

# Quantitative Reasoning about Constraint Violations

Benny Kimelfeld

Technion Data & Knowledge Lab

[tdk.cs.technion.ac.il](http://tdk.cs.technion.ac.il)

EDBT-INTENDED Summer School 2022

JULY 4-9, 2022, Bordeaux, France

# Outline

1. Introduction & Background
2. Inconsistency Measures
3. Complexity of Calculation
4. Probabilistic Database Viewpoint
5. Responsibility Attribution
6. Concluding Remarks

# Examples of Inconsistency (DBPedia)



Marion Jones

dbo:height

- 1.524
- 1.778



Cullen Douglas

dbo:birthPlace

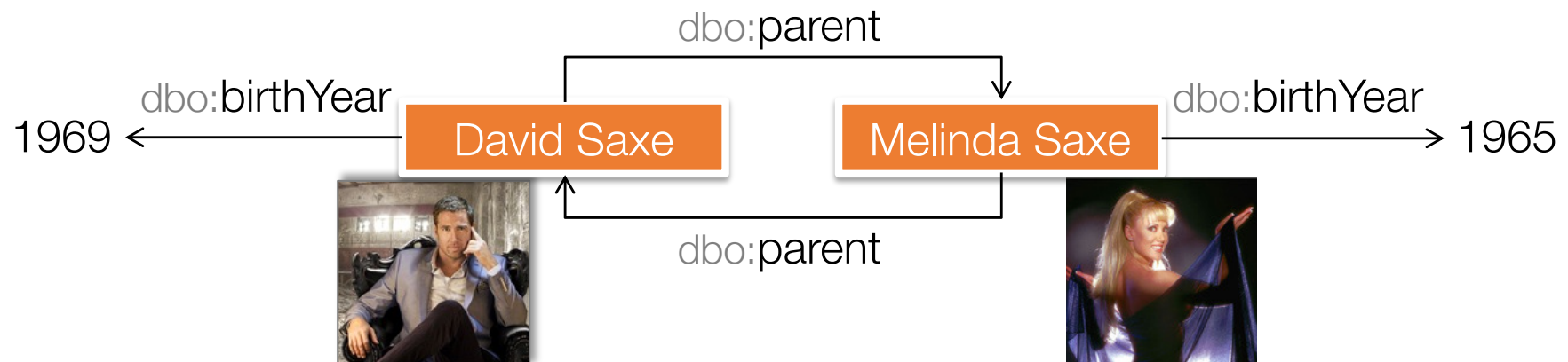
- dbr:California
- dbr:Florida



Irene Tedrow

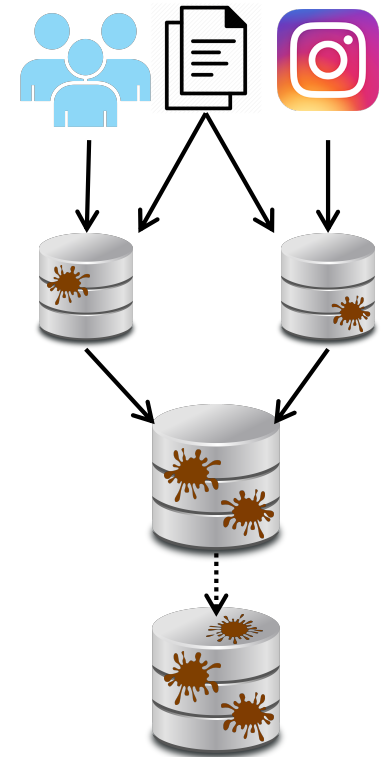
dbo:deathPlace

- dbr:California
- dbr:Hollywood,\_Los\_Angeles
- dbr:New\_York\_City



# Inconsistency

- “Inconsistent data”: integrity constraints violated
- Why so?
  - Imprecise **data sources**
    - Crowd, Web pages, social encyclopedias, sensors, ...
  - Imprecise **data generation**
    - Natural-language processing, sensor/signal processing, image recognition, ...
  - Conflicts in **data integration**
    - Crowd + enterprise data + KB + Web + ...
  - Data **staleness**
    - Entities change address, status, ...
  - *And so on ...*





# Measuring Inconsistency

---

To what extent are constraints being **violated**?

*Who studies it?*

- **KR** research: measuring the inconsistency of a KB (set of logical statements)
- **DB** research: constraint mining, data cleaning, probabilistic databases
- **AI/SRL** research: Markov Logic Networks, Probabilistic Soft Logic, ...

# Inconsistency Measures to Soften Logic

---

*Inconsistency measures have been around, explicitly or implicitly, playing different roles*

Approximate constraints



Low level of inconsistency under some measure of choice

Markov Logic



Start with a space of possible worlds  $D$  :

$$\Pr(D) \sim F(\text{inconsistency}(D))$$

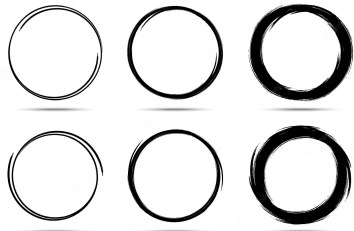
Prob. DB via soft rules



Start with an initial database  $D$ , make random changes to build  $D'$  :

$$\Pr(D') \sim F_1(\text{intervention}(D' | D)) \cdot F_2(\text{inconsistency}(D'))$$

# Some Usage of Inconsistency Measures



Notions of **soft**  
(weak/approx)  
constraints

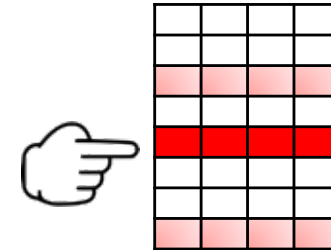
[Kivinen-Mannila-95]  
[Sen-Deshpande-Getoor-09]  
[Chu-Ilyas-Papotti-13]  
[Rekatsinas-Chu-Ilyas-Ré-17]  
[Kruse-Naumann-18]  
[Rammelaere-Geerts-18]

...



**Progress** indication  
for data repairing  
processes

[Livshits-Kochirgan-Tsur-  
Ilyas-K-Roy-21]



Attribution of  
**responsibility** to  
inconsistency

[Hunter-Konieczny-10]  
[Yun-Vesic-Croitoru-  
Bisquert-18]  
[Deutch-Frost-Gilad-  
Sheffer-20]  
[Livshits-K-21]

...

# Plan for this Lecture

---

- Discuss inconsistency measurement from the viewpoint of the usages
- Reference past research projects with collaborators
- Focus on **algorithms and complexity** analysis for relevant tasks



Leopoldo Bertossi



Ihab Ilyas



Ester Livshits



Sudeepa Roy

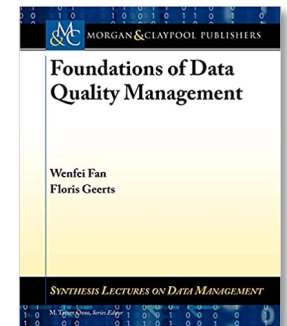
Main collaborators

# Types of Integrity Constraints

- Key constraints
  - `Person(ssn,name,birthCity,birthState)`
- Functional Dependencies (FDs)
  - `birthCity → birthState`
  - Generally, `X → Y` where `X` and `Y` are sets of attributes
- Conditional FDs
  - `zip → city` whenever `country="France"`
- Denial constraints
  - `not[ Parent(x,y) & Parent(y,x) ]` (forbidden patterns)
- Referential (foreign-key) constraints
  - `Parent(x,y) → Person(x) & Person(y)`

*Anti-monotonic constraints:  
consistency preserved by subsets*

Cf. [Fan-Geerts-12] for a comprehensive study of constraints in data quality management



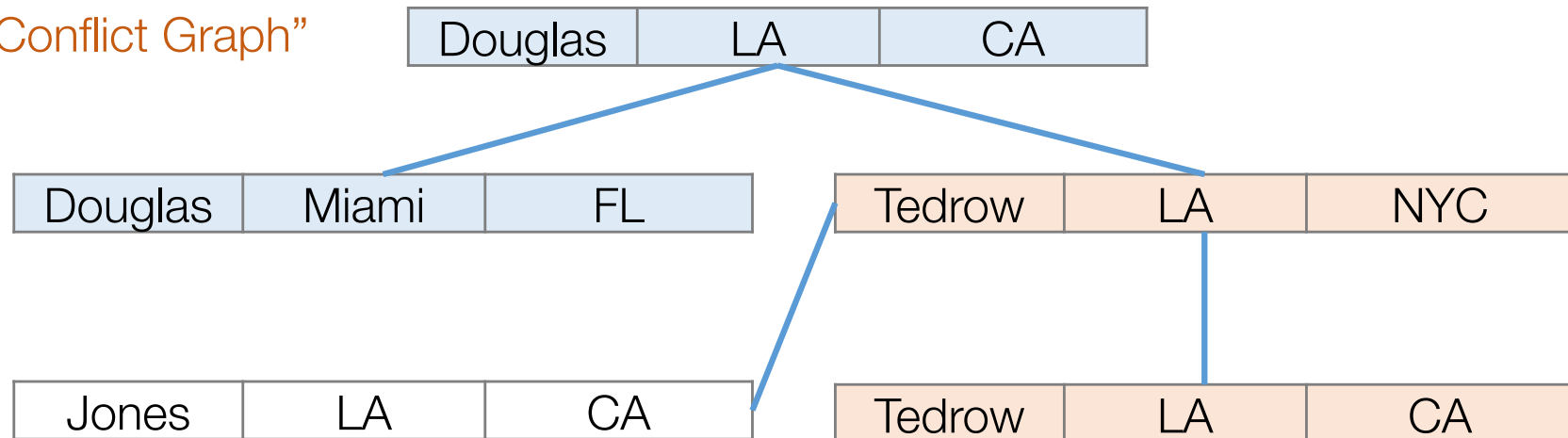
# Example of Functional Dependencies

$\text{person} \rightarrow \text{birthCity}$

$\text{birthCity} \rightarrow \text{birthState}$

person	birthCity	birthState
Douglas	LA	CA
Douglas	Miami	FL
Tedrow	LA	CA
Tedrow	LA	NYC
Jones	LA	CA

“Conflict Graph”



# Repairs

---

- **Inconsistent database** violates constraints
  - Representation:  $(\Sigma, D)$  where the database  $D$  violates the set  $\Sigma$  of constraints
- **Repair**: a consistent variant via a *legitimate fix*
  - **Subset repairs**: set-max consistent subset
  - **Cardinality repairs**: cardinality-max consistent subset
  - More: *update* repairs (value updates), *symmetric-difference* repairs (tuple insertion/deletion), ...
  - [Arenas-Bertossi-Chomicki-99]

# Example: Subset/Cardinality Repairs

person  $\rightarrow$  birthCity

birthCity  $\rightarrow$  birthState

person	birthCity	birthState
Douglas	LA	CA
Douglas	Miami	FL
Tedrow	LA	CA
Tedrow	LA	NYC
Jones	LA	CA

person	birthCity	birthState
Douglas	LA	CA
Douglas	Miami	FL
Tedrow	LA	CA
Tedrow	LA	NYC
Jones	LA	CA

(Subset) repair

person	birthCity	birthState
Douglas	LA	CA
Douglas	Miami	FL
Tedrow	LA	CA
Tedrow	LA	NYC
Jones	LA	CA

Cardinality repair



# Classic Repair Problems

---

- Repair checking
  - Given  $D$  and  $D'$ , is  $D'$  a repair of  $D$ ?
  - [Chomicki-Marcinkowski-05] [Afrati-Kolaitis-09]
- Consistent Query Answering (CQA)
  - *Which query answers hold albeit inconsistency?* Tuples in  $Q(D')$  for all repairs  $D'$  [Arenas+99] [Koutris-Wijesen-17]
- Repairing / Cleaning
  - Compute a (good/best) repair
  - [Bertossi+08] [Kolahi-Lakshmanan-09] [Livshits-K-Roy-18]
- Repair counting
  - For databases [Maslowski-Wijesen-14] [Livshits+21] [Calautti+22]
  - For knowledge bases [DeBona-Grant-18] [Hunter-Konieczn-18]

# Outline

1. Introduction & Background

▶ 2. Inconsistency Measures **we are here**

3. Complexity of Calculation

4. Probabilistic Database Viewpoint

5. Responsibility Attribution

6. Concluding Remarks

# Definition: Inconsistency Measure

---

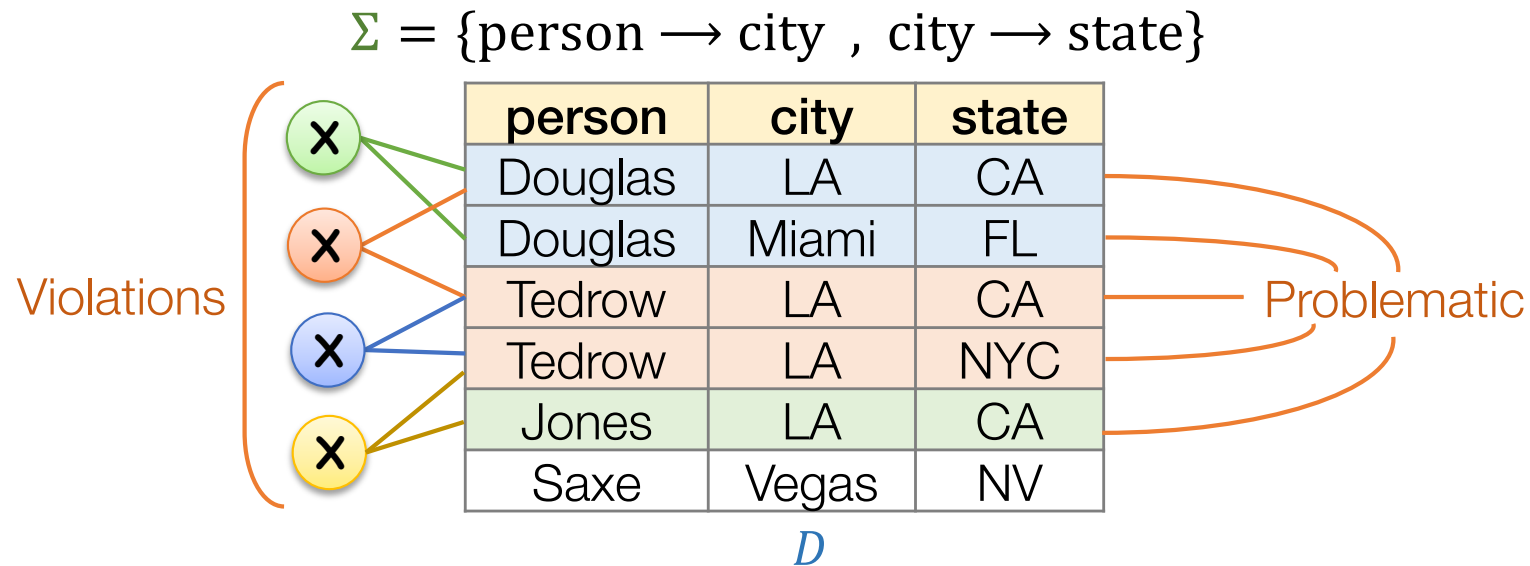
- Notation:
  - $\Sigma$  denotes a set of integrity constraints
  - $D$  denotes a database
- An *inconsistency measure* is a function  $I$  that maps each pair  $(\Sigma, D)$  to a non-negative number, such that  $I(\Sigma, D) > 0$  iff  $D$  violates  $\Sigma$ 
  - Intuitively,  $I(\Sigma, D) > I(\Sigma', D')$  means that  $D$  is farther from satisfying  $\Sigma$  than  $D'$  from satisfying  $\Sigma'$
- We focus on **anti-monotonic**  $\Sigma$  (e.g., FDs, DCs)

# Basic Inconsistency Measures

---

- **Drastic:** 1 or 0 (inconsistent or consistent)
  - [Thimm-17] (*Later: makes sense in responsibility attribution*)
- **#violations** (i.e., set-min inconsistent subsets)
  - [Kivinen-Mannila-95] [Hunter-Konieczny-08] (“MI Shapley Inconsistency”)
- **#problematic** tuples (i.e., tuples in violations)
  - [Kivinen-Mannila-95] [Grant-Hunter-11]
- **#repairs:** number of maximal consistent subsets
  - [Grant-Hunter-11]
- **repair\_cost:** minimal #tuples to delete to attain consistency (**cardinality repair**)
  - [Huhtala+98] [Kivinen-Mannila-95] [Grant-Hunter-13] [Bertossi-18] [Rammelaere-Geerts-18] (constraint “confidence”)

# Example 1



$$\text{drastic}(\Sigma, D) = 1$$

$$\#\text{violations}(\Sigma, D) = 4$$

$$\#\text{problematic}(\Sigma, D) = 5$$

$$\#\text{repairs}(\Sigma, D) = 3$$

$$\text{repair\_cost}(\Sigma, D) = 2$$

## Repairs

person	city	state
Douglas	LA	CA
Tedrow	LA	CA
Jones	LA	CA
Saxe	Vegas	NV

person	city	state
Douglas	Miami	FL
Tedrow	LA	CA
Jones	LA	CA
Saxe	Vegas	NV

person	city	state
Douglas	Miami	FL
Tedrow	LA	NYC
Saxe	Vegas	NV

# Example 2

$$\Sigma = \{\text{person} \rightarrow \text{city}, \text{city} \rightarrow \text{state}\}$$

person	city	state
$p_1$	LA	CA
$p_2$	LA	CA
$\vdots$	$\vdots$	$\vdots$
$p_{50}$	LA	CA
Douglas	LA	NY

50

$$\text{drastic}(\Sigma, D_1) = 1$$

$$\# \text{violations}(\Sigma, D_1) = 50$$

$$\# \text{problematic}(\Sigma, D_1) = 51$$

$$\# \text{repairs}(\Sigma, D_1) = 2$$

$$\text{repair\_cost}(\Sigma, D_1) = 1$$

50

person	city	state
$p_1$	LA	CA
$p_2$	Miami	FL
$\vdots$	$\vdots$	$\vdots$
$p_{50}$	Utica	NY
Douglas	LA	NY

$$\text{drastic}(\Sigma, D_2) = 1$$

$$\# \text{violations}(\Sigma, D_2) = 1$$

$$\# \text{problematic}(\Sigma, D_2) = 2$$

$$\# \text{repairs}(\Sigma, D_2) = 2$$

$$\text{repair\_cost}(\Sigma, D_2) = 1$$

# Some Concepts of Soft Constraints

---

- Mining approximate constraints
  - [Kivinen-Mannila-95]: **#violations**, **#problematic**, **repair\_cost** ;  
[Huhtala+98]: **repair\_cost** ; DCFiner [Pena-Almeida-Naumann-19]:  
**#violations** ; [Livshits-Heidari-Ilyas-K-20] *abstraction*
- Markov Logic Networks [Richardson-Domingos-06]
  - Factor for every violation/satisfaction (weighted **#violations**)
  - Symmetric – every possible tuple is a variable
  - Instances: DeepDive [DeSa+16], Pr. Datalog+/- [Gottlob+11]
  - Similar concept: PrDB [Sen-Deshpande-Getoor-09]
- Soft-key constraints [Jha-Rastogi-Suciu-08]
  - Factor for every key violation of a specified size
  - Probabilistic graphical model – similar to MLN (factor for each violation/satisfaction)
- Approximate multivalued dependencies (MVD)
  - Conditional entropy as a measure of satisfaction [Kenig-Suciu-20]

# Postulates for Inconsistency Measures

---

- Goodness properties (postulates) of inconsistency measures studied by the KR community
  - [\[Hunter-Konieczny-08\]](#) [\[Grant-Hunter-11\]](#) [\[Thimm-17\]](#) [\[Grant-Parisi-19\]](#) ...
  - Different focus from databases
    - KB = set of logical statements
    - Postulates mainly talk about how changes in the KB affect the measure
- We studied properties desired for *progress indication* in data repairing



# The Basic Measures for Repairing Progress

## Monotonicity

Stricter constraints can only increase inconsistency

Studied in KR  
(w./ differences)  
[Grant-Parisi-19]

## Continuity

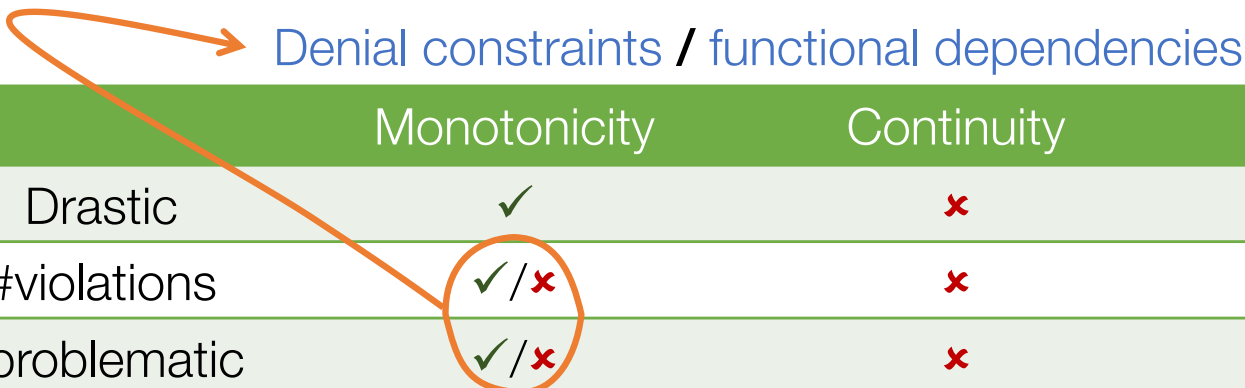
A single operation has a limited impact on inconsistency

“acceptable  
pacing”  
[Luo-Naughton-Ellmann-Watzke-04]

## Progression

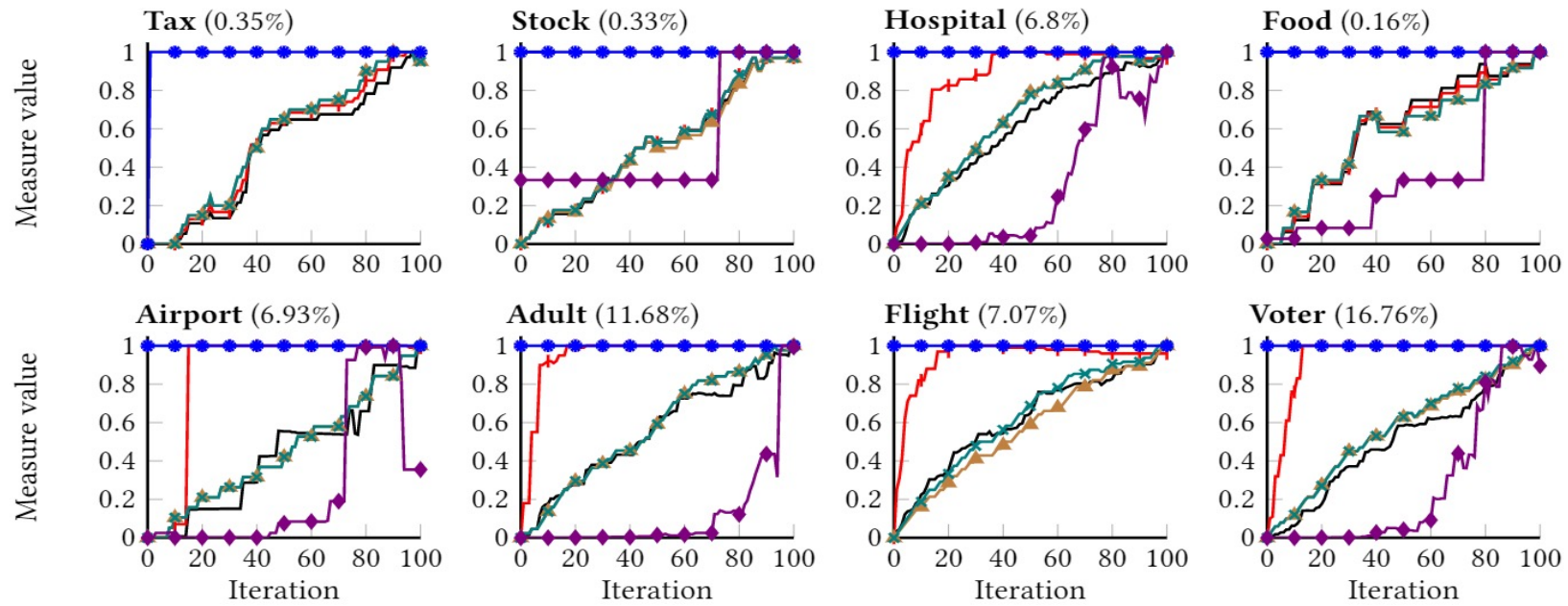
We can always find an op that reduces inconsistency

“continuously  
revised estimates”



	Monotonicity	Continuity	Progression
Drastic	✓	✗	✗
#violations	✓/✗	✗	✓
#problematic	✓/✗	✗	✓
#repairs	✗	✓	✗
repair_cost	✓	✓	✓

# Experiments



$I_d$  (—\*),  $I_{ML}$  (—),  $I_P$  (—+),  $I_{MC}$  (—◆),  $I_R$  (—▲), and  $I_R^{lin}$  (—×)

??

# Rationality & Tractability?

	Monotonicity	Continuity	Progression
Drastic	✓	✗	✗
#violations	✓/✗	✗	✓
#problematic	✓/✗	✗	✓
#repairs	✗	✓	✗
repair_cost	✓	✓	✓

→ NP-hard for DCs [Lopatenko-Bertossi-07] ; even FDs [Livshits-K-Roy-18]

*Tractable measure with all 3?*

# Repair-Cost as an ILP

---

minimize:  $\sum_{t \in D} x_t$

$x_t$  for every tuple  $t$   
 $x_t = 1$  : delete  $t$

subject to:  $\forall_{\text{violation } c} \sum_{t \in C} x_t \geq 1$

$$\forall_t x_t \in \{0,1\}$$

Recall: min set of  
tuples that violates  
a constraint (DC)

# Linear Relaxation

minimize:  $\sum_{t \in D} x_t$

$x_t$  for every tuple  $t$   
 $x_t = 1$  : delete  $t$

subject to:  $\forall_{\text{violation } c} \sum_{t \in C} x_t \geq 1$

Recall: min set of  
tuples that violates  
a constraint (DC)

~~$\forall_t x_t \in \{0, 1\}$~~   $0 \leq x_t \leq 1$

# Rationality & Tractability?

	Monotonicity	Continuity	Progression
Drastic	✓	✗	✗
#violations	✓/✗	✗	✓
#problematic	✓/✗	✗	✓
#repairs	✗	✓	✗
repair_cost	✓	✓	✓
frac_cost	✓	✓	✓

→ Poly. time

*Tractable measure with all 3?*

# Outline

1. Introduction & Background

2. Inconsistency Measures

 3. Complexity of Calculation

we are here

4. Probabilistic Database Viewpoint

5. Responsibility Attribution

6. Concluding Remarks

# Complexity Analysis

---

- We now study the complexity of computing the basic measures
- Restrictions:
  - Functional dependencies
    - Some results apply to denial constraints... and to any anti-monotonic constraints where we can materialize all (minimal) violations
  - Coarse-grained complexity (exptime vs. ptime)



# Basic Inconsistency Measures

- **Drastic:** 1 or 0 (inconsistent or consistent)
  - [Thimm-17] Polynomial time (basic SQL)
- **#violations** (i.e., set-min inconsistent subsets)
  - [Kivinen-Mannila-95] [Hunter-Konieczny-08] (“MI Shapley Inconsistency”) Polynomial time (basic SQL)
- **#problematic** tuples (i.e., tuples in violations)
  - [Kivinen-Mannila-95] [Grant-Hunter-11] Polynomial time (basic SQL)
- **#repairs:** number of maximal consistent subsets
  - [Grant-Hunter-11] Next
- **repair\_cost:** minimal #tuples to delete to attain consistency (**cardinality repair**) Next
  - [Huhtala+98] [Kivinen-Mannila-95] [Grant-Hunter-13] [Bertossi-18] [Rammelaere-Geerts-18] (constraint “confidence”)

# Studied Computational Problems

**Problem 1:** *Compute a Cardinality Repair* (repair\_cost)

**Params:** Relation schema  $S$  ; set  $\Sigma$  of constraints

**Input:** Relation  $D$  over  $S$

**Goal:** Find a smallest  $E \subseteq D$  s.t.  $D \setminus E$  satisfies  $\Sigma$

**Greatest**  
consistent  
subset

**Problem 2:** *Repair Counting* (#repairs)

**Params:** Relation schema  $S$  ; set  $\Sigma$  of constraints

**Input:** Relation  $D$  over  $S$

**Goal:** Compute the number of repairs of  $D$  w.r.t.  $\Sigma$

**Set-max**  
consistent  
subsets

# Data Complexity

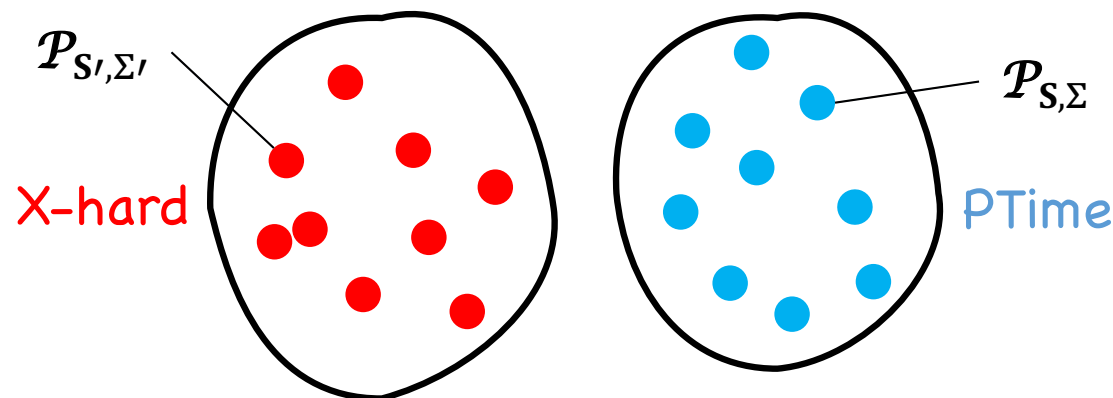
---

- Typically, the problems we consider involve:
  - A database  $D$  (typically one relation)
  - A set  $\Sigma$  of constraints
  - Both  $D$  and  $\Sigma$  are over a relational schema  $S$
- When we analyze the complexity of problems, we adopt the conventional **data complexity** [Vardi-82]
- Hence, the input consists of only the database  $D$ ; everything else (e.g.,  $S$  and  $\Sigma$ ) is fixed
  - Treated as *parameters*
- Hence, every  $S$  and  $\Sigma$  give rise to a separate computational problem  $\mathcal{P}_{S,\Sigma}$
- Possible that one  $\mathcal{P}_{S,\Sigma}$  is tractable & other  $\mathcal{P}_{S',\Sigma'}$  is hard

# Classifications (Dichotomies)

---

- In our case, *every set of functional dependencies can have a different complexity*
- Hence, we aim for **complete characterizations** that will determine the complexity of **every** set of functional dependencies
  - A.k.a. **dichotomy results** or **meta-theorems**



# Problem 1: Cardinality Repair

Compute a Cardinality Repair ( $\text{repair\_cost}$ )

**Params:** Relation schema  $\mathbf{S}$  ; set  $\Sigma$  of constraints

**Input:** Relation  $D$  over  $\mathbf{S}$

**Goal:** Find a smallest  $E \subseteq D$  s.t.  $D \setminus E$  satisfies  $\Sigma$

Fixed

Weighted version: tuples have cost; “*smallest*” replaced w/ “*least total score*”

Same as computing the size of such  $E$  w/o finding  $E$  itself

# Vertex Cover with Structure

person  $\rightarrow$  birthCity

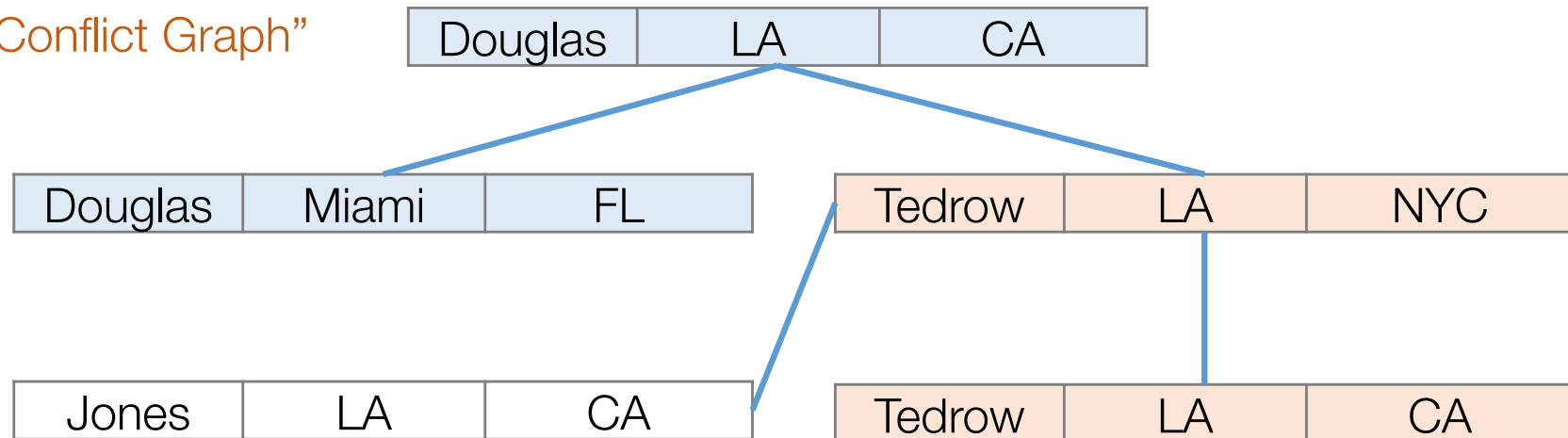
birthCity  $\rightarrow$  birthState

person	birthCity	birthState
Douglas	LA	CA
Douglas	Miami	FL
Tedrow	LA	CA
Tedrow	LA	NYC
Jones	LA	CA

**Note:** While minimum VC is NP-hard, the conflict graphs are **not** general graphs; they are **special** graphs defined by relations and a fixed set of FDs

Cardinality repair of **D**  
= min VC of the conflict graph

“Conflict Graph”



# Example

---

$\Sigma = \{\text{fid} \rightarrow \text{fname}, \text{fname} \rightarrow \text{fid}, \text{fid} \rightarrow \text{city}, \text{fid room} \rightarrow \text{floor}\}$

fid	fname	room	floor	city
F01	HQ	322	3	Paris
F02	HQ	122	30	Madrid
F02	HQ	122	1	Madrid
F03	Lab1	B35	3	London
F01	Lab1	B25	2	London

# Simplification 1: Common lhs

$$\Sigma = \{\overset{x}{\text{facility}} \rightarrow \text{city}, \overset{x}{\text{facility}} \text{ room} \rightarrow \text{floor}\}$$



$$\{\emptyset \rightarrow \text{city}, \text{room} \rightarrow \text{floor}\}$$

facility	room	floor	city
HQ	322	3	Paris
HQ	322	30	Madrid
HQ	122	1	Madrid
Lab1	B35	3	London



# Simplification 2: Consensus

---

$$\Sigma = \{\overset{x}{\emptyset} \rightarrow \overset{x}{\text{city}}, \text{room} \rightarrow \text{floor}\}$$



$$\{\text{room} \rightarrow \text{floor}\}$$


facility	room	floor	city
HQ	322	3	Paris
HQ	322	30	Madrid
HQ	122	1	Madrid

# Simplification 3: Matching

$$\Sigma = \{\overset{x}{\text{fid}} \rightarrow \overset{x}{\text{fname}}, \overset{x}{\text{fname}} \rightarrow \overset{x}{\text{fid}}, \overset{x}{\text{fid}} \rightarrow \text{city}, \overset{x}{\text{fid}} \text{ room} \rightarrow \text{floor}\}$$

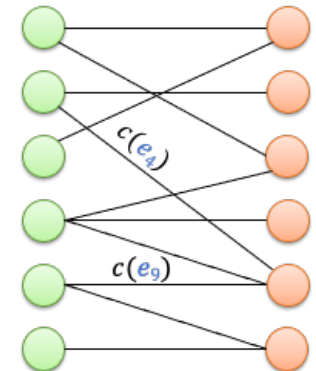


$$\{\emptyset \rightarrow \text{city}, \text{room} \rightarrow \text{floor}\}$$



fid	fname	room	floor	city
F01	HQ	322	3	Paris
F02	HQ	122	30	Madrid
F02	HQ	122	1	Madrid
F03	Lab1	B35	3	London
F01	Lab1	B25	2	London

Reduction to  
**Maximum-Weight  
Matching** of a  
bipartite graph



# Repeated Simplification

---

$$\Sigma = \{\overset{x}{\text{fid}} \rightarrow \overset{x}{\text{fname}}, \overset{x}{\text{fname}} \rightarrow \overset{x}{\text{fid}}, \overset{x}{\text{fid}} \rightarrow \text{city}, \overset{x}{\text{fid}} \text{ room} \rightarrow \text{floor}\}$$



$$\{\emptyset \rightarrow \text{city}, \text{room} \rightarrow \text{floor}\}$$



$$\{\text{room} \rightarrow \text{floor}\}$$



$$\{\emptyset \rightarrow \text{floor}\}$$



$$\{\}$$

# The Unified Simplification Rule

---

Let  $\Sigma$  be a set of FDs,  $X, Y$  attribute sets such that:

1. Sets  $X$  and  $Y$  functionally determine each other  
i.e.,  $\text{Closure}_{\Sigma}(X) = \text{Closure}_{\Sigma}(Y)$
  2. Every FD in  $\Sigma$  contains either  $X$  or  $Y$  in its lhs
- 

Finding a cardinality repair under  $\Sigma$

reduces in polynomial time to

finding a cardinality repair under  $\Sigma - XY$ .

# Example 1: $X = Y$ (Common lhs)

$\Sigma = \{\overset{x}{\text{facility}} \rightarrow \text{city}, \overset{x}{\text{facility}} \text{ room} \rightarrow \text{floor}\}$



$\{\emptyset \rightarrow \text{city}, \text{room} \rightarrow \text{floor}\}$

$X = \{\text{facility}\}$

$Y = \{\text{facility}\}$

facility	room	floor	city
HQ	322	3	Paris
HQ	322	30	Madrid
HQ	122	1	Madrid

Lab1	B35	3	London
------	-----	---	--------

## Example 2: $X = \emptyset$ (Consensus)

$$\Sigma = \{\overset{x}{\emptyset} \rightarrow \overset{x}{\text{city}}, \text{room} \rightarrow \text{floor}\}$$



$$\{\text{room} \rightarrow \text{floor}\}$$

$$X = \emptyset$$

$$Y = \{\text{city}\}$$

facility	room	floor	city
HQ	322	3	Paris

HQ	322	30	Madrid
HQ	122	1	Madrid

# Example 3: General X,Y (Matching)

$\Sigma = \{\overset{x}{\text{fid}} \rightarrow \overset{x}{\text{fname}}, \overset{x}{\text{fname}} \rightarrow \overset{x}{\text{fid}}, \overset{x}{\text{fid}} \rightarrow \text{city}, \overset{x}{\text{fid}} \text{ room} \rightarrow \text{floor}\}$



$\{\emptyset \rightarrow \text{city}, \text{room} \rightarrow \text{floor}\}$

$X = \{\text{fname}\}$

$Y = \{\text{fid}\}$

fid	fname	room	floor	city
F01	HQ	322	3	Paris
F02	HQ	122	30	Madrid
F02	HQ	122	1	Madrid
F03	Lab1	B35	3	London
F01	Lab1	B25	2	London

# Completeness

---

- Simplification rule simplifies the computation of `repair_cost` by eliminating attributes and dependencies
- Not an arbitrary algorithmic trick...
- It is **complete** for computing `repair_cost`!



### THEOREM [Livshits-K-Roy-20]

Fix any set  $\Sigma$  of FDs. The following are equivalent (under standard complexity assumptions):

1. The measure  $\text{repair\_cost}(\Sigma, \cdot)$  can be computed (and a cardinality repair can be found) in poly-time
2. The FD set  $\Sigma$  can be **simplified until emptied**

# Proof Technique: Fact-Wise Reduction

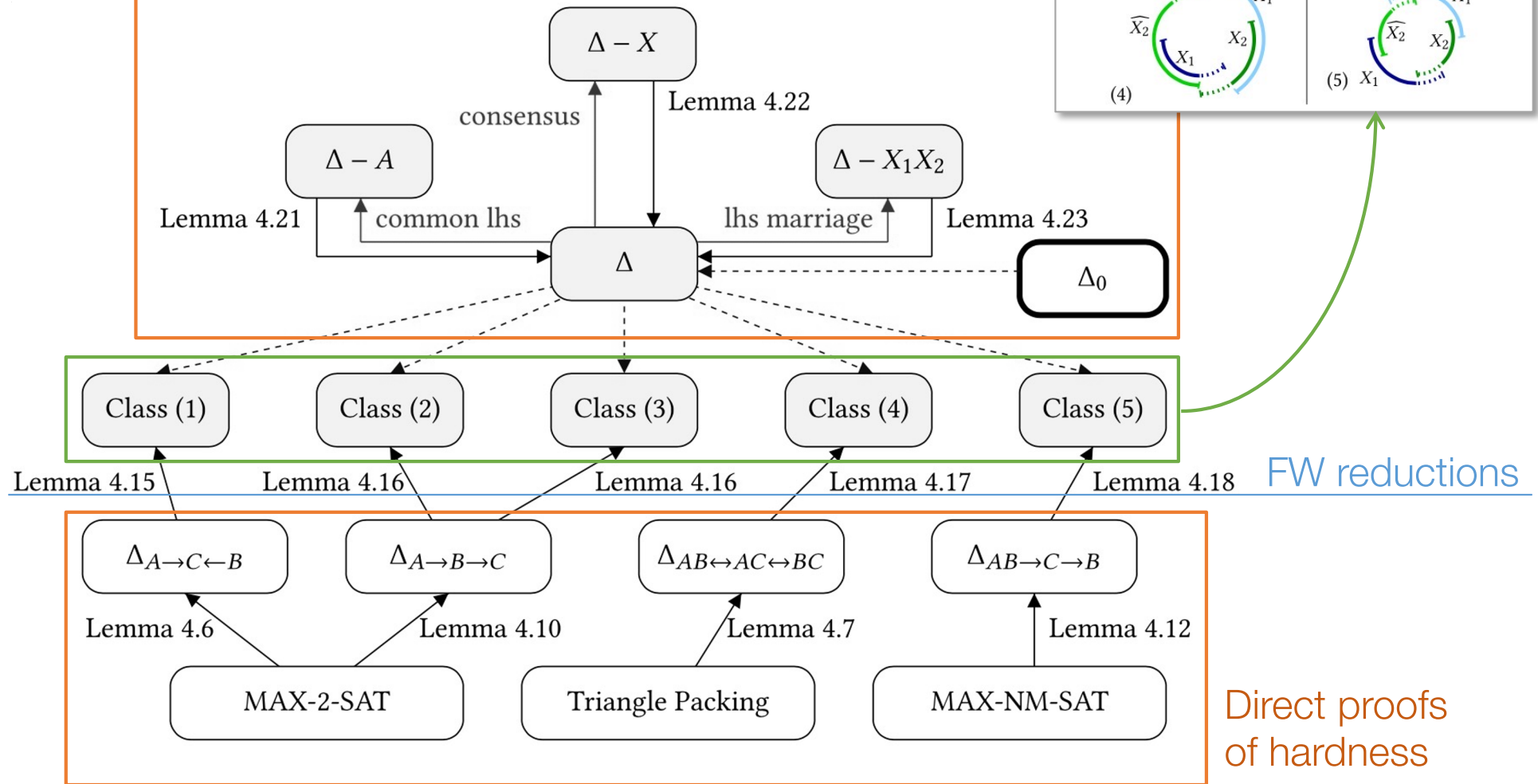
- How do we prove hardness for *infinitely many* FD sets?
- A common approach is the **fact-wise reduction**

Let  $S_1$  and  $S_2$  be database schemes with the constraints  $\Sigma_1$  and  $\Sigma_2$ . A *fact-wise reduction* is a mapping from facts  $R_1(a_1, \dots, a_n)$  over  $S_1$  to facts  $R_2(b_1, \dots, b_m)$  over  $S_2$  that:

- Is injective (one-one)
    - Examples:  $(a, b) \Rightarrow (a, a.b, b)$      $(a, b, c) \Rightarrow (b, a.b, c)$
  - Preserves consistency and inconsistency
  - Is computable in polynomial time
- General mechanism to translate (reduce) problems on  $(S_1, \Sigma_1)$  to problems on  $(S_2, \Sigma_2)$  ; if former hard, so latter

# About the Proof of Hardness

## Simplification



# Approximations

---

- A 2-approx of repair-cost can be obtained easily using a 2-approx for Vertex Cover
  - Can be generalized to **denial constraints** (constant approx)
- [Miao et al.] used the dichotomy and fact-wise reductions in follow-up work on approx bounds for cardinality repair
- Improved known upper bounds for general VC in the case of conflict graphs, based on the set of FDs
- Developed optimization techniques and heuristics to establish efficient, high-quality approximations

# Problem 2: Counting Repairs

---

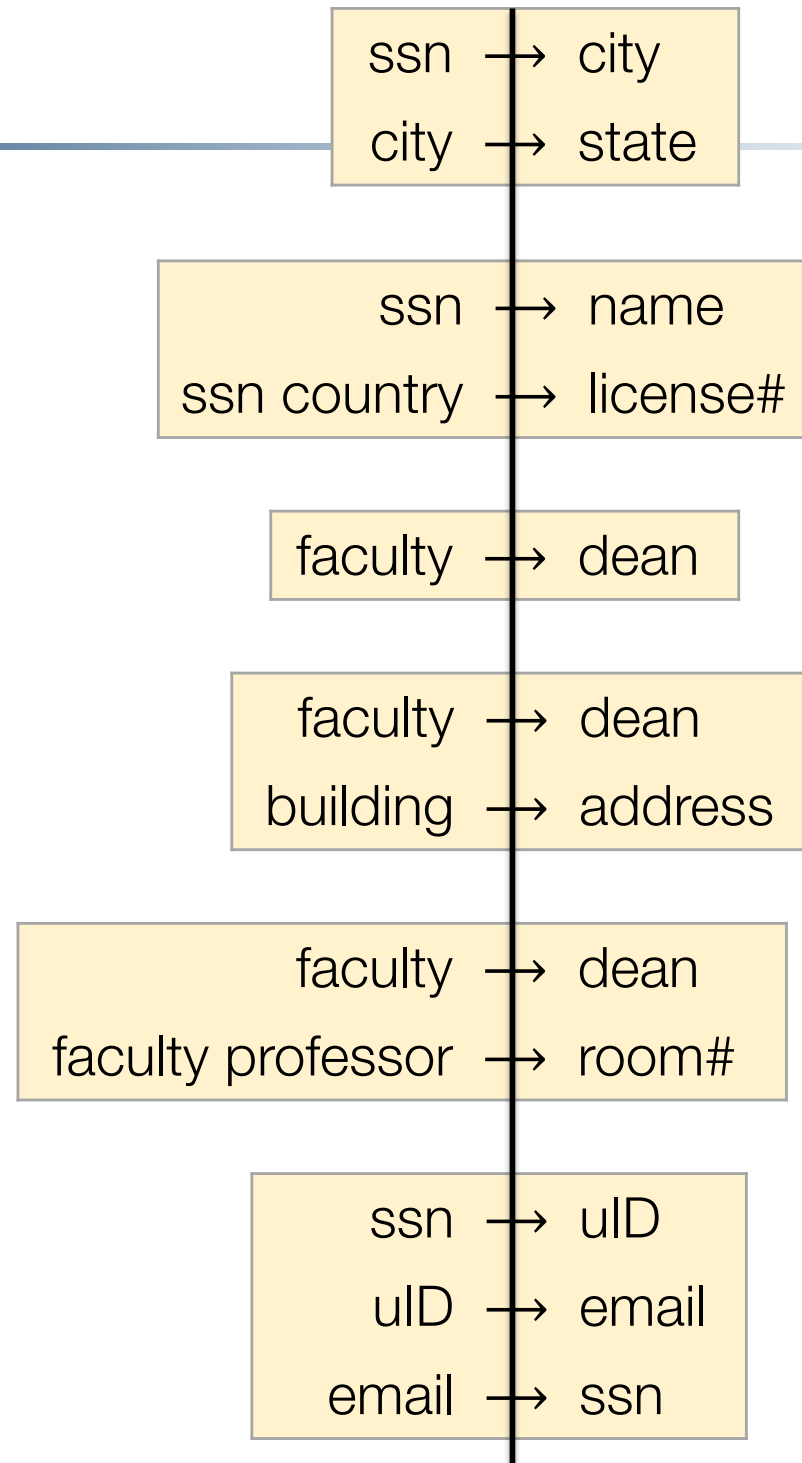
**Problem 2:** *Repair Counting* (#repairs)

**Params:** Relation schema  $S$  ; set  $\Sigma$  of constraints

**Input:** Relation  $D$  over  $S$

**Goal:** Compute the number of repairs of  $D$  w.r.t.  $\Sigma$

# Examples



## Hard

ssn  $\rightarrow$  city  
city  $\rightarrow$  state

faculty  $\rightarrow$  dean  
building  $\rightarrow$  address

ssn  $\rightarrow$  uID  
uID  $\rightarrow$  email  
email  $\rightarrow$  ssn

## Poly time

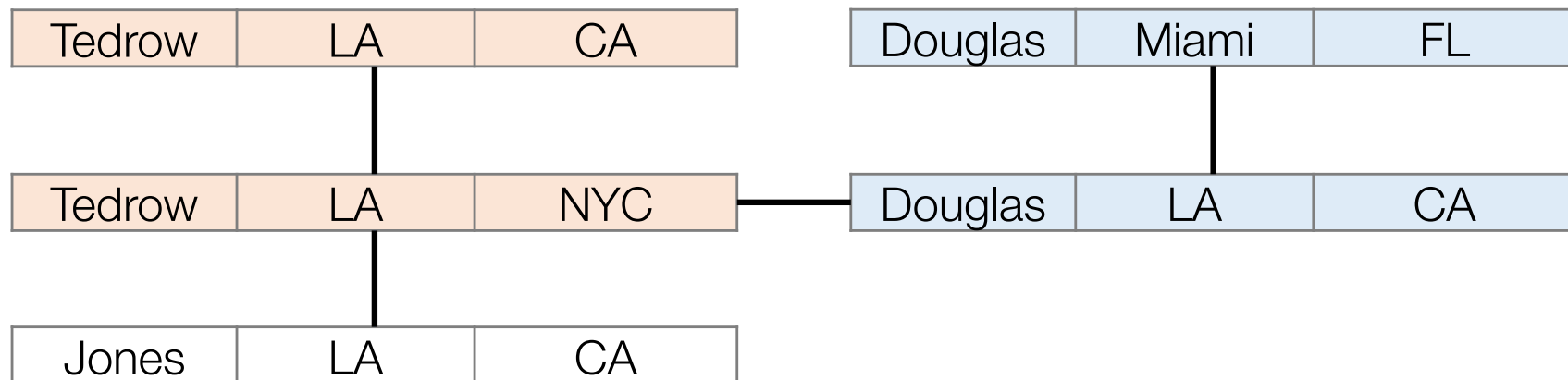
ssn  $\rightarrow$  name  
ssn country  $\rightarrow$  license#

faculty  $\rightarrow$  dean

faculty  $\rightarrow$  dean  
faculty professor  $\rightarrow$  room#

# Repair Counting as MIS Counting

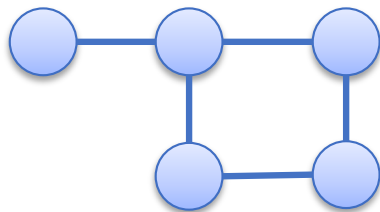
- For FDs, a repair is a **Maximal Independent Set (MIS)** of the **conflict graph** of the database
- Hence, repair counting amounts to MIS counting
  - Over conflict graphs
  - Again, these are not general graphs...



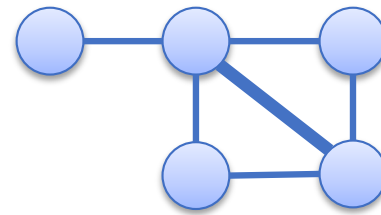


# Counting Set-Minimal Repairs

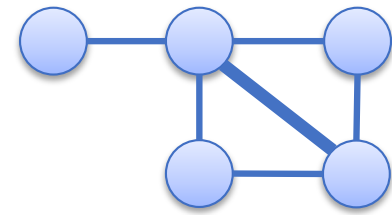
- MIS counting is **#P-complete** [Provan-Ball-83] and inapproximable [Roth-96]
- Special tractable cases, e.g.,  **$P_4$ -free** graphs
  - **$P_4$ -free** graph (a.k.a. **cograph**): no induced path of length 4
- *What about the conflict graphs?*
- If the constraints are such that every conflict graph is  $P_4$ -free, then the repairs can be counted in poly. time
- *This is also a necessary condition!*



Not  $P_4$ -free



$P_4$ -free



# Dichotomy Theorem

---

## THEOREM [Livshits-K-Wijzen-21]

The following are equivalent (under standard complexity assumptions) for *every fixed set of FDs*:

1. Repairs can be counted in **poly. time**
2. Every conflict graph is  **$P_4$ -free**

Extension: classification for counting repairs that satisfy a CQ (w/o self-joins) [Calautti+22]

# Tractable Characterization: lhs-Chain

The property that every conflict graph is  $P_4$ -free has a syntactic characterization:

## THEOREM [Livshits-K-Wijzen-21]

The following are equivalent for every set  $\Sigma$  of FDs:

1. Every conflict graph is  $P_4$ -free
2. For every two FDs  $X_1 \rightarrow Y_1$  and  $X_2 \rightarrow Y_2$ , either  $X_1 \subseteq X_2$  or  $X_2 \subseteq X_1$ 
  - Up to equivalence!

$X_1 \rightarrow Y_1, \dots, X_m \rightarrow Y_m$   
so that  $X_1 \subseteq X_2 \subseteq \dots \subseteq X_m$

Testing: take a **minimal cover** of  $\Sigma$  (i.e., remove redundancy) and test whether it is *syntactically* an lhs-chain

Hard

Poly time

Approx.  
open...

Coincides w/  
long-standing  
open problem  
(#max  
matchings)

ssn  $\rightarrow$  city  
city  $\rightarrow$  state

ssn  $\rightarrow$  name  
ssn country  $\rightarrow$  license#

faculty  $\rightarrow$  dean  
building  $\rightarrow$  address

faculty  $\rightarrow$  dean

Inapproximable  
[Calautti+22]

ssn  $\rightarrow$  uID  
uID  $\rightarrow$  email  
email  $\rightarrow$  ssn

faculty  $\rightarrow$  dean  
faculty professor  $\rightarrow$  room#

# Proof Structure

## What we need to prove:

Let  $\Sigma$  be a set of FDs.

1. If  $\Sigma$  is an lhs-chain up to equivalence, then the conflict graph is  $P_4$ -free.
2. Otherwise repair counting is #P-hard.

The proof is fairly simple:

1. If  $\Sigma$  is an lhs-chain, conflict graph is  $P_4$ -free
  - Use a known characterization of  $P_4$ -freeness: *cograph*
2. If  $\Sigma$  is **not** an lhs-chain:
  - Take a minimal cover and use it to construct a small example w/ induced  $P_4$
  - Show a fact-wise reduction from  $\{A \rightarrow B, B \rightarrow A\}$ 
    - Hardness for  $\{A \rightarrow B, B \rightarrow A\}$  is easy

Use it again later...

# Outline

1. Introduction & Background

2. Inconsistency Measures

3. Complexity of Calculation

▶ 4. Probabilistic Database Viewpoint

we are here

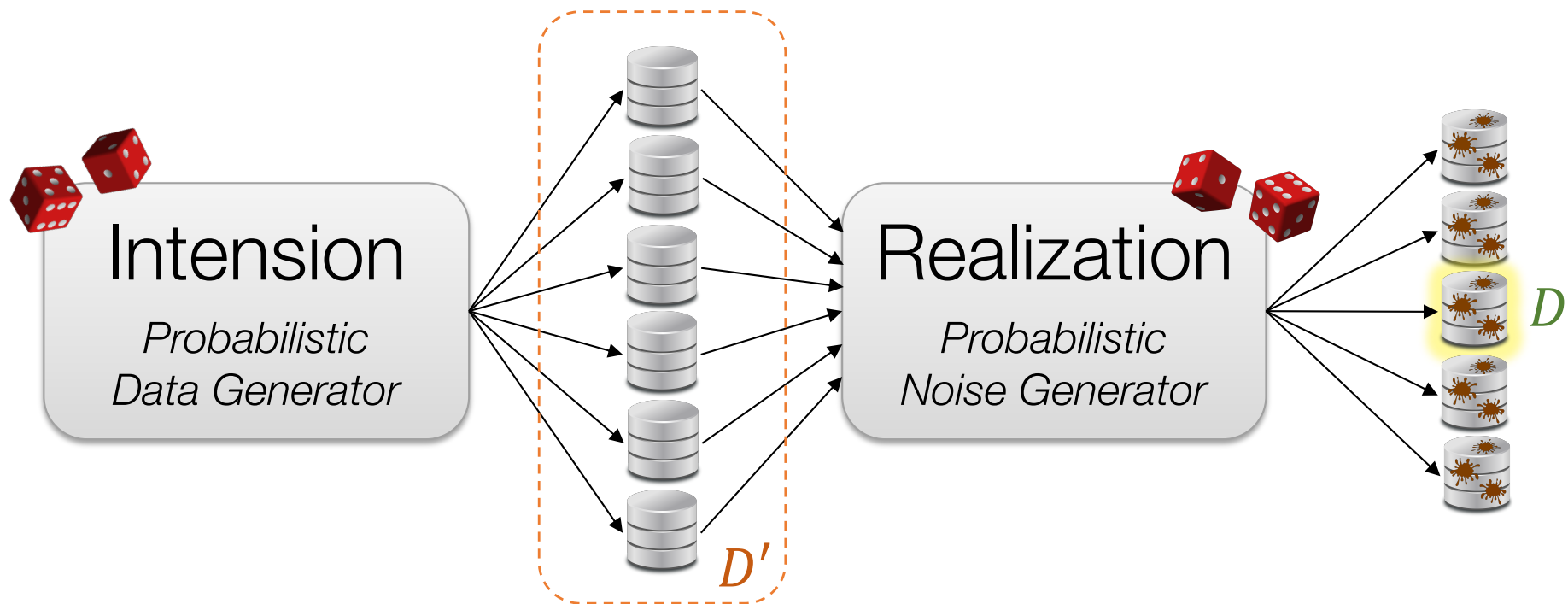
5. Responsibility Attribution

6. Concluding Remarks

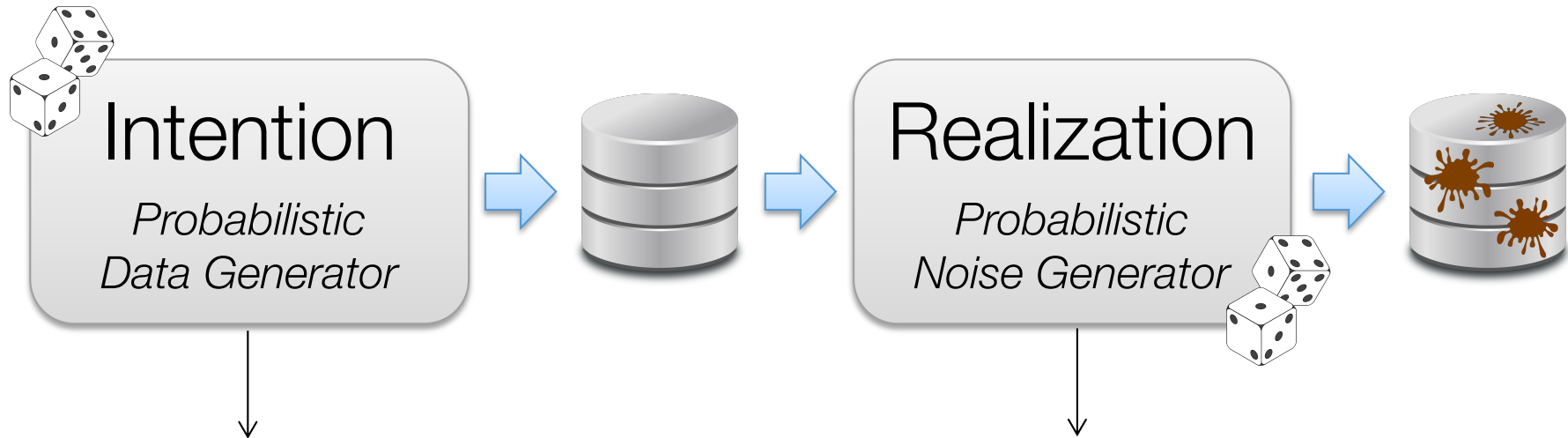
# Noisy Data as a Noisy Channel

The Probabilistic Unclean Data (**PUD**) model [DeSa-Ilyas-K-Ré-Rekatsinas-18]

- Examples:
  - HoloClean [Rekatsinas-Chu-Ilyas-Ré-17]
  - HoloDetect [Heidari-McGrath-Ilyas-Rekatsinas-19]



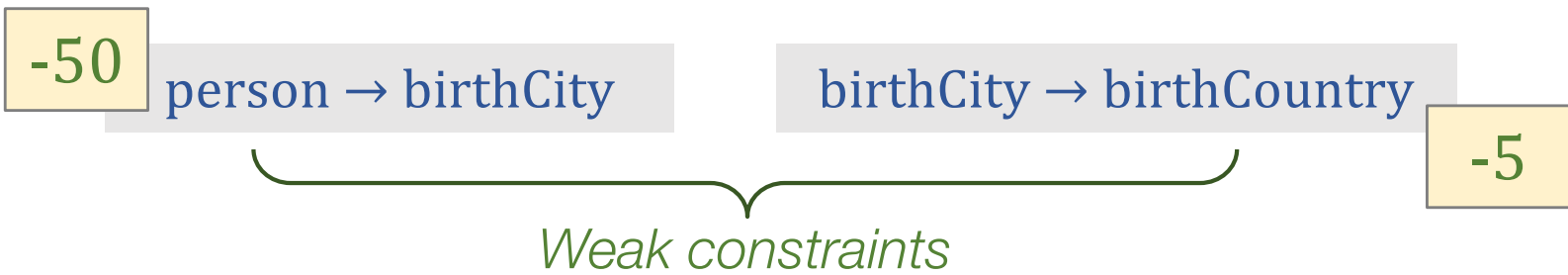
# PUD Example 1: Update Repairs



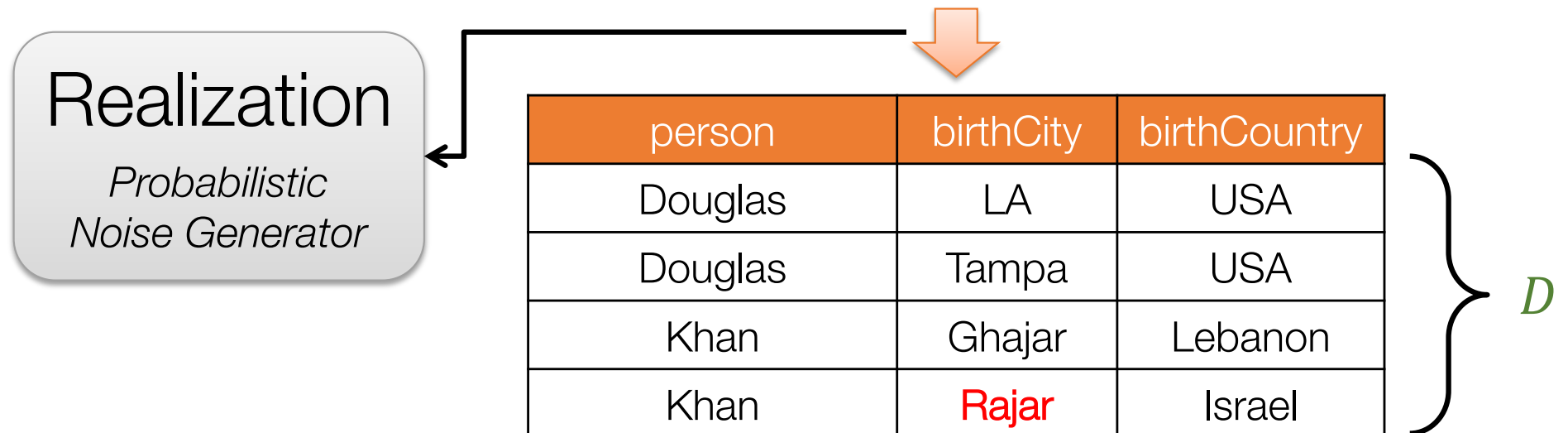
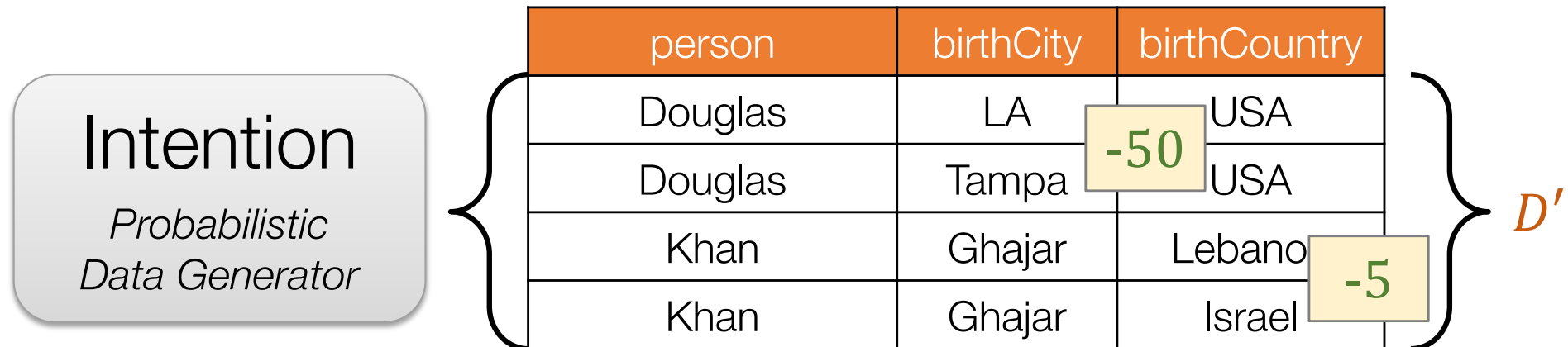
- Uniform, i.i.d. tuple generation
- Markov logic (factors) for weak constraints

Randomly **change**  
**cell values**

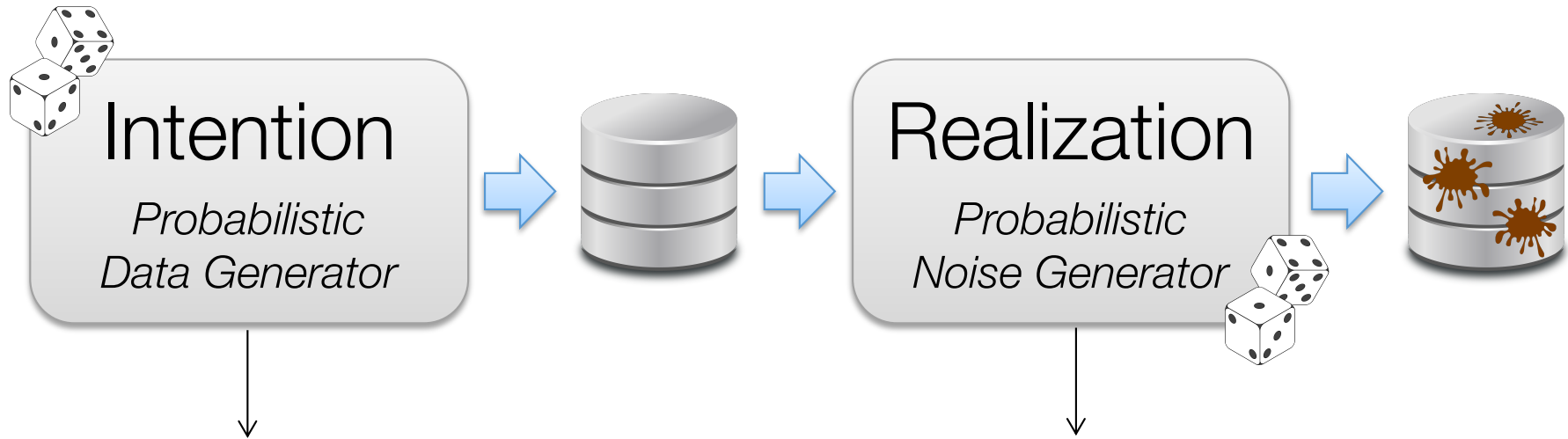




Markov Logic:  $\Pr(D') \sim \exp(\sum \text{penalties}(D'))$



# PUD Example 2: Subset Repairs



- Uniform, i.i.d. tuple generation
- Markov logic (factors) for weak constraints

Randomly **add**  
**new tuples**

-50

person → birthCity

birthCity → birthCountry

-5

$$\Pr(D') \sim \exp(\sum \text{penalties}(D'))$$

Intention

*Probabilistic  
Data Generator*

person	birthCity	birthCountry
Douglas	LA	USA
Douglas	Tampa	USA
Khan	Ghajar	Lebanon
Khan	Ghajar	Israel
Khan	NYC	USA

$D'$

$D$

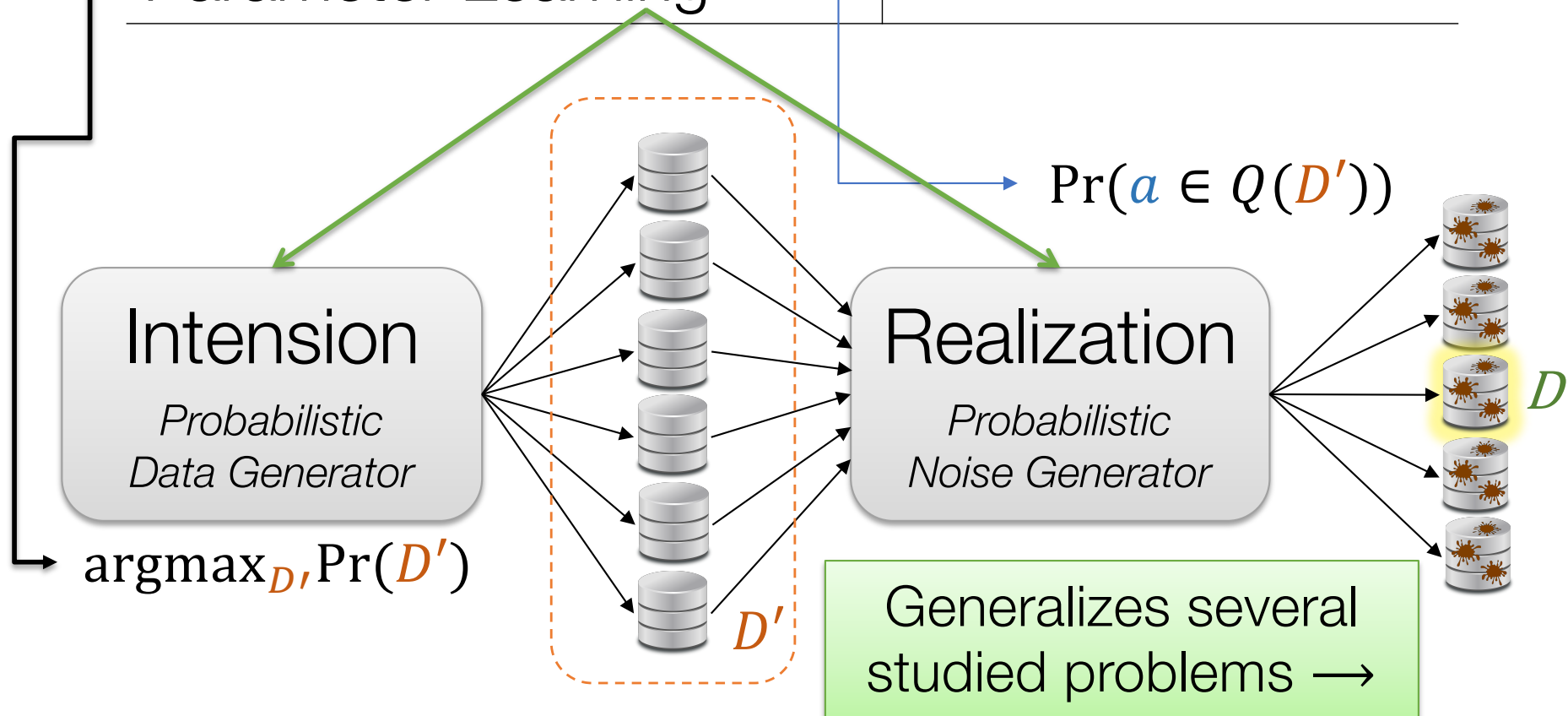
Realization

*Probabilistic  
Noise Generator*

Douglas	NYC	NY
Khan	Ghajar	Syria

# Fundamental Problems

Problem	Deterministic Variant
Most Likely Intent	Repair generation
Prob. Query Answering	Consistent Query Answering, repair counting
Parameter Learning	



# Probabilistic Duplicates [Andritsos-Fuxman-Miller-06]

person  $\rightarrow$  birthCity, birthState

				person	birthCity	birthState	$p$
indep.	disjoint	{		Cullen Douglas	LA	CA	0.6
				Cullen Douglas	Tampa	FL	0.4
	disjoint	{		Marion Jones	LA	CA	1.0
	disjoint	{		Irene Tedrow	NYC	NY	0.3
				Irene Tedrow	LA	FL	0.4
				Irene Tedrow	Hollywood	FL	0.2
				Irene Tedrow	Hollywood	CA	0.1

Later termed **Block-Independent Databases** (BID) [Dalvi-Ré-Suciu-11]

# Beyond Key Constraints?

---

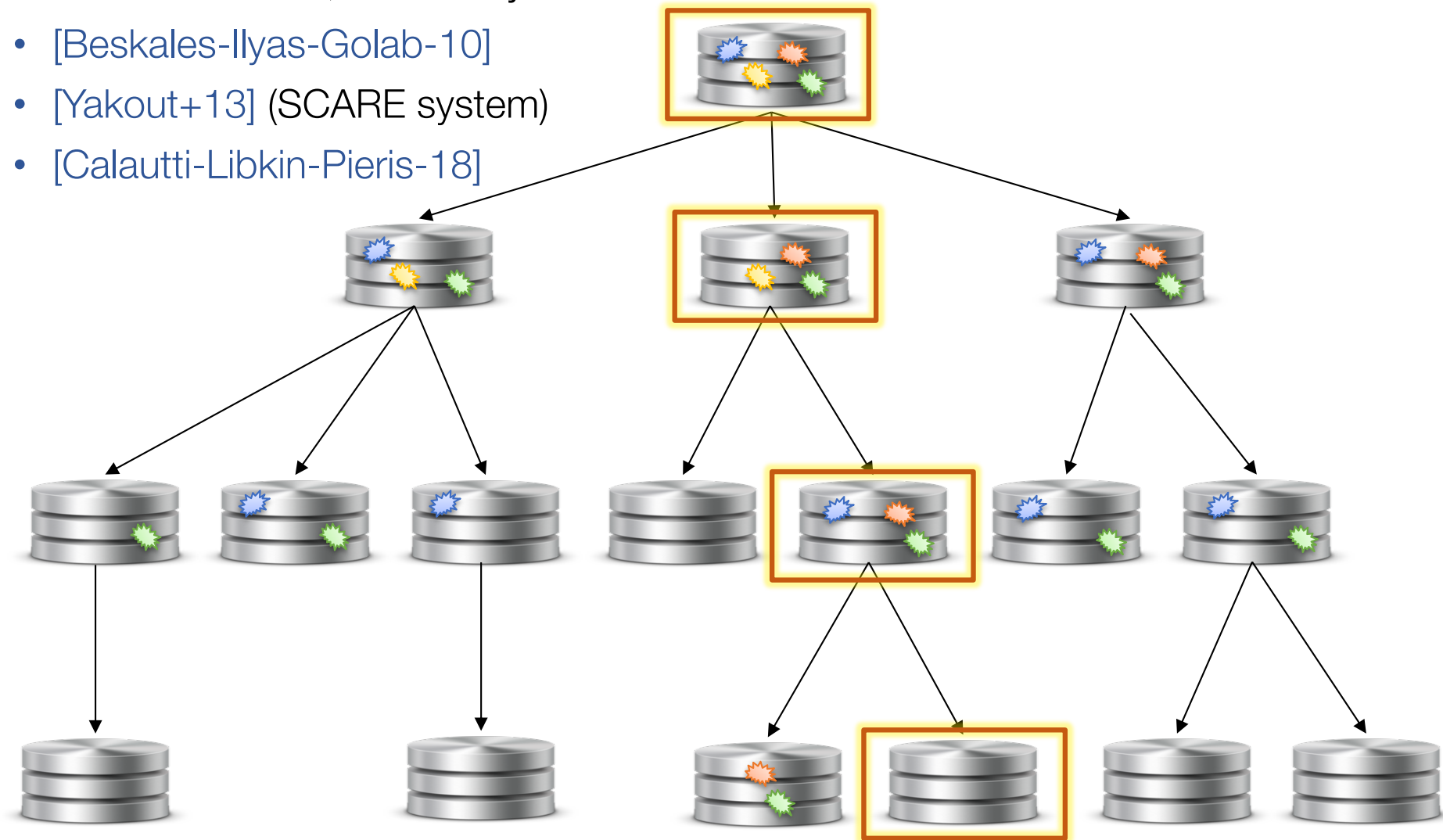
person  $\rightarrow$  birthCity  
birthCity  $\rightarrow$  birthState

person	birthCity	birthState
Cullen Douglas	LA	CA
Cullen Douglas	Tampa	FL
Marion Jones	LA	CA
Irene Tedrow	NYC	NY
Irene Tedrow	LA	FL
Irene Tedrow	Hollywood	FL
Irene Tedrow	Hollywood	CA

# Approach 1: Repair as Markov Chain

Idea: Iteratively select violations and fix, randomly

- [Beskales-Ilyas-Golab-10]
- [Yakout+13] (SCARE system)
- [Calautti-Libkin-Pieris-18]



# Approach 2: TID Conditioning

$$\Pr(D') = \prod_{t \in D'} p(t) \times \prod_{t \notin D'} (1 - p(t))$$

person	city	state	$p$
Cullen	LA	CA	0.6
Cullen	Tampa	FL	0.4
Marion	LA	CA	1.0
Irene	NYC	NY	0.3

person	qualification	$p$
Cullen	9	0.3
Cullen	5	0.7
Marion	8	1.0
Irene	9	0.8

[deRougemont-95] [Grädel-Gurevich-Hirsch-98] [Dalvi-Suciu-04]

*Tuple-Independent Database*



# Constrained TID

person  $\rightarrow$  birthCity  
birthCity  $\rightarrow$  birthState

person	birthCity	birthState	$p$
Cullen Douglas	LA	CA	0.6
Cullen Douglas	Tampa	FL	0.7
Marion Jones	LA	CA	0.9
Irene Tedrow	NYC	NY	0.6
Irene Tedrow	LA	FL	0.9
Irene Tedrow	Hollywood	FL	0.5
Irene Tedrow	Hollywood	CA	0.8

$$p(D') = \Pr(D' | C)$$

Computational problem: find a most probable  $D'$  (MPD)

# MPD

person  $\rightarrow$  birthCity  
birthCity  $\rightarrow$  birthState

<i>factor</i>	person	birthCity	birthState	<i>p</i>
1-0.6	Cullen Douglas	LA	CA	0.6
0.7	Cullen Douglas	Tampa	FL	0.7
0.9	Marion Jones	LA	CA	0.9
1-0.6	Irene Tedrow	NYC	NY	0.6
1-0.9	Irene Tedrow	LA	FL	0.9
1-0.5	Irene Tedrow	Hollywood	FL	0.5
0.8	Irene Tedrow	Hollywood	CA	0.8

Can compute efficiently?

$$\max_{\text{consistent } D'} \left( \prod_{t \in D'} p(t) \times \prod_{t \notin D'} (1 - p(t)) \right)$$

# MPD Complexity

---

- [Gribkoff-VanDenBroeck-Suciu-14] studied the computation of an MPD in the case of FDs
- They covered the case of unary FDs (single-attribute on the lhs)
  - With a gap remaining
- They left open the case of a general set of FDs (and the remainder of the unary case)
- *Interestingly, the open problem has been resolved in a different context*
  - ... that we have seen already!

## THEOREM [Livshits-K-Roy-18]

Fix **any set of FDs**. The following are equivalent (under standard complexity assumptions):

1. An **MPD** can be found in poly-time
2. The measure  $\text{repair\_cost}(\Sigma, \cdot)$  can be computed (and a **cardinality repair** can be found) in poly-time

## COROLLARY

Fix any set of FDs. The following are equivalent (under standard complexity assumptions):

1. An **MPD** can be found in poly-time
2. The FD set can be **simplified until emptied** according to the simplification process of Livshits+

# Hardness of Constraints

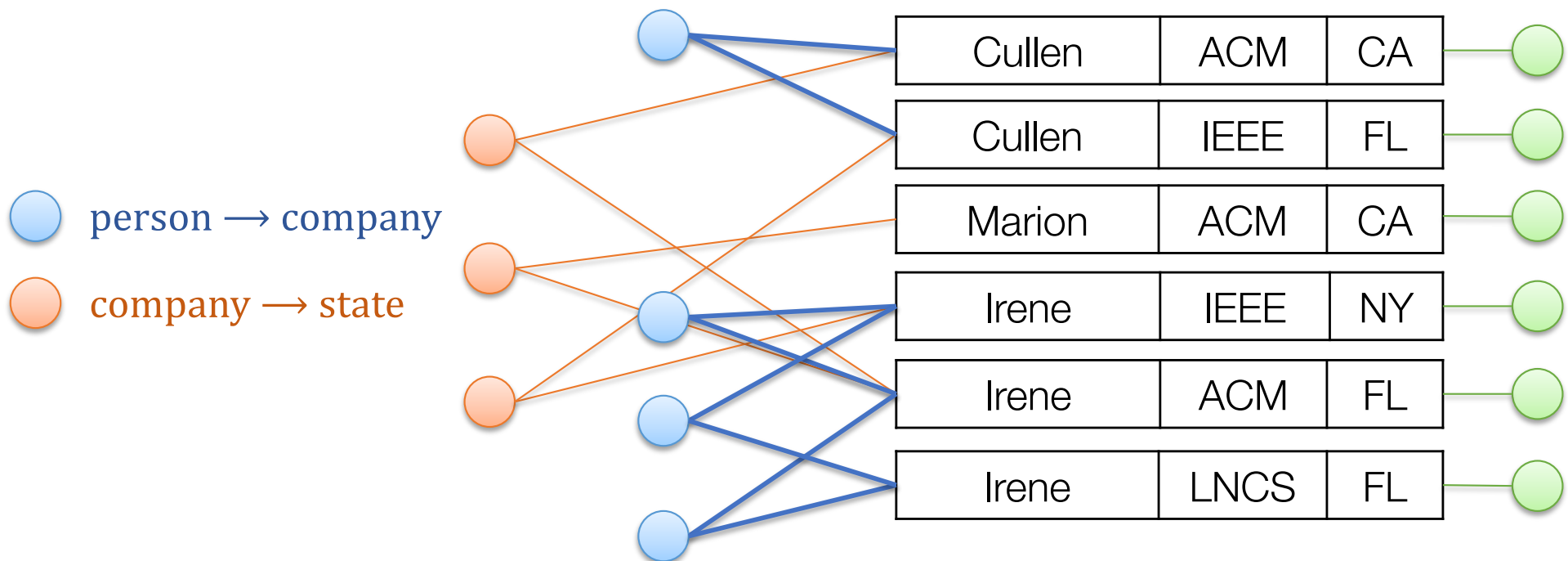
---

- Recall that we started with the problem of finding a most likely repair of a PUD
- The previous results cover the case where constraints are [hard](#) constraints
- *What about **soft constraints**?*
- Still largely open, yet considerable progress
  - [\[Carmeli-Grohe-K-Livshits-Tibi-21\]](#)

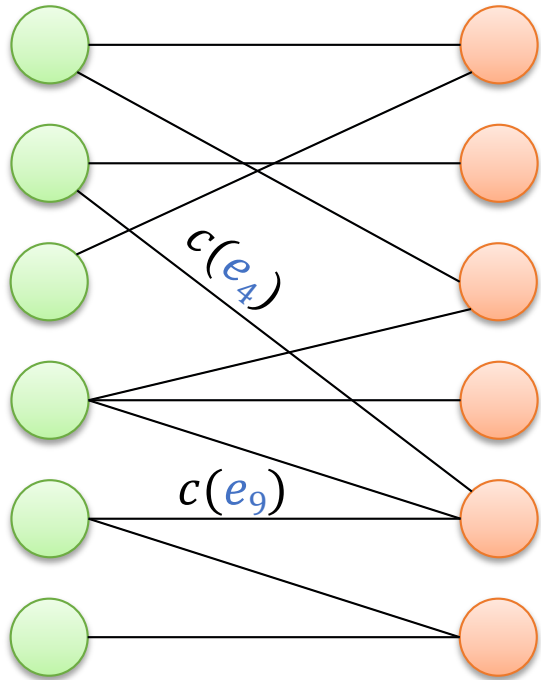
# MPD for Weak Constraints

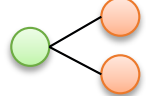
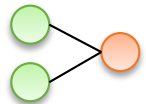
MPD:  $\max_{\text{consistent } D'} \left( \prod_{t \in D'} p(t) \times \prod_{t \notin D'} (1 - p(t)) \right)$

Soft constraints:  $\max_{\text{subset } D'} \left( \prod_{t \in D'} w(t) \times \prod_{\text{FD } \varphi} \prod_{\text{violations } (t, t') \subseteq D'} \frac{1}{\text{cost}(\varphi)} \right)$



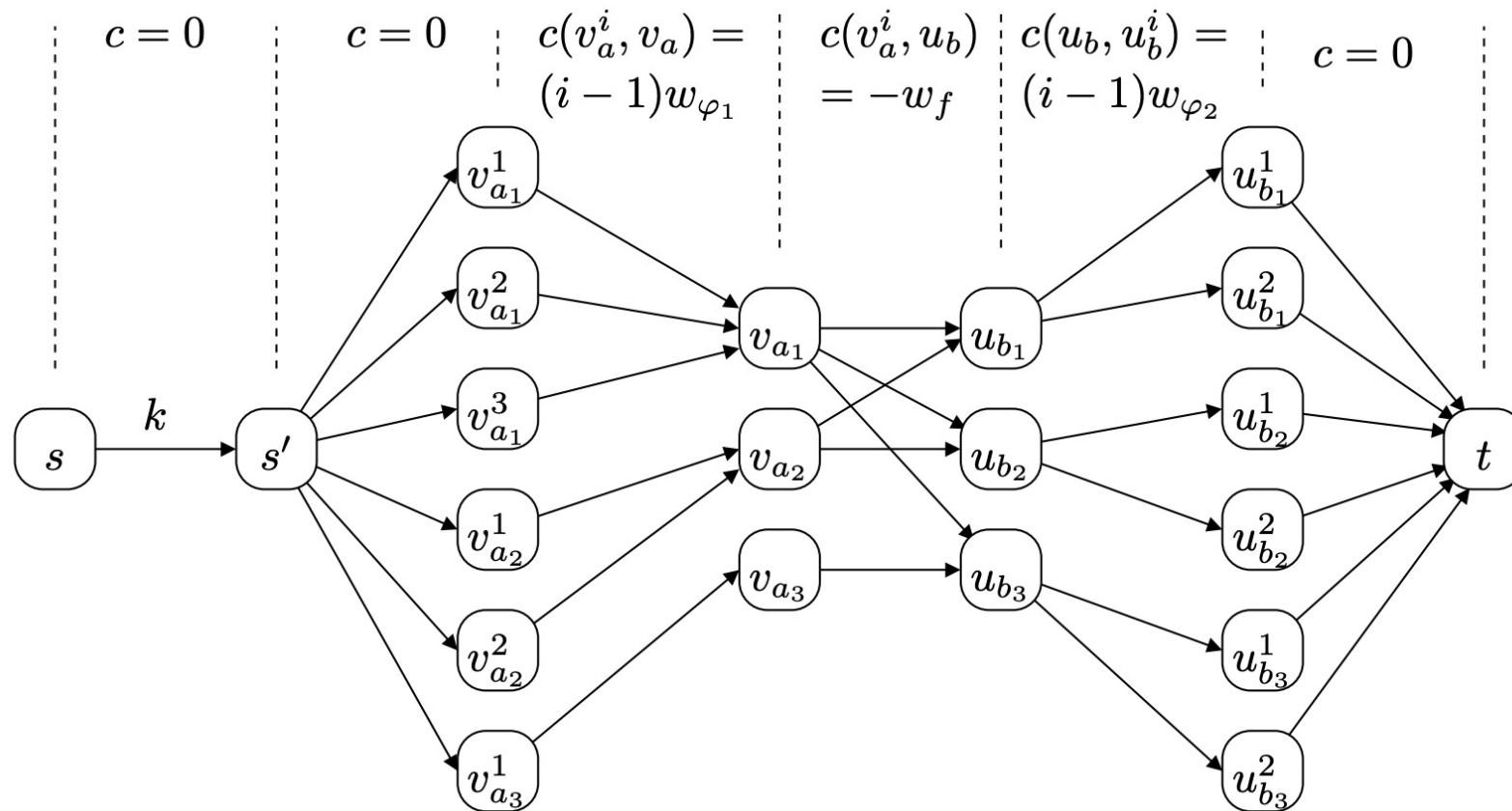
# Example: “Liberal” Matching



- We need to select a subset of the relationships
- We pay a cost  $c(e)$  for denying each relationship  $e$
- We pay a cost  $c_1$  for each 
- We pay a cost  $c_2$  for each 
- Goal: least-cost liberal matching

Algorithm via *minimum-cost maximum flow*  
[Carmeli-Grohe-K-Livshits-Tibi-21]

# Algorithm: Network Flow with Costs



- **Min-cost max-flow:** Given a network with capacities and costs on edges, find a maximal source-to-sink flow with a minimal cost
- Solvable in polynomial time, including the integral variant (capacities and flow are all integers) [Ahuja-Magnanti-Orlin-93]

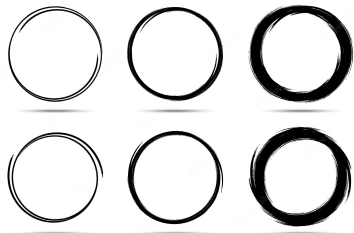


# Outline

1. Introduction & Background
2. Inconsistency Measures
3. Complexity of Calculation
4. Probabilistic Database Viewpoint
- ▶ 5. Responsibility Attribution
6. Concluding Remarks

we are here

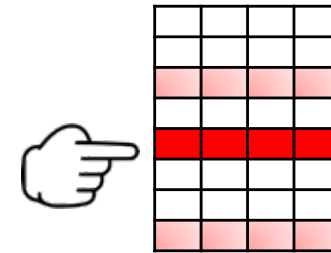
# Usage of Inconsistency Measures



Notions of **soft**  
(weak/approx)  
constraints

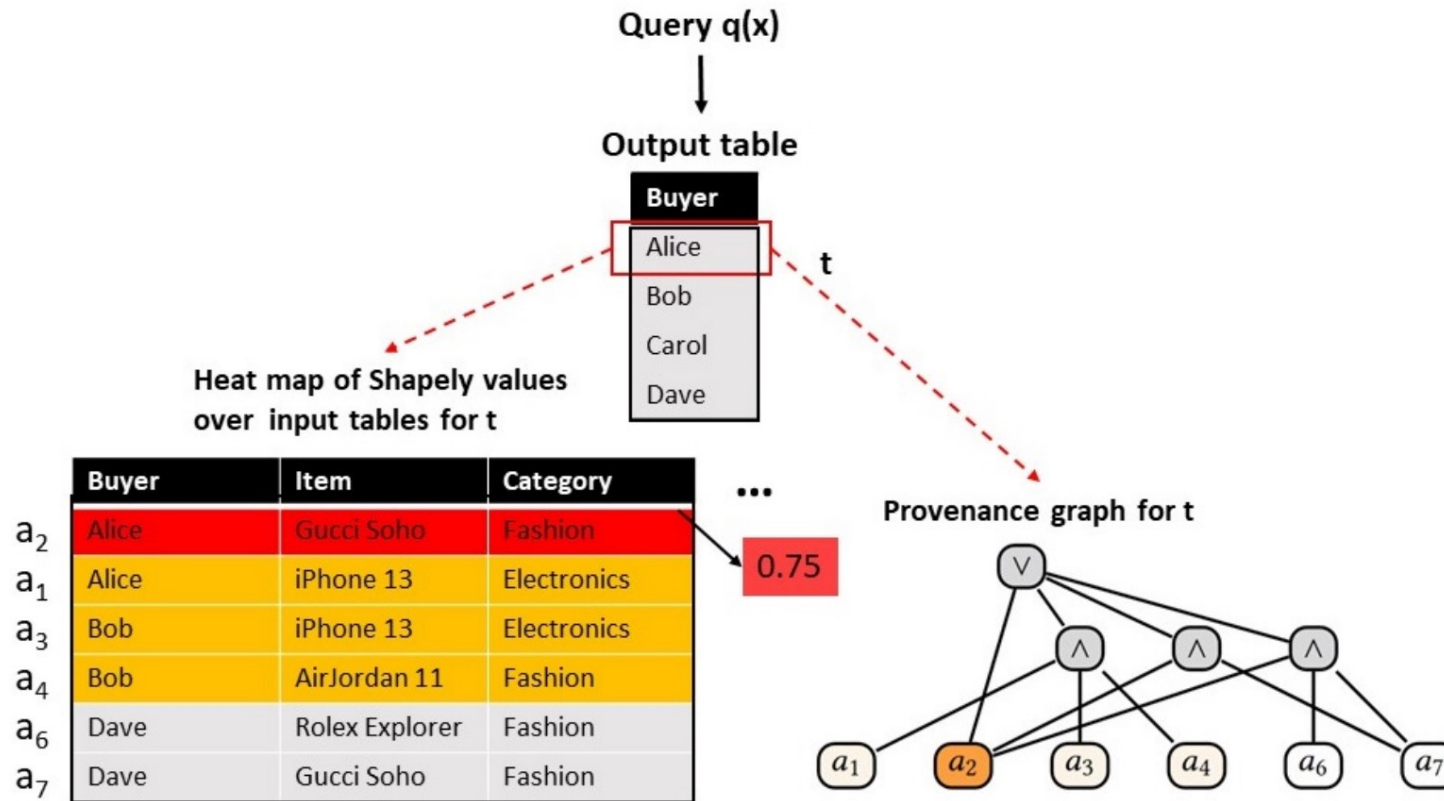


**Progress** indication  
for data repairing  
processes



Attribution of  
**responsibility** to  
inconsistency

# Background: Explaining Query Answers



ShapGraph, SIGMOD-22 Demo  
[Davidson-Deutch-Frost-K-Koren-Monet-22]

# Background: Explaining Query Answers

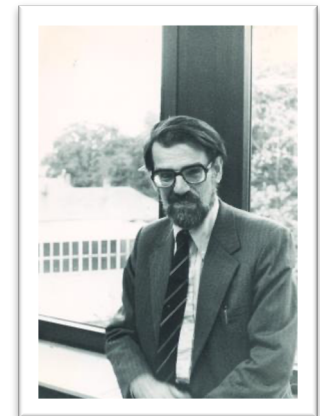
---

- *Which DB tuples explain a query answer?*  
Quantify each tuple's responsibility
- Various past proposals
  - Counterfactual analysis [Melious+10] [Freire+15]
    - Minimal change for the tuple to matter [Chockler-Halpern04]
  - Causal effect [Salimi+16]
    - Based on probabilistic databases
  - **Shapley value** [Livshits+20] (next)
- If the query asks about inconsistency, we get to attribute a responsibility to each tuple
  - In turn, can be used to rank tuples for inspection / fix
- This query can be any **inconsistency measure**

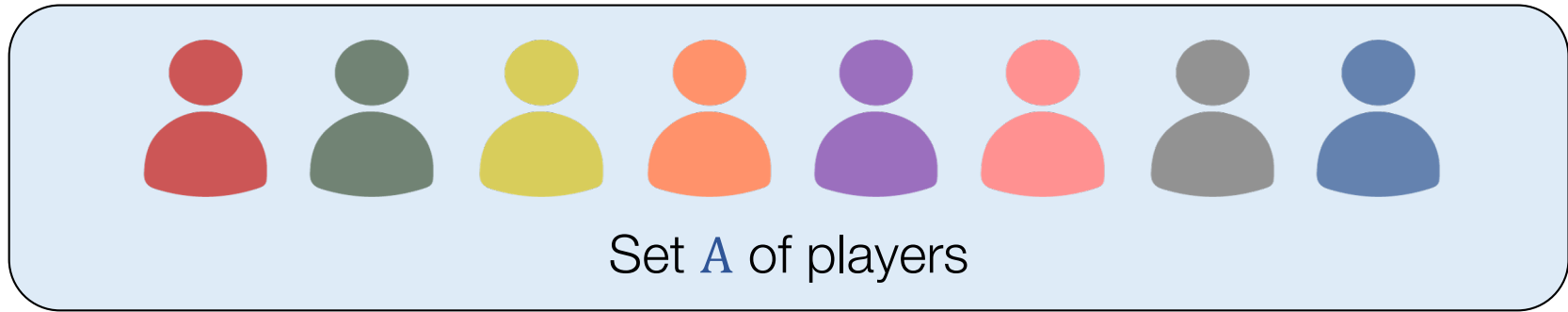
# The Shapley Value

---

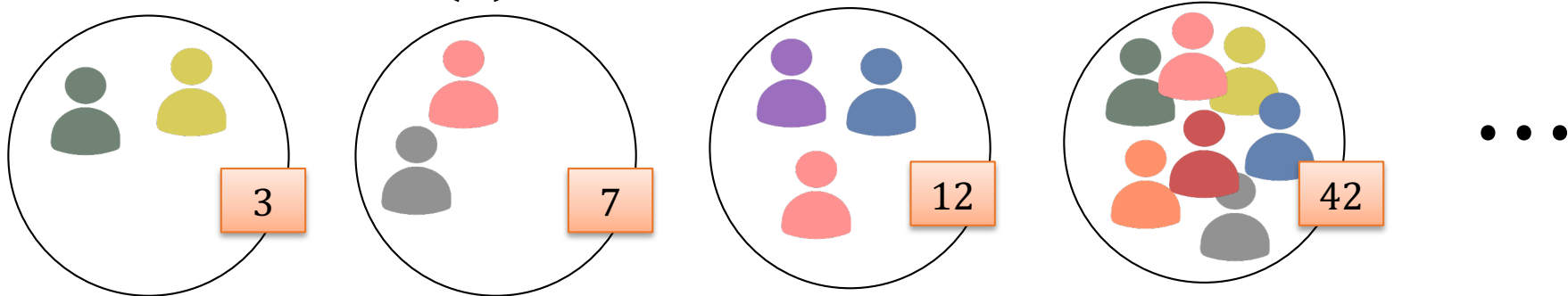
- A widely known profit-sharing formula in cooperative game theory by Shapley
  - [L.S. Shapley: *A value for  $n$ -person games*, 1953] [Roth-88]
- Theoretical justification: **unique modulo rationality desiderata**
- Applied in various areas:
  - Pollution responsibility in environmental management
  - Influence measurement in social network analysis
  - Identifying candidate autism genes
  - Bargaining foundations in economics
  - Takeover corporate rights in law
  - Explanations (local) in machine learning
  - Explanations in databases



# Shapley Definition



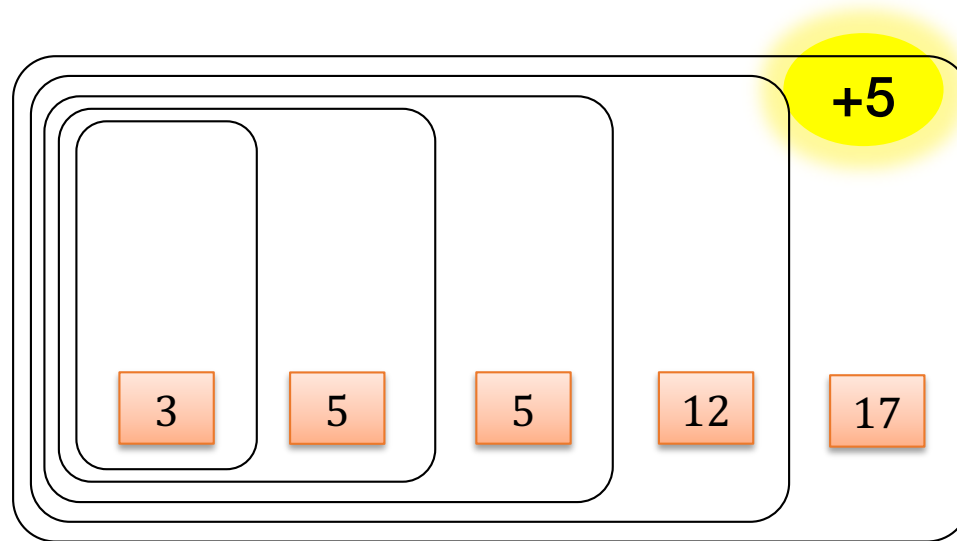
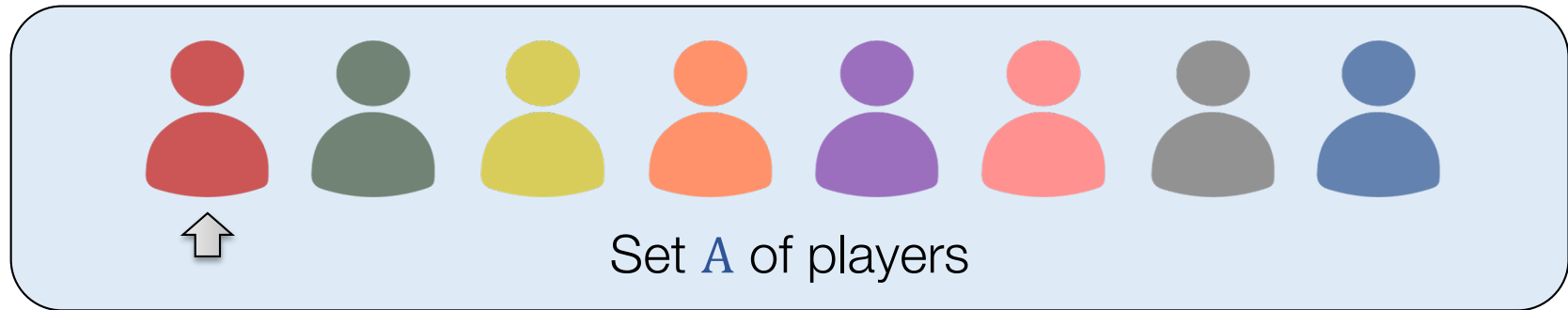
Wealth function  $v: \mathcal{P}(A) \rightarrow \mathbb{R}$



How to share the wealth among the players?

$$\text{Shapley}(A, v, a) = \sum_{B \subseteq A \setminus \{a\}} \frac{|B|! (|A| - |B| - 1)!}{|A|!} (v(B \cup \{a\}) - v(B))$$

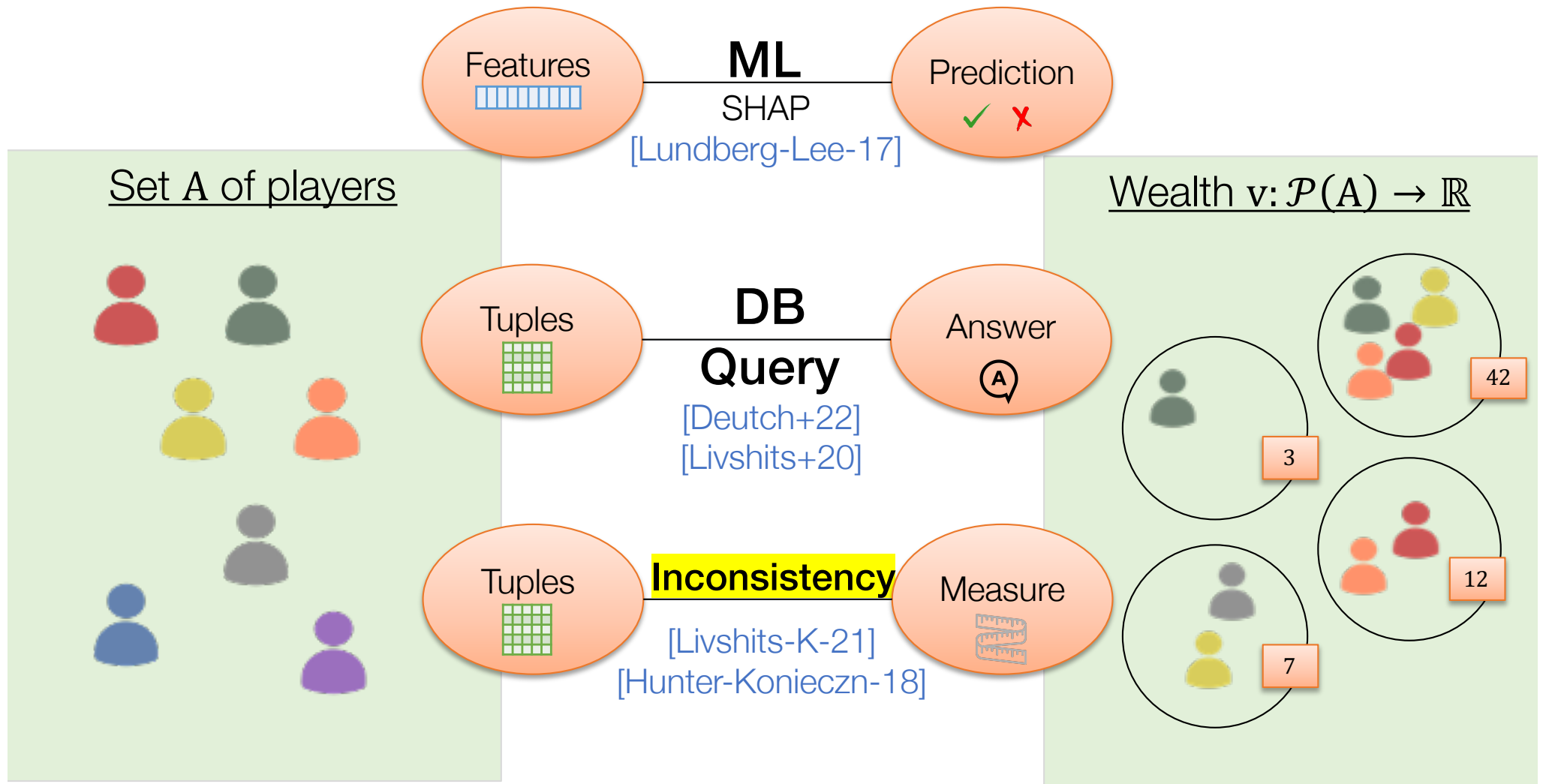
# Shapley Explained



$$\text{Shapley}(A, v, a) = \sum_{B \subseteq A \setminus \{a\}} \frac{|B|! (|A| - |B| - 1)!}{|A|!} (v(B \cup \{a\}) - v(B))$$

Shapley value: expected delta

# Instantiations of the Shapley Value



*How to share wealth among players?*



# Problem: Shapley Calculation

---

Compute a Shapley Value for Inconsistency

---

**Params:** Relation schema  $S$  ; set  $\Sigma$  of constraints ;  
inconsistency measure  $I$

---

**Input:** Relation  $D$  over  $S$  ; tuple  $t$  of  $D$

---

**Goal:** Compute the Shapley value of  $t$  under  $I(\Sigma, D)$

---

# Example 1: Number of Violations

Easily computable coefficients

$$\text{Shapley}(D, \mathbf{t}) = \sum_{k=0}^{|D|-1} c_k \cdot \mathbb{E}[\text{\#violations of } \mathbf{t} \text{ w/ random } k \text{ facts}]$$

Facts in conflict with  $\mathbf{t}$

$$= \sum_{k=0}^{|D|-1} c_k \cdot \underbrace{\mathbb{E}[\text{\#facts from } F \text{ in a random } k\text{-subset}]}_{\text{Simple combinatorics}}$$

# Example 2: Number of Problematic Tuples

$$\text{Shapley}(D, \mathbf{t}) = \sum_{k=0}^{|D|-1} c_k \cdot ( \mathbb{E}[\text{\#problematic in random } k \text{ facts and } \mathbf{t}] - \mathbb{E}[\text{\#problematic in random } k \text{ facts excluding } \mathbf{t}] )$$

⋮

$$\mathbb{E}[\text{\#problematic among random } k \text{ facts}]$$

$$\begin{aligned} &= \mathbb{E} \left[ \sum_{s \in D} 1[s \text{ is selected together with some conflict } s'] \right]. \\ \text{Linearity of expectation} \downarrow &= \sum_{s \in D} \Pr(s \text{ is selected together with some conflict } s') \\ &= \sum_{s \in D} \Pr(s \text{ and at least one of } F_s \text{ are selected}) \end{aligned}$$

Simple combinatorics

# Complexity Picture

Measure	lhs chain	No lhs chain, tractable rep_cost	other
drastic	PTIME	FP <sup>#P</sup> -complete	
#repairs	PTIME	FP <sup>#P</sup> -complete	
repair_cost	PTIME	Open	NP-hard
#violations	PTIME		
#problematic	PTIME		

Next

Discussed

Discussed

# Hardness Technique 1: Measure Hardness

Measures: **#repairs** , **repair\_cost**

- Consider the cooperative game with the set  $A$  of players and utility  $v$
- A general property of the Shapley values is that the sum of values is equal to the overall utility:

$$\sum_{a \in A} \text{Shapley}(A, v, a) = v(A)$$

- Hence, from the Shapley values of facts we can compute the inconsistency measure over the whole database
- **Conclusion 1:** If  $\Sigma$  is not an lhs-chain (u.t.e.), then Shapley value is #P-hard for #repairs
- **Conclusion 2:** If  $\Sigma$  is not emptied by the simplification of Livshits+, then Shapley value is NP-hard for repair\_cost

# Complexity Picture

Measure	lhs chain	No lhs chain, tractable rep_cost	other
drastic	PTIME	FP <sup>#P</sup> -complete <span>Next</span>	
#repairs	PTIME	FP <sup>#P</sup> -complete <span>Discussed</span>	
repair_cost	PTIME	Open	NP-hard
#violations	PTIME		
#problematic	PTIME		

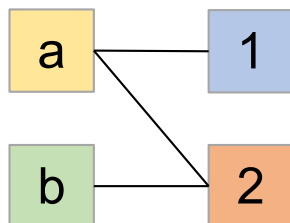
Discussed

Discussed

# Hardness Technique 2: Linear Algebra (1)

$$\Sigma = \{A \rightarrow B, B \rightarrow A\}$$

A	B
a	1
b	2
a	2

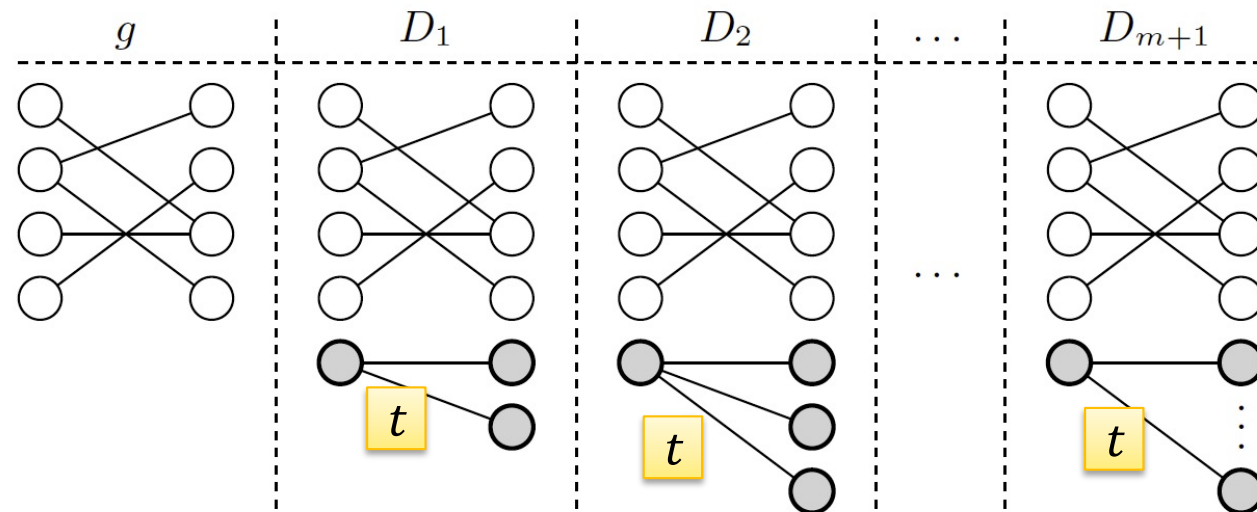


Consistency =  
being a partial  
matching

Measure: **drastic** (1 / 0)

$$\text{Shapley}(D_i, \mathbf{t}) \approx \sum_{k=0}^m |M(g, k)| \cdot r \cdot (k+1)! \cdot (m+r-k-1)!$$

Partial  
matchings



# Hardness Technique 2: Linear Algebra (2)

Measure: **drastic** (1 / 0)

$$\text{Shapley}(D_i, \mathbf{t}) \approx \sum_{k=0}^m |M(g, k)| \cdot r \cdot (k+1)! \cdot (m+r-k-1)!$$

$$\begin{pmatrix}
 1 \cdot 1!m! & 1 \cdot 2!(m-1)! & \dots & 1 \cdot (m+1)!0! \\
 2 \cdot 1!(m+1)! & 2 \cdot 2!m! & \dots & 2 \cdot (m+1)!1! \\
 \vdots & \vdots & \vdots & \vdots \\
 (m+1) \cdot 1!2m! & (m+1) \cdot 2!(m-1)! & \dots & (m+1) \cdot (m+1)!m!
 \end{pmatrix}
 \begin{bmatrix}
 |M(g, 0)| \\
 |M(g, 1)| \\
 \vdots \\
 |M(g, m)|
 \end{bmatrix}
 =
 \begin{bmatrix}
 \text{Shapley}(D_1, \mathbf{t}) \\
 \text{Shapley}(D_2, \mathbf{t}) \\
 \vdots \\
 \text{Shapley}(D_{m+1}, \mathbf{t})
 \end{bmatrix}$$

Assume PTime

We can compute #partial-matchings of a bipartite graph... #P-complete!

$$\begin{pmatrix}
 0! & 1! & \dots & m! \\
 1! & 2! & \dots & (m+1)! \\
 \vdots & \vdots & \vdots & \vdots \\
 m! & (m+1)! & \dots & 2m!
 \end{pmatrix}$$

Non-singular [Bacher-02]



# Hardness Technique 3: Fact-Wise Reduction

---

Measure: **drastic** (1 / 0)

- We showed hardness for  $\{A \rightarrow B, B \rightarrow A\}$
- We need to show hardness for every set of FDs that is *not* an lhs-chain
- But this, we get for free, since...
- For the hardness of #repairs, we already showed fact-wise reductions from  $\{A \rightarrow B, B \rightarrow A\}$

# + Approximation

Measure	lhs chain	No lhs chain, tractable rep_cost	other
drastic	PTIME	FP <sup>#P</sup> -complete	
<i>approx</i>		FPRAS	
#repairs	PTIME	FP <sup>#P</sup> -complete	
<i>approx</i>		Open	
repair_cost	PTIME	Open	NP-hard
<i>approx</i>		FPRAS	No FPRAS
#violations	PTIME		
#problematic	PTIME		

Would imply an FPRAS for #MIS in a bipartite graph – long standing open problem

# Approximation Algorithms

Measures: **drastic** ; **cardinality**

- In the case of drastic and cardinality, a tuple can increase the measure by either 1 or 0
  - In the sampling-w/o-replacement trial
- Hence, the Shapley value of a fact is **the probability that it increases the measure**
- An additive approx. is straightforward: average over multiple trials
- The additive approx. gives a **multiplicative** approx. via the **gap property** that holds here:

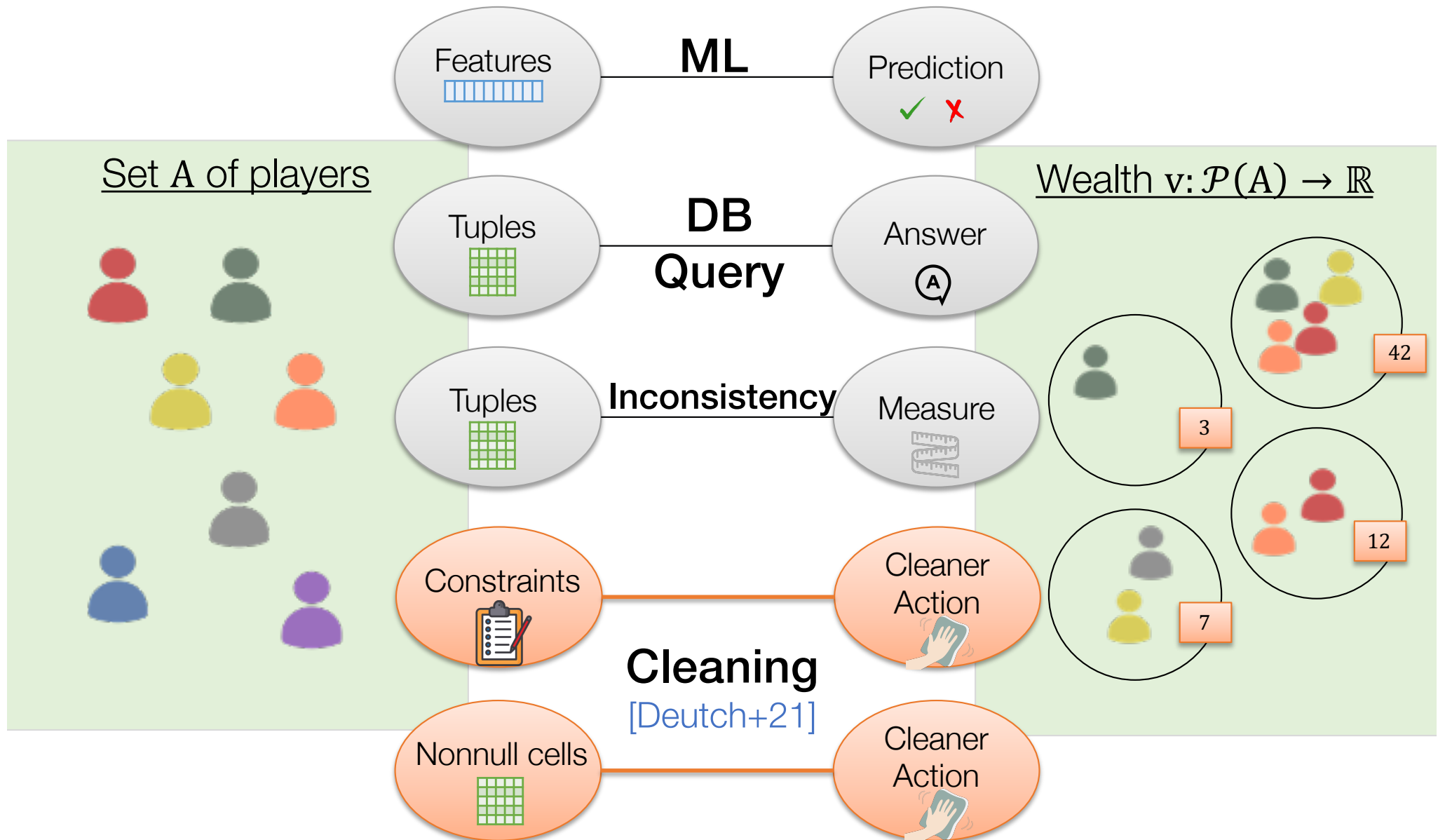
*If Shapley is nonzero, it is at least  $1/\text{poly}$*

# Explanation of Cleaning Algorithms

---

- Deutch et al. studied explanations for the actions of **black-box tools for data cleaning**
  - [\[Deutch-Frost-Gilad-Sheffer-ClKM21\]](#)
- Specifically,
  - *Why has this cell changed?*
  - *Which components are most responsible to the the cell value produced by the cleaner?*
- Two types of components:
  - Constraints (DCs)
  - Cell values (non-null)

# Instantiations of the Shapley Value



# Outline

1. Introduction & Background
2. Inconsistency Measures
3. Complexity of Calculation
4. Probabilistic Database Viewpoint
5. Responsibility Attribution
- ▶ 6. Concluding Remarks

we are here

# Summary & Conclusions

---

- Quantitative measurement of inconsistency arise in various situations in database management
  - Classic/recent, implicit/explicit
- We discussed 3 use cases: notions of **soft constraints**, **progress indication**, attribution of **responsibility** to noise
- Interesting computational challenges, good understanding of complexity in limited settings
  - Functional dependencies (sometimes denial constraints) and tuples deletions
- Connections to probabilistic databases
  - Fundamental problems coincide, unified models studied

# Some Open Problems (1)

---

- Empirical user studies on how measurements help quality management / data prep
  - “Data preparation accounts for about 80% of the work of data scientists” - Forbes
- Repairing model – measures heavily based on **tuple deletion**
  - Insufficient theory about **cell updates**
  - For example, the complexity of repair-cost?
  - What would be good measures?



# Some Open Problems (2)

---

- Beyond **anti-monotonic** – what about foreign keys? Inclusion constraints?
  - Then, we should also consider **tuple addition**
- **Shapley values** – we lack approximation algorithms and practical techniques
  - Approximation for Shapley value for repair-cost
  - Knowledge compilation? (e.g., via provenance tracking using ProvSQL as in DB queries [\[Deutch+22\]](#) )
- **Soft constraints** – we know the complexity of very few cases, basic problems still open

**Thank you!**

# Main References (1)

---

- Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin: Network flows - theory, algorithms and applications. Prentice Hall 1993, ISBN 978-0-13-617549-0, pp. I-XV, 1-846
- Marcelo Arenas, Leopoldo E. Bertossi, Jan Chomicki: Consistent Query Answers in Inconsistent Databases. PODS 1999: 68-79
- Leopoldo E. Bertossi: Measuring and Computing Database Inconsistency via Repairs. SUM 2018: 368-372
- Marco Calautti, Georg Gottlob, Andreas Pieris: Non-Uniformly Terminating Chase: Size and Complexity. PODS 2022: 369-378
- Christopher De Sa, Ihab F. Ilyas, Benny Kimelfeld, Christopher Ré, Theodoros Rekatsinas: A Formal Framework for Probabilistic Unclean Databases. ICDT 2019: 6:1-6:18
- Daniel Deutch, Nave Frost, Amir Gilad, Oren Sheffer: Explanations for Data Repair Through Shapley Values. CIKM 2021: 362-371
- Daniel Deutch, Nave Frost, Benny Kimelfeld, Mikaël Monet: Computing the Shapley Value of Facts in Query Answering. SIGMOD Conference 2022: 1570-1583
- Wenfei Fan, Floris Geerts: Foundations of Data Quality Management. Synthesis Lectures on Data Management, Morgan & Claypool Publishers 2012

# Main References (2)

---

- John Grant, Anthony Hunter: Measuring the Good and the Bad in Inconsistent Information. IJCAI 2011: 2632-2637
- Eric Gribkoff, Guy Van den Broeck, and Dan Suciu. 2014. The Most Probable Database Problem. In BUDA.
- Jyrki Kivinen, Heikki Mannila: Approximate Inference of Functional Dependencies from Relations. Theor. Comput. Sci. 149(1): 129-149 (1995)
- Ester Livshits, Alireza Heidari, Ihab F. Ilyas, Benny Kimelfeld: Approximate Denial Constraints. Proc. VLDB Endow. 13(10): 1682-1695 (2020)
- Ester Livshits, Leopoldo E. Bertossi, Benny Kimelfeld, Moshe Sebag: The Shapley Value of Tuples in Query Answering. Log. Methods Comput. Sci. 17(3) (2021)
- Ester Livshits, Benny Kimelfeld: The Shapley Value of Inconsistency Measures for Functional Dependencies. ICDT 2021: 15:1-15:19
- Ester Livshits, Benny Kimelfeld, Sudeepa Roy: Computing Optimal Repairs for Functional Dependencies. ACM Trans. Database Syst. 45(1): 4:1-4:46 (2020)
- Ester Livshits, Benny Kimelfeld, Jef Wijsen: Counting subset repairs with functional dependencies. J. Comput. Syst. Sci. 117: 154-164 (2021)

# Main References (3)

---

- Ester Livshits, Rina Kochirgan, Segev Tsur, Ihab F. Ilyas, Benny Kimelfeld, Sudeepa Roy: Properties of Inconsistency Measures for Databases. SIGMOD Conference 2021: 1182-1194
- Dongjing Miao, Zhipeng Cai, Jianzhong Li, Xiangyu Gao, Xianmin Liu: The Computation of Optimal Subset Repairs. Proc. VLDB Endow. 13(11): 2061-2074 (2020)
- Roth, A. E. 1988. The Shapley value: essays in honor of Lloyd S. Shapley. Cambridge University Press.
- Shapley, L. S. 1953. A value for n-person games. Contributions to the Theory of Games 2(28): 307–317.