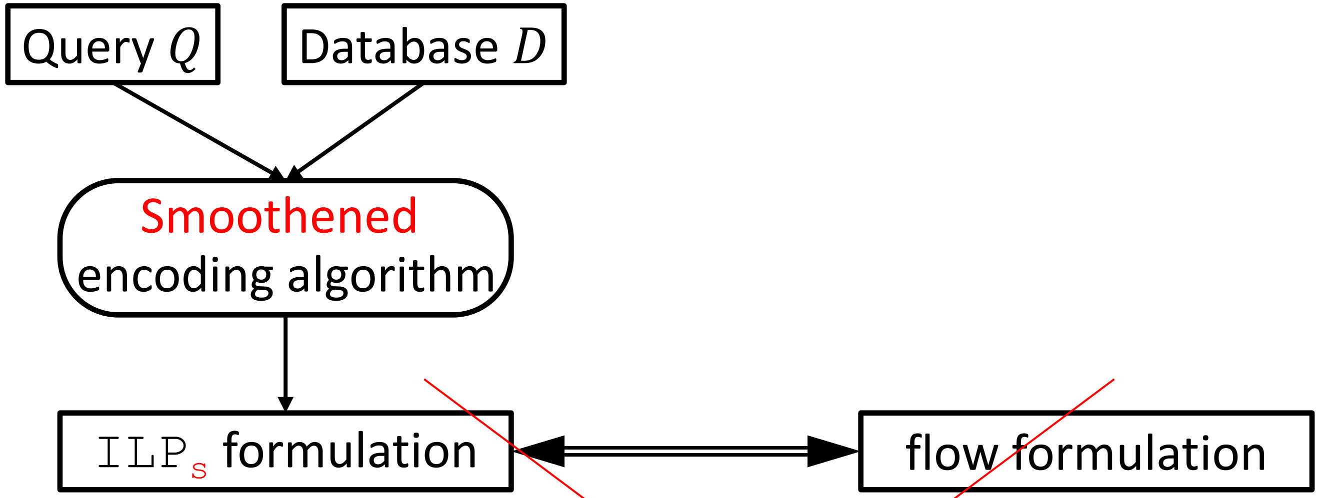
- For an "ideal" constraint matrix, the vertices (extreme points) of its polytope are all integral

1. Show correspondence to a flow formulation
2. Constraint matrix represented by a flow graph is an "ideal matrix"
3. COROLLARY: LP = ILP, PTIME ☺

So what do we do to show ILP=LP for PTIME cases?

`ILP` formulation:

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$
$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$



Query Q

Database D

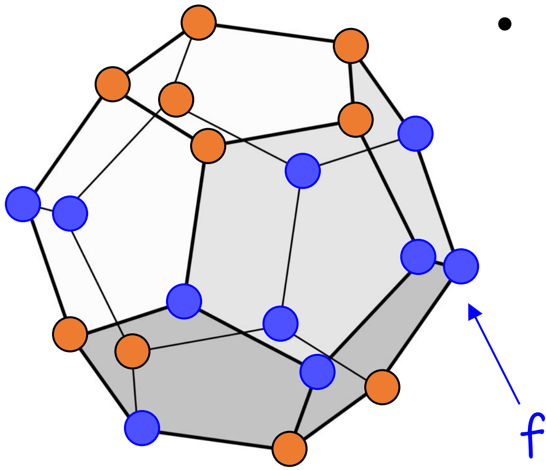
Smoothed encoding algorithm

$\texttt{ILP}_{\texttt{s}}$ formulation

flow formulation

1. ~~Show correspondence to a flow formulation~~
2. ~~Constraint matrix represented by a flow graph is an "ideal matrix"~~
3. ~~COROLLARY: LP = ILP, PTIME ☺~~

Polyhedral View



- For an "ideal" constraint matrix, the vertices (extreme points) of its polytope are all integral

But it is not as easy!

We have fractional extremal points ☹

So what do we do to show ILP=LP for PTIME cases?

`ILP` formulation:

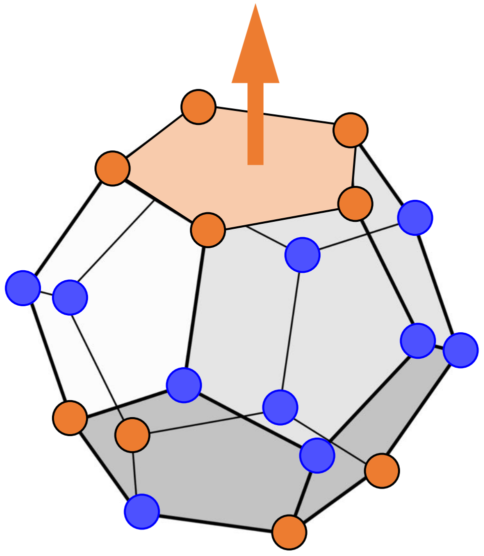
objective vector

$$f^* = \min[\mathbf{c} \cdot \mathbf{x}]$$

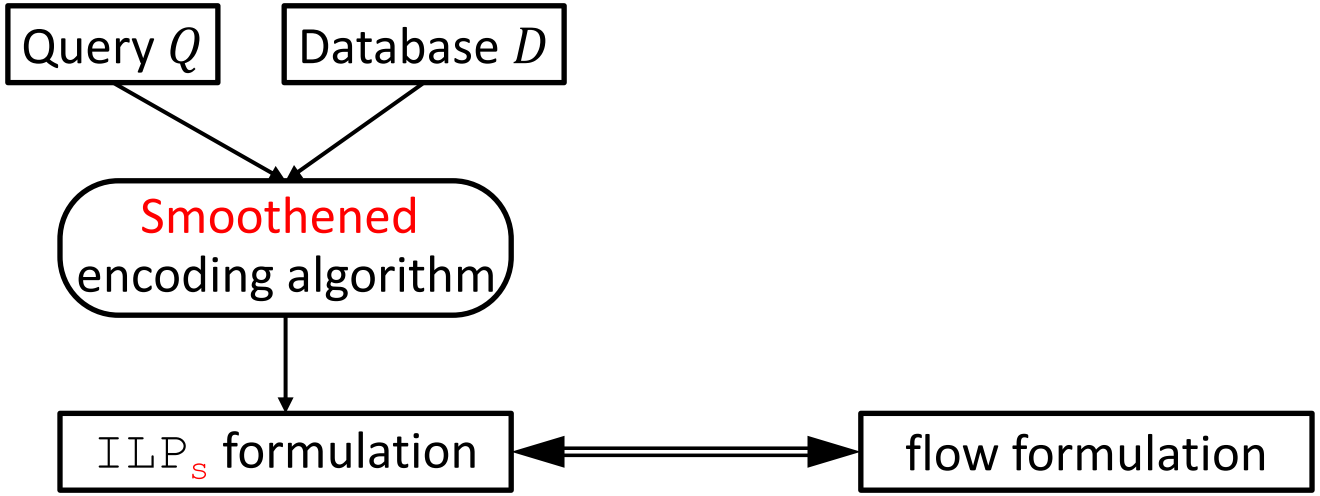
$$\text{s.t. } \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$$

$$\mathbf{x} \in \mathbb{N}^n / \mathbb{R}^n$$

Polyhedral View



- The optimal solution face contains only integral vertices
- Polytope may have non-integral vertices



Query Q

Database D

Smoothed encoding algorithm

`ILP`$_{\text{s}}$ formulation

flow formulation

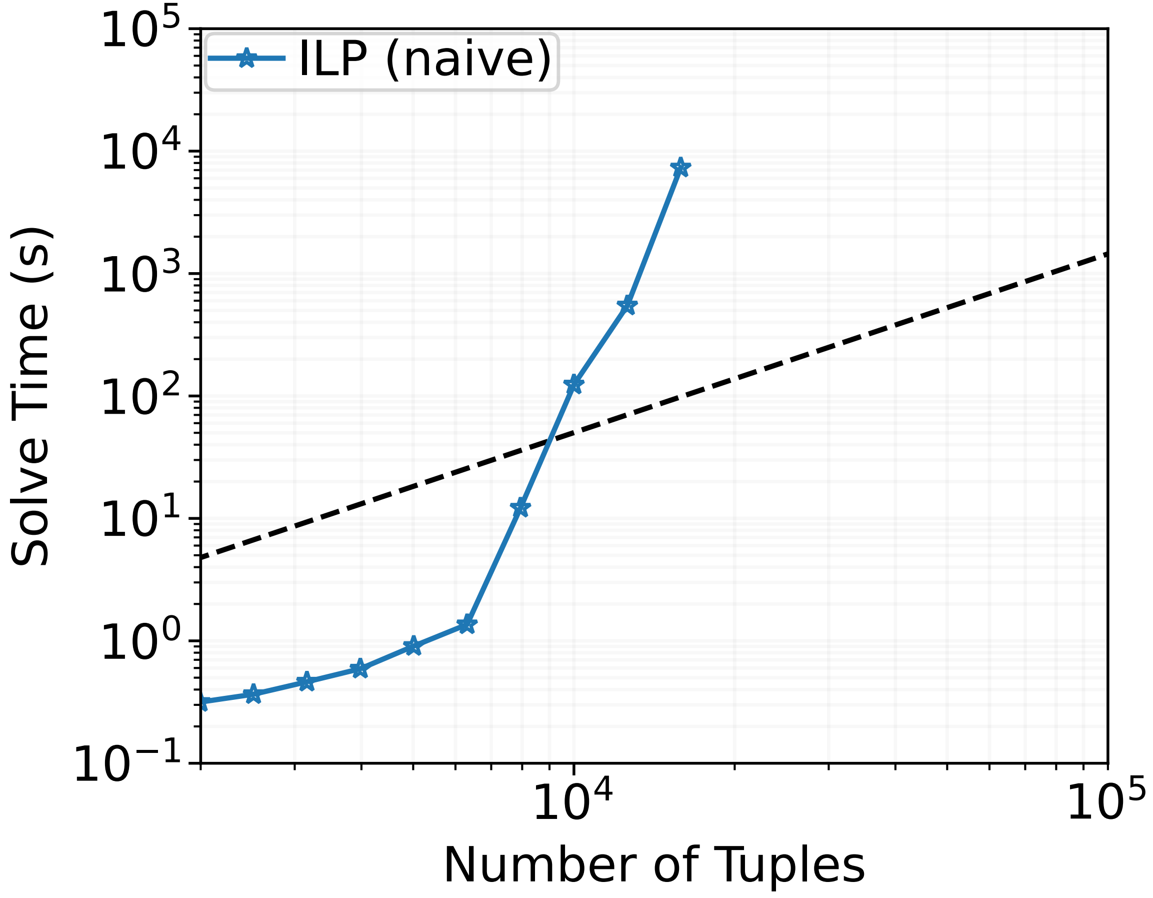
1. Show correspondence *after processing* that takes the objective vector into account
2. *After processing*, constraint matrix is ideal
3. COROLLARY: LP = ILP, PTIME ☺

- Showing such correspondences for all PTIME cases in all scenarios is non-trivial
- This moves the challenge from algorithm development to proofs!

Scalability of Naive vs. Smoothened ILP for PTIME query

Finding an optimal solution for the Smallest Witness Problem ([SIGMOD'19], [ICDT'24])

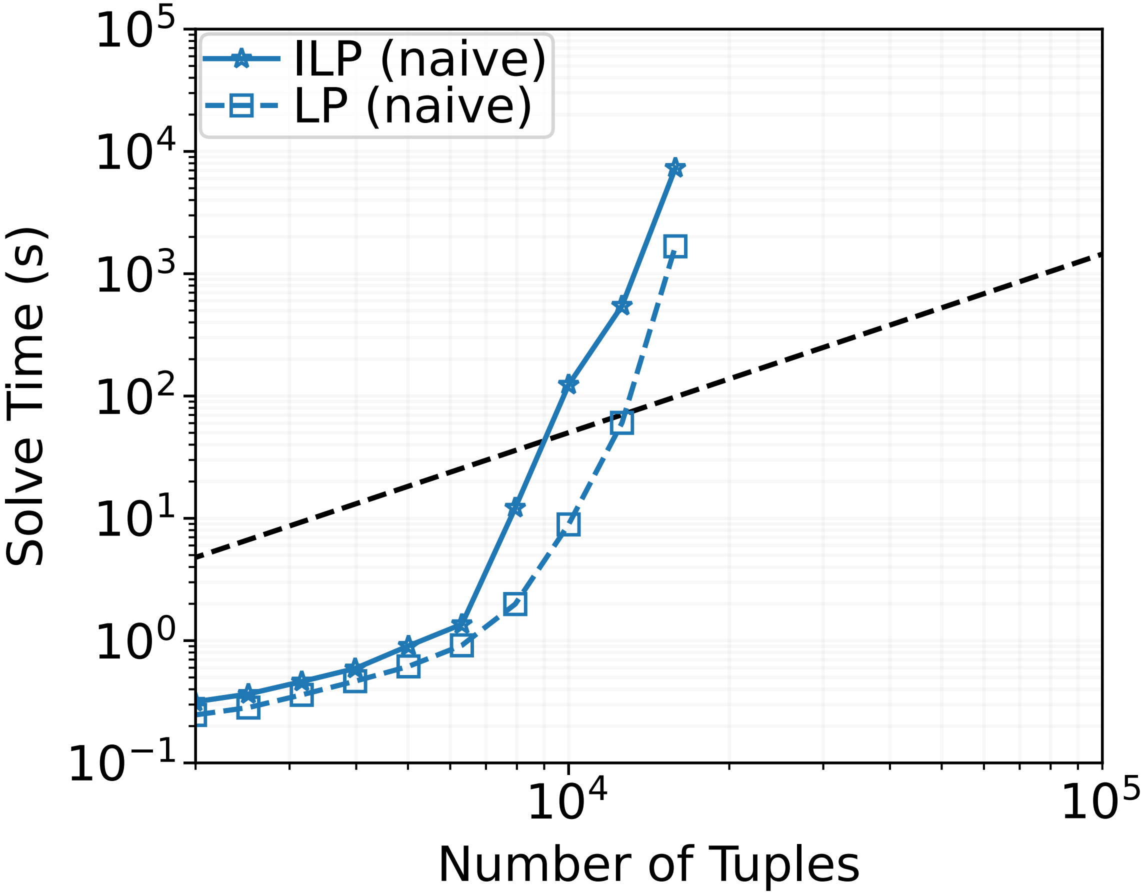
$Q_{5\text{star}}(x){:}-R(x,a),S(x,b),T(x,c),U(x,d),V(x,e),A(x)$



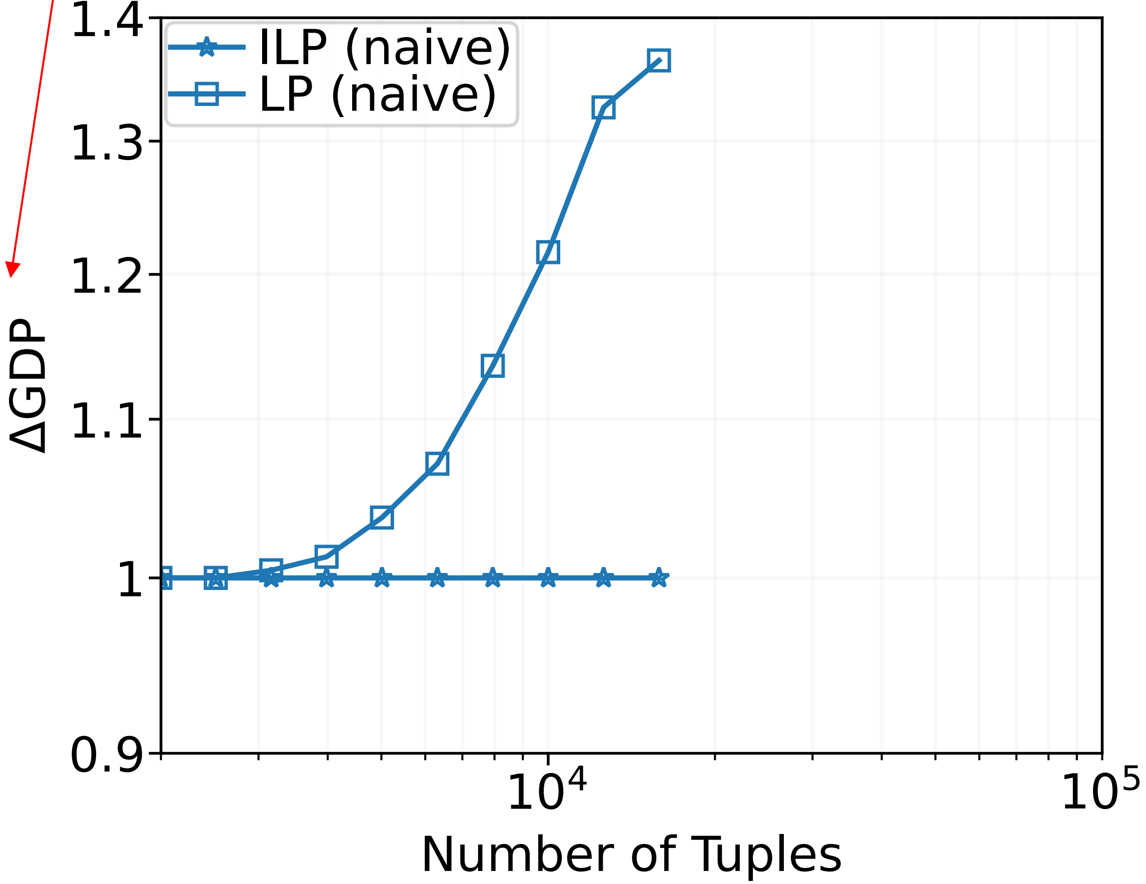
Scalability of Naive vs. Smoothened ILP for PTIME query

Finding an optimal solution for the Smallest Witness Problem ([SIGMOD'19], [ICDT'24])

$Q_{5\text{star}}(x){:}-R(x,y),S(x,z),T(x,u),U(x,v),V(x,w),W(x)$



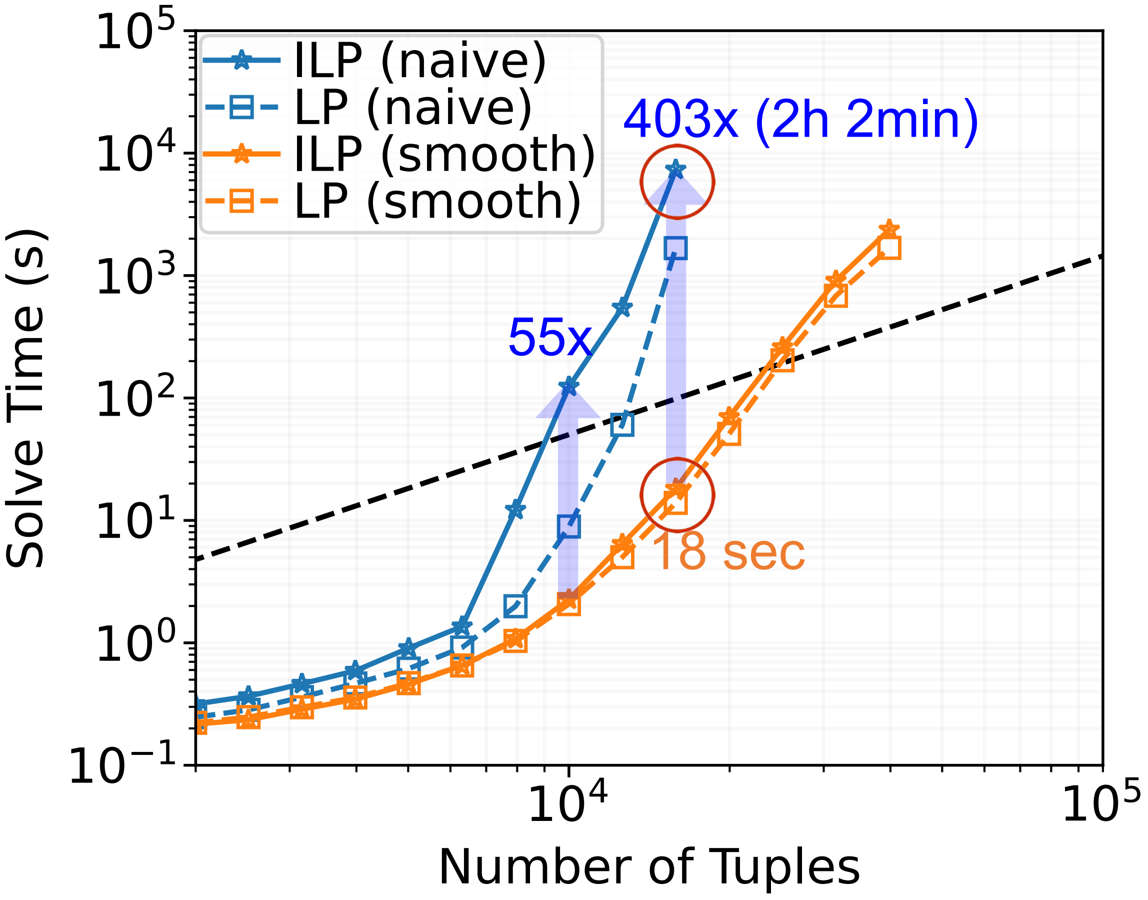
How different are the optimal values for the ILP formulation and its LP relaxation



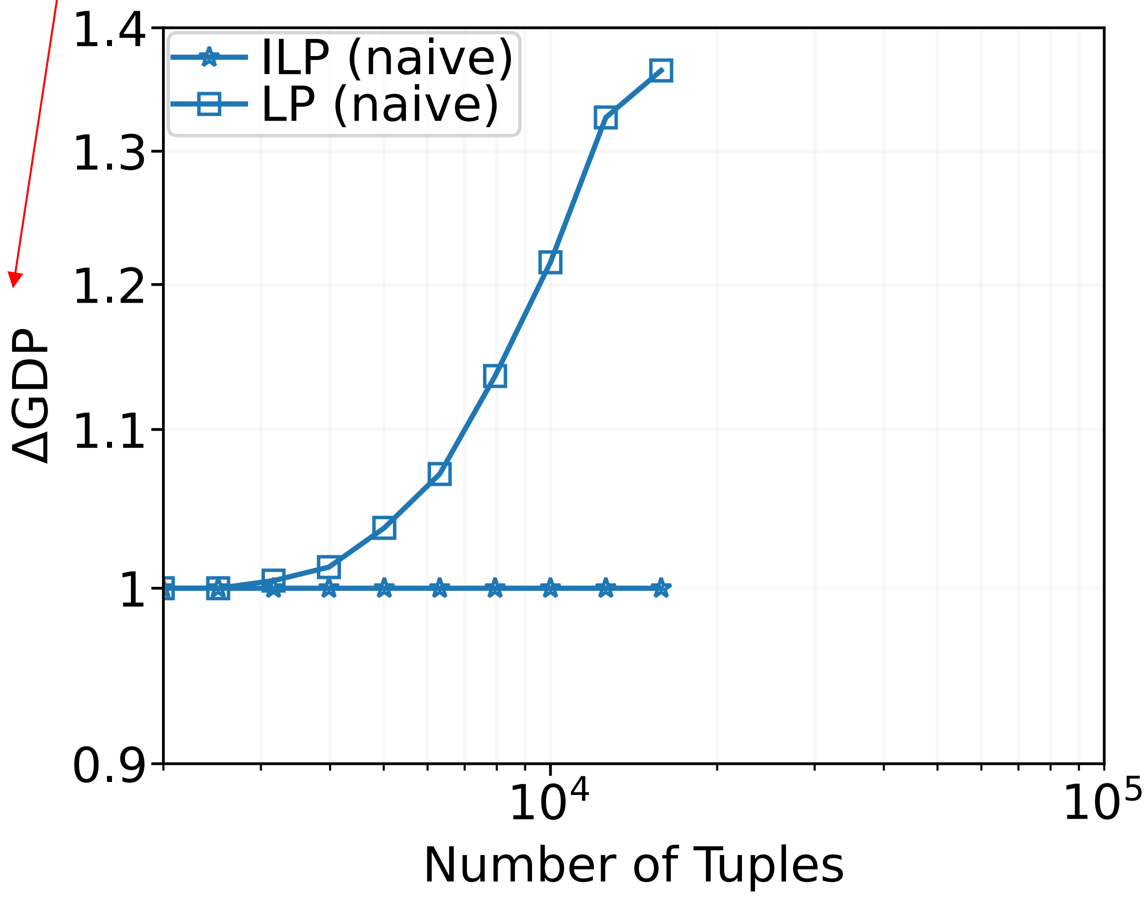
Scalability of Naive vs. Smoothened ILP for PTIME query

Finding an optimal solution for the Smallest Witness Problem ([SIGMOD'19], [ICDT'24])

$Q_{5\text{star}}(x){:}-R(x,a),S(x,b),T(x,c),U(x,d),V(x,e),A(x)$



How different are the optimal values for
the ILP formulation and its LP relaxation

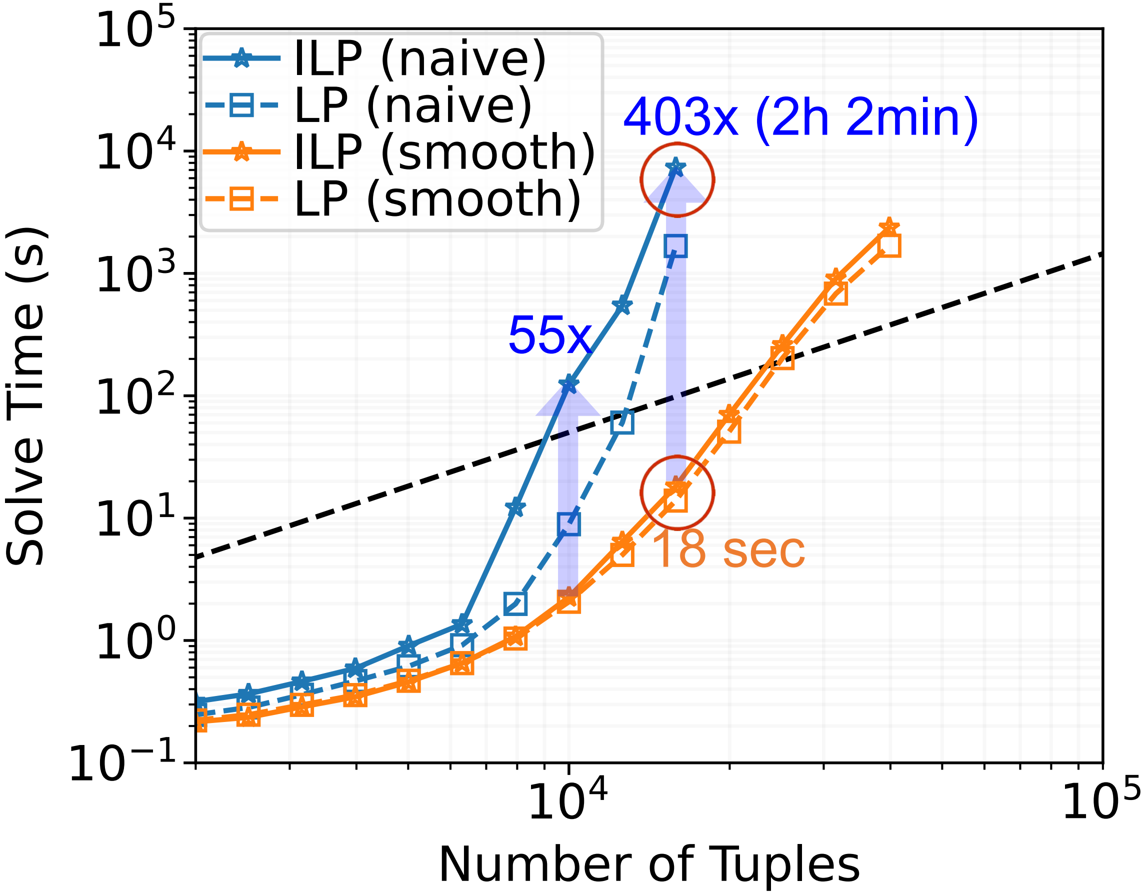


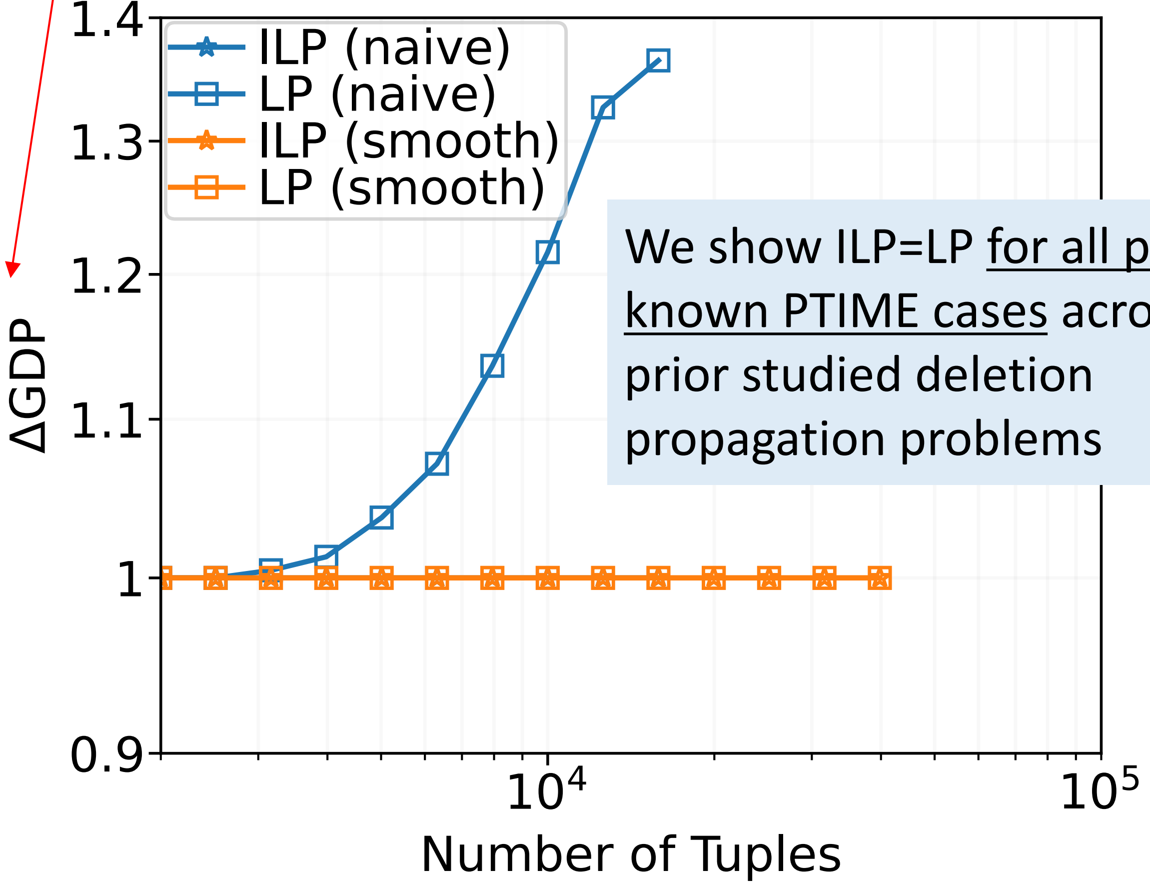
Scalability of Naive vs. Smoothened ILP for PTIME query

Finding an optimal solution for the Smallest Witness Problem ([SIGMOD'19], [ICDT'24])

$Q_{5\text{star}}(x){:}-R(x,a),S(x,b),T(x,c),U(x,d),V(x,e),A(x)$

How different are the optimal values for the ILP formulation and its LP relaxation



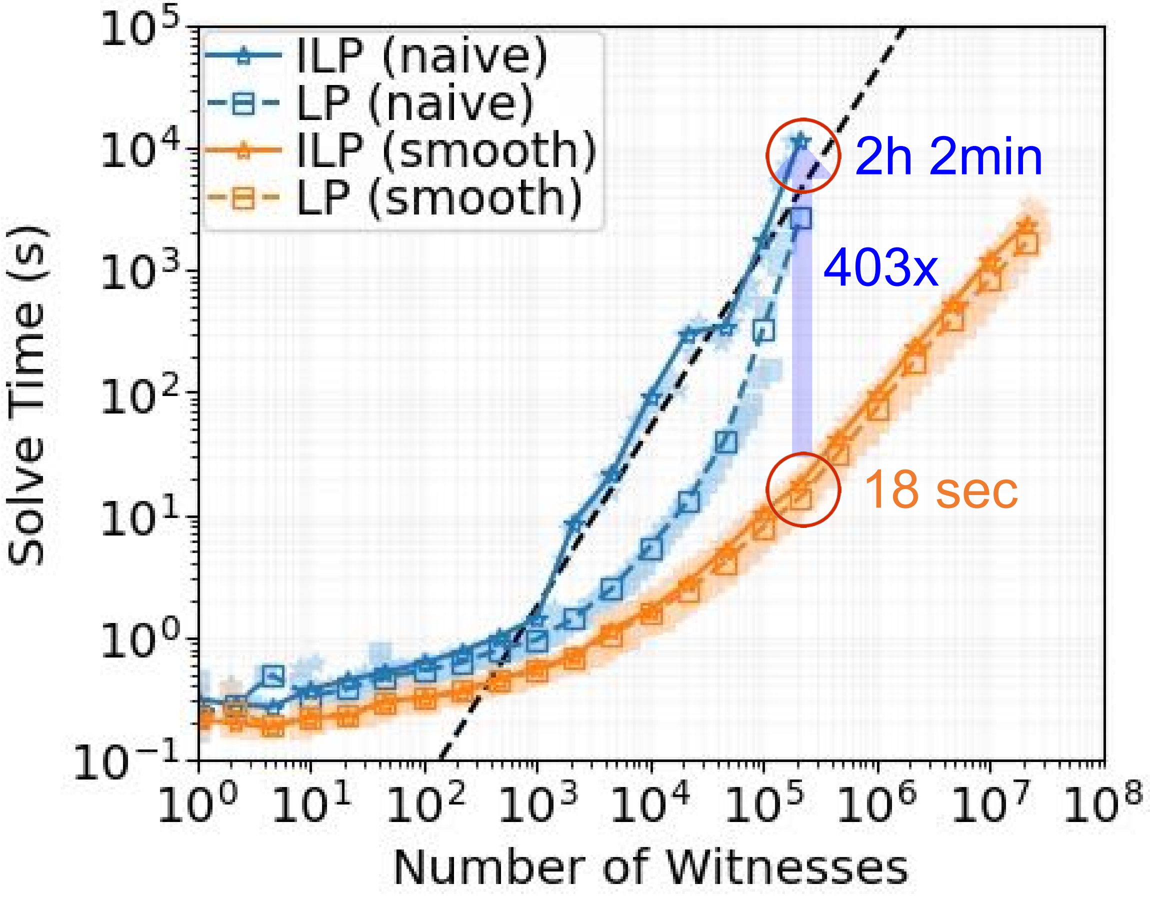


We show ILP=LP for all prior known PTIME cases across prior studied deletion propagation problems

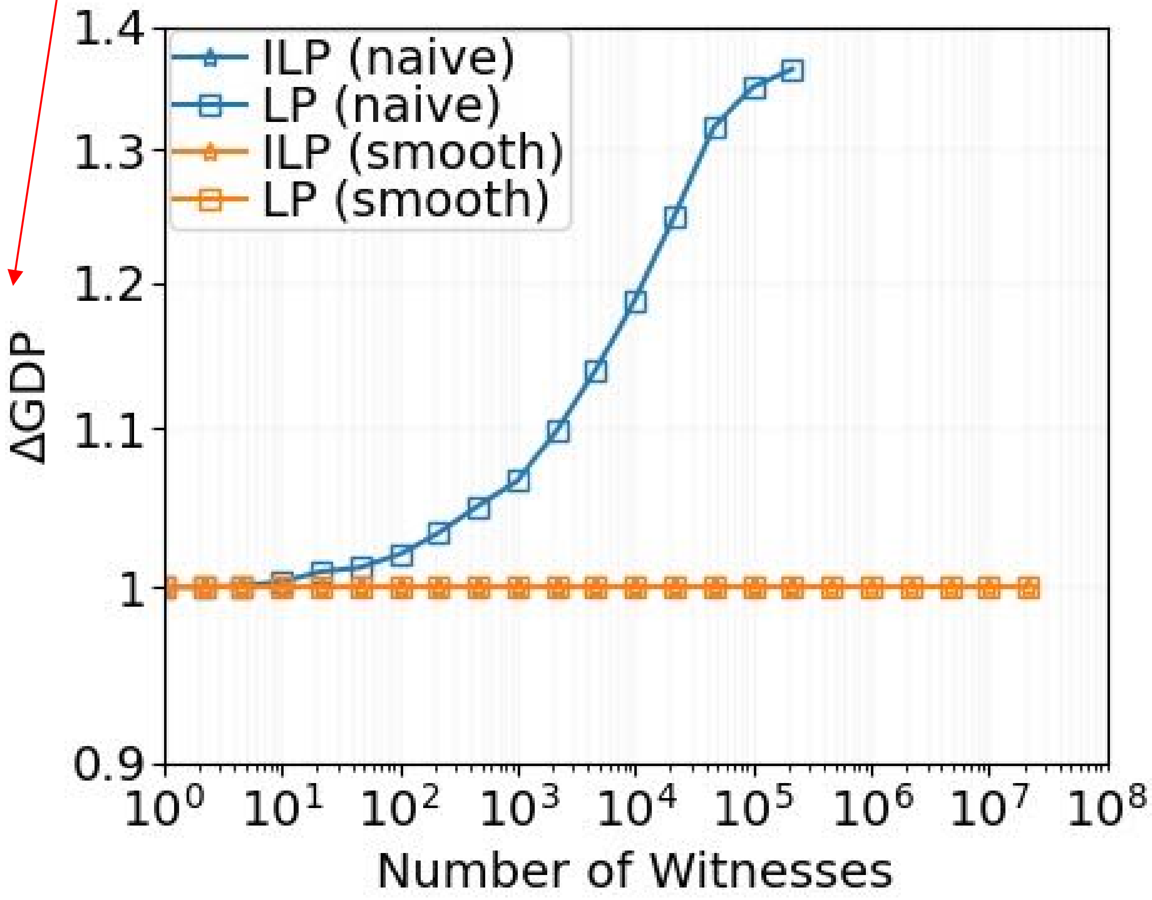
Scalability of Naive vs. Smoothened ILP for PTIME query

Finding an optimal solution for the Smallest Witness Problem ([SIGMOD'19], [ICDT'24])

$Q_{5\text{star}}(x){:}-R(x,a),S(x,b),T(x,c),U(x,d),V(x,e),A(x)$



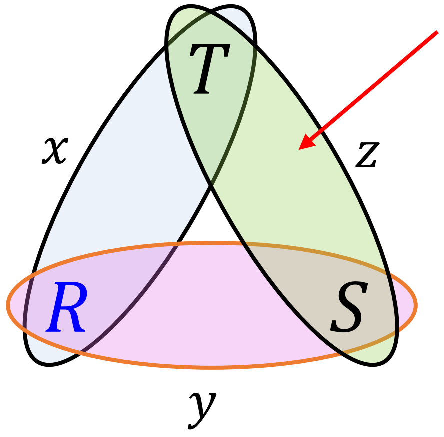
How different are the optimal values for the ILP formulation and its LP relaxation



Example complicated landscape for resilience

Triangle query

$Q^{\Delta}:-R(x,y),S(y,z),T(x,z)$



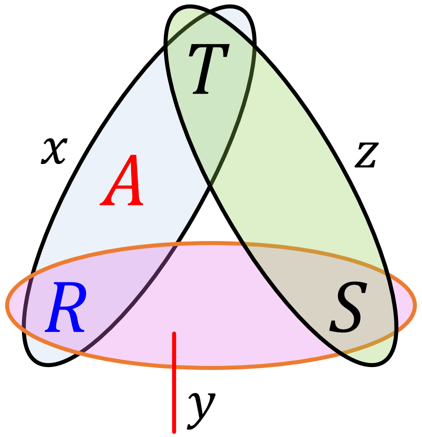
Dual Hypergraph

```
select exists(
   select 1
   from R, S, T
   where R.y=S.y
   and S.z=T.z
   and T.x=R.x)
```

NPC

Triangle unary

$Q_{A}^{\Delta}:-R(x,y),S(y,z),T(x,z),A(x)$



```
select exists(
   select 1
   from R, S, T, A
   where R.y=S.y
   and S.z=T.z
   and T.x=R.x
   and A.x=R.x)
```

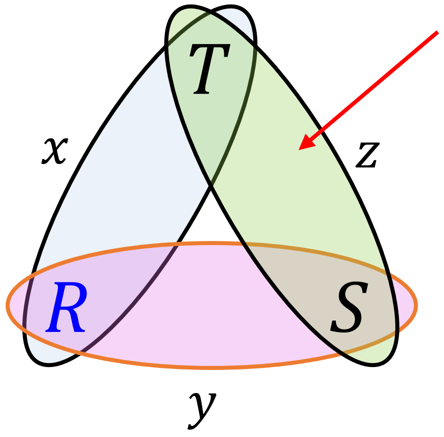
PTIME

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

Example complicated landscape for resilience

Triangle query

$Q^{\Delta}\colon -R(x,y), S(y,z), T(x,z)$



Dual Hypergraph

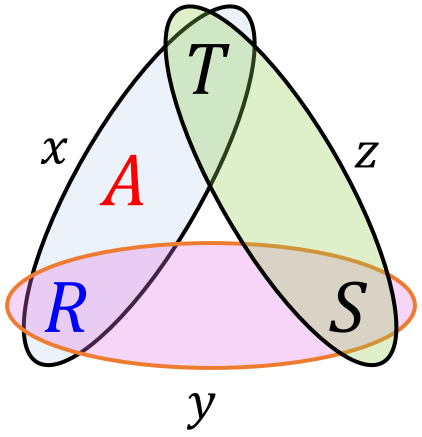
```
select exists(
  select 1
  from R, S, T
  where R.y=S.y
  and S.z=T.z
  and T.x=R.x)
```

NPC

PTIME for FD $x \rightarrow y$

Triangle unary

$Q_A^{\Delta}\colon -R(x,y), S(y,z), T(x,z), A(x)$



```
select exists(
  select 1
  from R, S, T, A
  where R.y=S.y
  and S.z=T.z
  and T.x=R.x
  and A.x=R.x)
```

PTIME

NPC under bag semantics

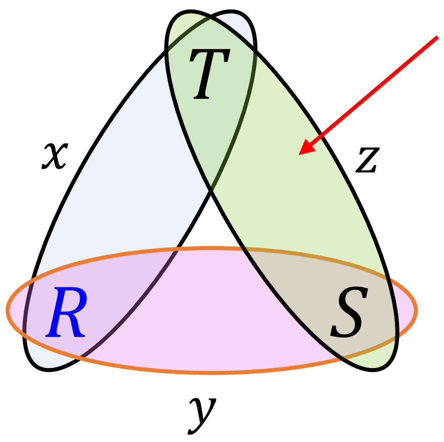
PTIME if provenance happens to be read-once

Freire, Gatterbauer, Immerman, Meliou. The complexity of resilience and responsibility for self-join-free conjunctive queries, VLDB 2015. https://doi.org/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://doi.org/10.1145/3626715

Example complicated landscape for resilience

Triangle query

$Q^{\Delta}{:}-R(x,y),S(y,z),T(x,z)$



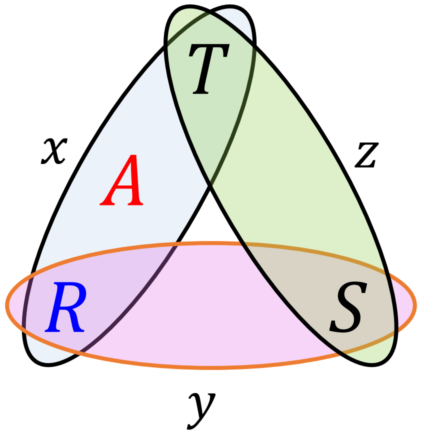
```
select exists(
  select 1
  from R, S, T
  where R.y=S.y
  and S.z=T.z
  and T.x=R.x)
```

NPC

PTIME for FD $x \rightarrow y$

Triangle unary

$Q^{\Delta}_{A}{:}-R(x,y),S(y,z),T(x,z),A(x)$



```
select exists(
  select 1
  from R, S, T, A
  where R.y=S.y
  and S.z=T.z
  and T.x=R.x
  and A.x=R.x)
```

PTIME

NPC under bag semantics

PTIME if provenance happens to be read-once

"Coarse-grained instance-optimal" algorithm

Freire, Gatterbauer, Immerman, Meliou. The complexity of resilience and responsibility for self-join-free conjunctive queries, VLDB 2015. https://doi.org/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://doi.org/10.1145/3626715
Makhija, Gatterbauer: Is ILP all you need for Deletion Propagation? PVLDB 2025.

Outline

1. Reverse Data Management (RDM)
2. A magical ILP formulation
3. Take-aways

(An illustrated example: only if time remains)

A possible shift of focus of algorithm design in database theory?

Current focus: discrete algorithms

Identify tractable cases for a class of problems that can be solved with some dedicated discrete algorithm (like dynamic programming or greedy) or a reduction to flow

For the hard cases:

- prove hardness via some dedicated reduction from some NPC problem.
- optionally design a separate dedicated approximation algorithm

Partial solutions: Often, the algorithm (the dichotomy) does not extend to all types of queries like CQs with self-joins, or problems under bag semantics

Future: polyhedral algorithms

Design one "appropriate" ILP program to solve all problems

- "appropriate" here means that their natural LP relaxation has the same optimal objective for all PTIME cases ("LP=ILP"), which proves the ILP can be solved in PTIME.

All cases are covered (including the hard ones) ✓

- also approximation algorithms, just stop evaluation early, anytime algorithm comes for free ✓

Complete solutions: All problem types are covered ✓ (including self-joins or bag semantics)

A possible shift of focus of algorithm design in database theory?

Current focus: discrete algorithms

Identify tractable cases for a class of problems that can be solved with some dedicated discrete algorithm (like dynamic programming or greedy) or a reduction to flow

For the hard cases:

- prove hardness via some dedicated reduction from some NPC problem.
- optionally design a separate dedicated approximation algorithm

Partial solutions: Often, the algorithm (the dichotomy) does not extend to all types of queries like CQs with self-joins, or problems under bag semantics

Future: polyhedral algorithms

Design one "appropriate" ILP program to solve all problems

- "appropriate" here means that their natural LP relaxation has the same optimal objective for all PTIME cases ("LP=ILP"), which proves the ILP can be solved in PTIME.

All cases are covered (including the hard ones) ✓

- also approximation algorithms, just stop evaluation early, anytime algorithm comes for free ✓

Complete solutions: All problem types are covered ✓ (including self-joins or bag semantics)

An anonymous concern: "...the ILP constructed is not a simple mathematical object... Since the construction given is .... not a simple mathematical object, it is not clear to me how deep one can push this further by analyzing it."

A possible shift of focus of algorithm design in database theory?

	Current focus: discrete algorithms	**Future: polyhedral algorithms**
Theory	Identify tractable cases for a class of problems that can be solved with some dedicated discrete algorithm (like dynamic programming or greedy) or a reduction to flow	Design one "appropriate" ILP program to solve all problems • "appropriate" here means that their natural LP relaxation has the same optimal objective for all PTIME cases ("LP=ILP"), which proves the ILP can be solved in PTIME.
	For the hard cases: • prove hardness via some dedicated reduction from some NPC problem. • optionally design a separate dedicated approximation algorithm	All cases are covered (including the hard ones) ✓ • also approximation algorithms, just stop evaluation early, anytime algorithm comes for free ✓
	Partial solutions: Often, the algorithm (the dichotomy) does not extend to all types of queries like CQs with self-joins, or problems under bag semantics	Complete solutions: All problem types are covered ✓ (including self-joins or bag semantics)
	Practical aspects: usually only some problem cases are solved, hard cases often not treated, the practical nature of the algorithms is not always clear	**Practical aspects**: it works from day one ✓

Take-aways: "Is ILP all you need? ..."

- Polyhedral theory solves many database theory problems "out of the box".
  - Shift in focus: instead of trying to find a dedicated PTIME algorithm for PTIME cases, start with a general formulation and prove it finishes in PTIME for PTIME cases.
  - There is some magic in getting the "right" formulation (ILP = LP for PTIME), we don't yet have "the" recipe
  - The proofs for LP=ILP go beyond standard optimization literature. Polyhedral theory alone does not help.
- The overall philosophy is way more general than reverse data management.
  - Makhija, Gatterbauer. *Minimally Factorizing the Provenance of Self-Join Free Conjunctive Queries*, PODS 2024. https://doi.org/10.1145/3651605
  - What about consistent query answering? And even more general logic optimization problems?
- More concretely open: Unifying deletion and insertion propagation ("change propagation"), basically positive and negative provenance / Why or Why not?
  - Meliou, Gatterbauer, Moore, Suciu. *Why so? or Why no? Functional causality for explaining query answers*. MUD 2010. https://arxiv.org/pdf/0912.5340
- Please also talk to Neha ☺
  Faculty at UMass Amherst from Fall'25

Thank you ☺