

Quantitative Reasoning about Dependency Violation in Databases

Benny Kimelfeld

Joint work with **Ester Livshits**

Examples of Inconsistency (DBPedia)



Marion Jones

dbo:height

- 1.524
- 1.778



Cullen Douglas

dbo:birthPlace

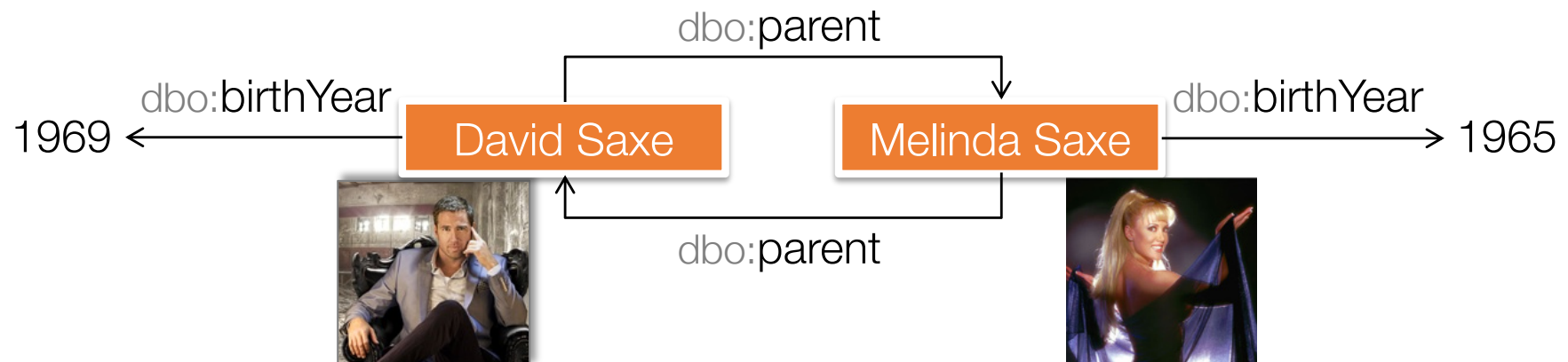
- dbr:California
- dbr:Florida



Irene Tedrow

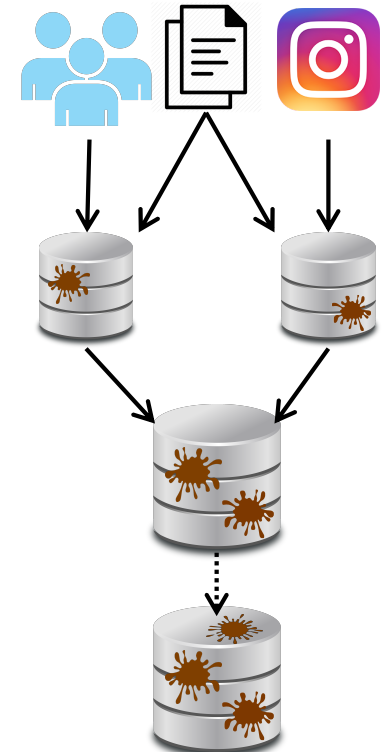
dbo:deathPlace

- dbr:California
- dbr:Hollywood,_Los_Angeles
- dbr:New_York_City



Sources of Inconsistent Data

- Imprecise **data sources**
 - Crowd, Web pages, social encyclopedias, sensors, ...
- Imprecise **data generation**
 - ETL, 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 ...*



Principled Declarative Approaches

- Several principled approaches proposed for reasoning about inconsistent data
- Concepts in declarative approaches
 - Integrity constraints (dependencies)
 - Or *dependencies*
 - Inconsistent database
 - Violates the constraints
 - Edit operations
 - Delete/insert tuple, update an attribute
 - Repairs
 - Consistent DB following a *legitimate* edit
- Theoretical formulation [Arenas,Bertossi,Chomicki 99]

Examples of Integrity Constraints

- Key constraints
 - `Person(ssn,name,birthCity,birthState)`
- Functional Dependencies (FDs)
 - `birthCity → birthState`
- Conditional FDs
 - `birthCity → birthState whenever country="USA"`
- Denial constraints
 - `not[Parent(x,y) & Parent(y,x)]`
- Referential (foreign-key) constraints
 - `Parent(x,y) → Person(x) & Person(y)`
- ...

Examples of 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 (& subset) repair

Classic Repair Problems

- Repairing / Cleaning
 - Compute a (good/best) repair
 - [Bertossi+ 08] [Kolahi,Lakshmanan 09] [Livshits,K,Roy 18]
- Consistent Query Answering (CQA)
 - *Which query answers are not affected by inconsistency?*
 - Formally, find the tuples that belong to $Q(J)$ for all repairs J
 - [Arenas+ 99] [Fuxman,Miller 05] [Koutris,Wijzen 17]
- Repair checking
 - Given I and J , is J a repair of I ? ; typically a complexity tool
 - [Afrati,Kolaitis 09] [Chomicki,Marcinkowski 05]
- Repair counting (& enumeration)
 - Measure consistency of query answers [Maslowski,Wijzen 14]
 - Measure inconsistency of data [Livshits,K 17] [Livshits+ 21] ; also, in the KR community [DeBona,Grant,Hunter,Konieczny 18]

Inconsistency Measure

- Idea: *quantify the extent to which integrity constraints are violated*
- Several reasons:
 - Given a new data source, how reliable is it?
 - Progress bar for data cleaning
 - [Livshits, Kochirgan, Tsur, Ilyas, K, Roy: *Properties of Inconsistency Measures for Databases*, SIGMOD 2021]
 - Which tuples are mostly responsible for inconsistency?
 - [Livshits, K: *The Shapley Value of Inconsistency Measures for Functional Dependencies*. ICDT 2021]
- Studied in KR community [Grant, Hunter, ...], recently in the DB community [Bertossi, ...]

Basic Inconsistency Measures

Complexity?

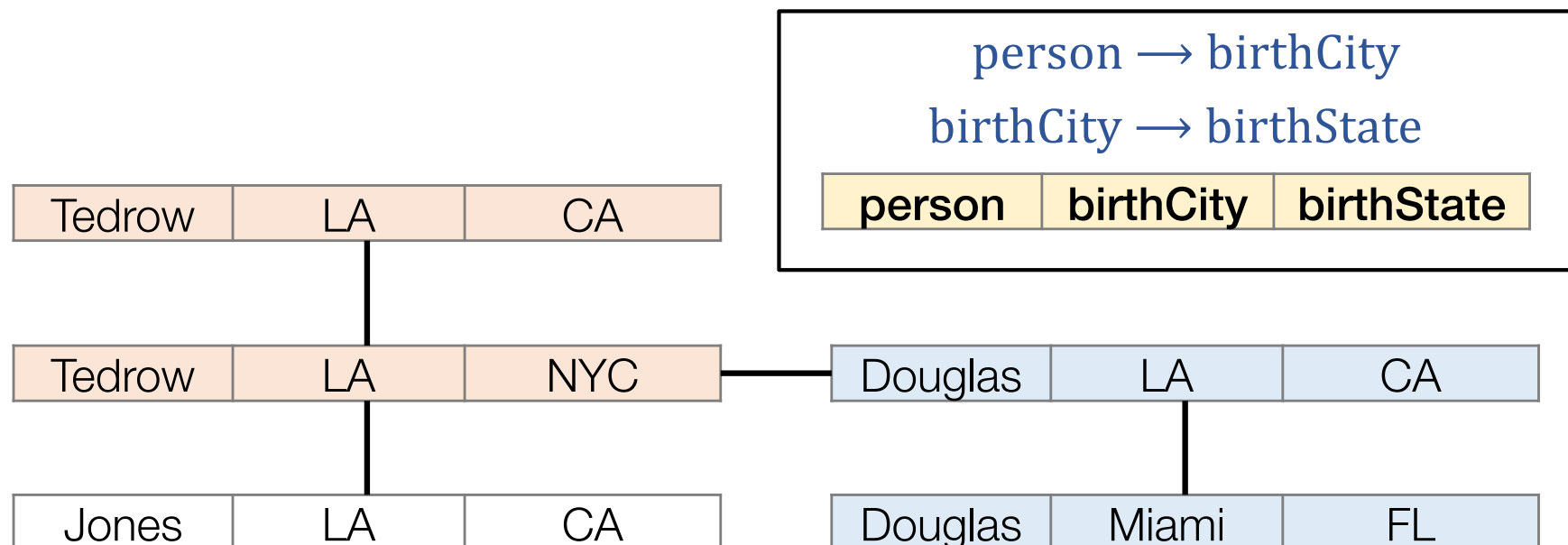
- **Drastic**: 1 or 0 (inconsistent or consistent)
 - [Thimm 2017]
- **#violations** (i.e., minimal inconsistent subsets)
 - [Hunter & Konieczny 2008]
- **#problematic** tuples (i.e., tuples in violations)
 - [Grant & Hunter 2011]
- **#repairs**: number of maximal consistent subsets
 - [Grant & Hunter 2011]
- Minimal #tuples to delete to attain consistency (**cardinality repair**)
 - [Grant and Hunter 2013], [Bertossi 2018]

Outline

1. Inconsistency Measures via Repairs
2. Repair Counting **We are here**
3. Repair Optimization
4. Responsibility to Inconsistency

Repair Counting as MIS Counting

- For FDs, a repair is a **Maximal Independent Set (MIS)** of the **conflict graph** of the database
 - Tuples \Rightarrow nodes, violations \Rightarrow edges
- Hence, repair counting amounts to MIS counting
 - Over conflict 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?*

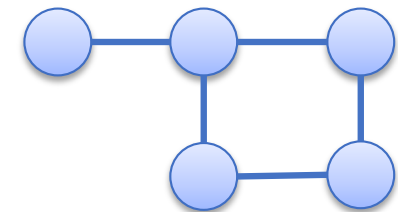
THEOREM [Livshits,K,Wijzen 2021 (JCSS)]

Equivalent for every fixed set of FDs:

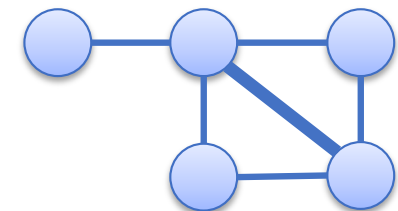
1. Repairs can be counted in **poly. time**
2. Every conflict graph is **P_4 -free**

Tractability testable in poly. time (given FDs)

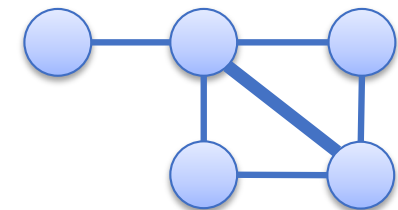
* Assuming $P \neq \#P$



Not P_4 -free



P_4 -free



Hard

Poly time

ssn \rightarrow city
city \rightarrow state

ssn \rightarrow name
ssn country \rightarrow license#

Approx. open...

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

faculty \rightarrow dean
faculty professor \rightarrow room#

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

Outline

1. Inconsistency Measures via Repairs
2. Repair Counting
3. Repair Optimization
4. Responsibility to Inconsistency

We are here

Detour to Probabilistic Repairing

Probabilistic Duplicates [Andritsos-Fuxman-Miller06]

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** probabilistic **Databases** (BID)
[Dalvi-Ré-Suciu11]

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

Constrained TID [Gribkoff-VanDenBroeck-Suciu14]

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

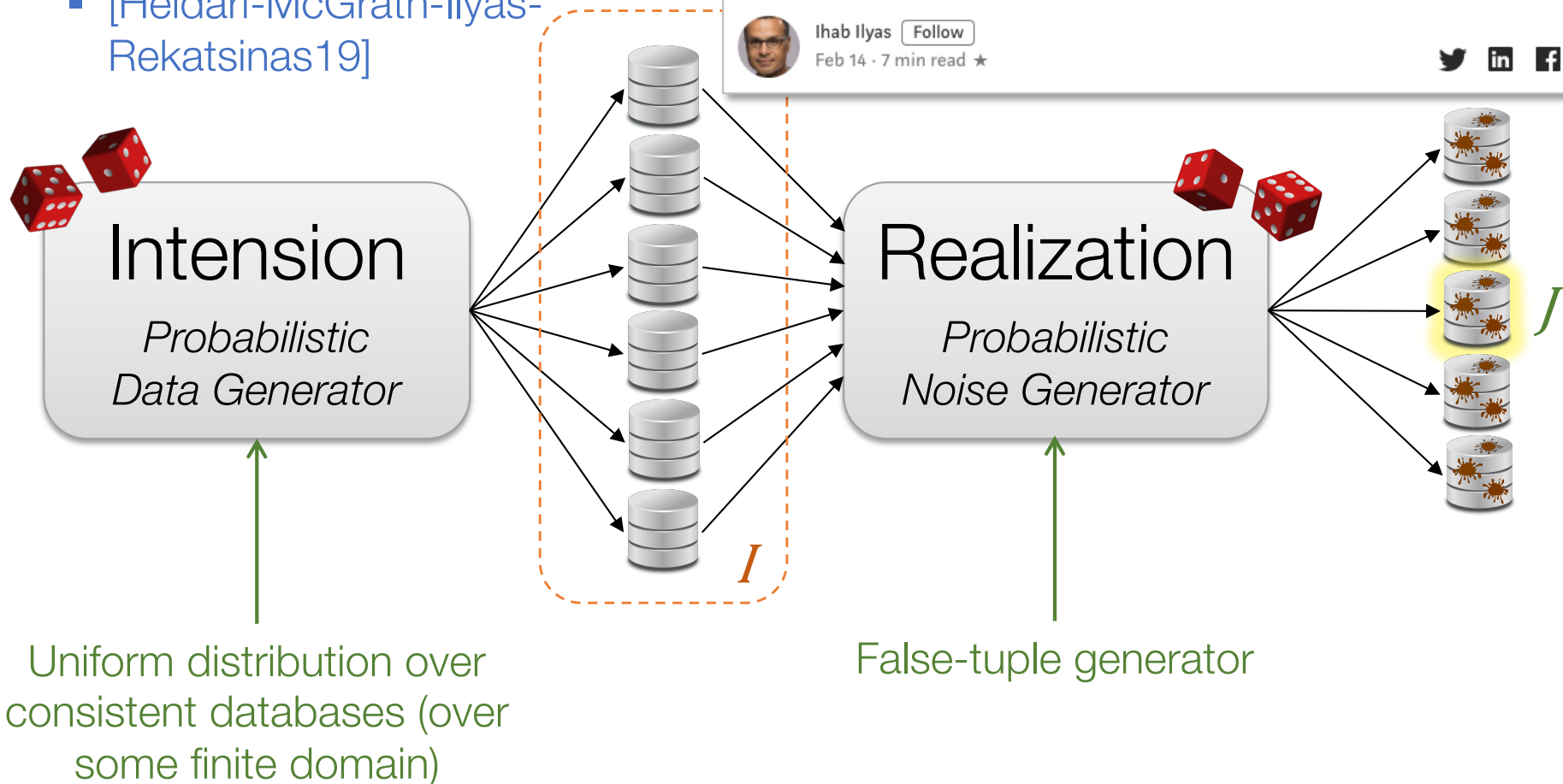
$$p(W) = \Pr(W \mid C)$$

Computational problem: find a most probable W (**MPD**)

Special Case of the Prob. Unclean DB (PUD)

[DeSa-Ilyas-K-Ré-Rekatsinas18]

- HoloClean
 - [Rekatsinas-Chu-Ilyas-Ré17]
- HoloDetect
 - [Heidari-McGrath-Ilyas-Rekatsinas19]



MPD

To solve this problem, we need to understand how to find a **cardinality repair** (largest consistent subset)

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?

$$\max_{\text{consistent } J} \left(\prod_{t \in J} p(t) \times \prod_{t \notin J} (1 - p(t)) \right)$$

... Back to Repair Optimization

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 FD

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



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

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

Repeated Simplification

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



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



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



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



$$\{\}$$

THEOREM [Livshits-K-Roy2018]

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

1. A **cardinality repair** can be found in poly-time.
2. An **MPD** can be found in poly-time.
3. The FD set can be **simplified until emptied**.

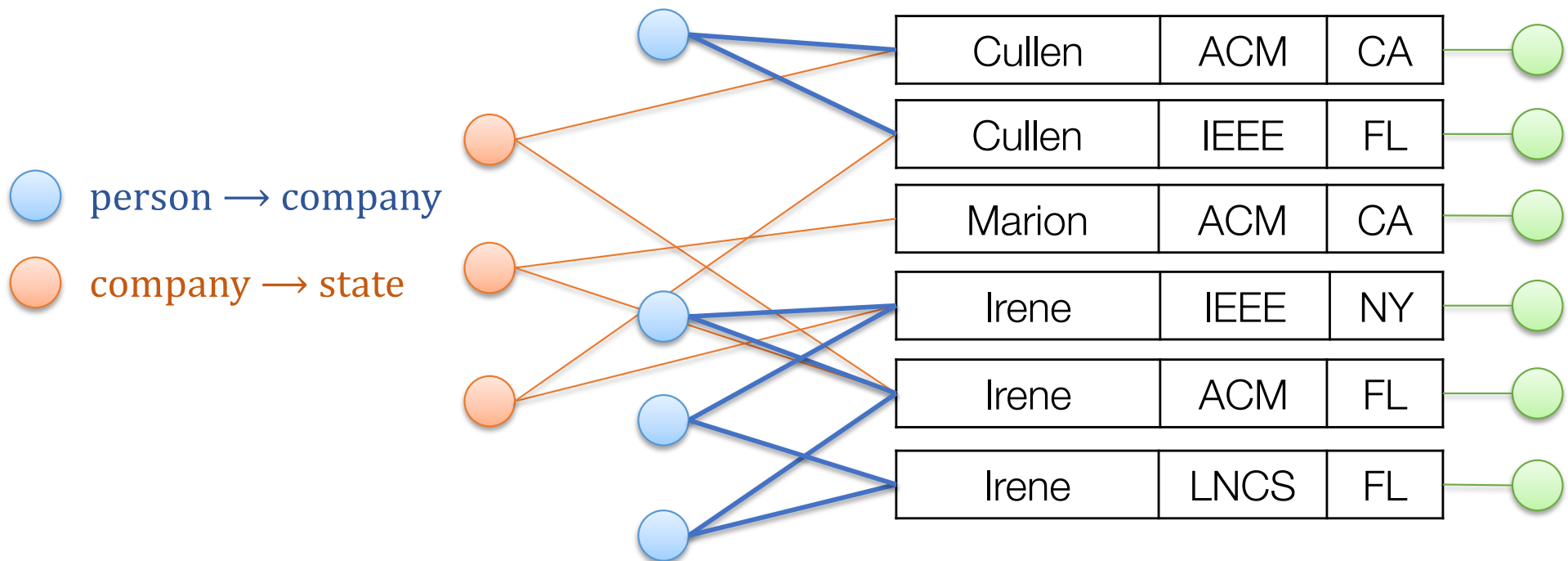
Generalization to *soft constraints*

[Carmeli-Grohe-K-Livshits-Tibi21]

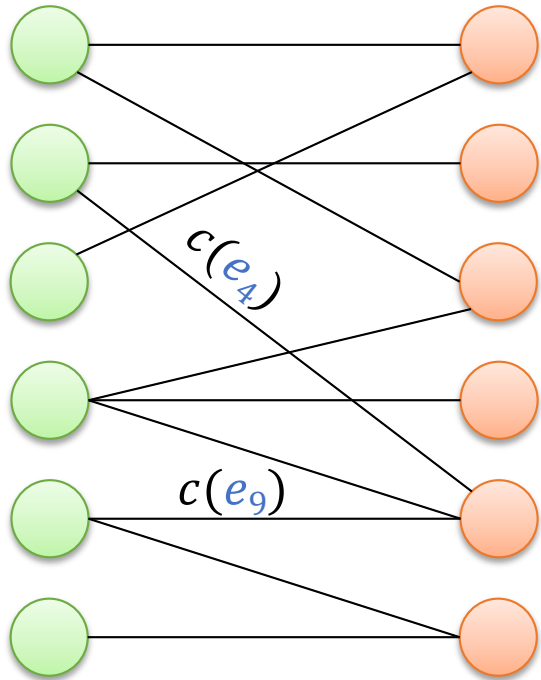
MPD for Weak Constraints

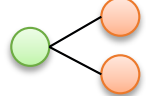
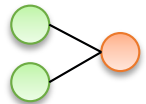
MPD: $\max_{\text{consistent } J} \left(\prod_{t \in J} p(t) \times \prod_{t \notin J} (1 - p(t)) \right)$

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



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

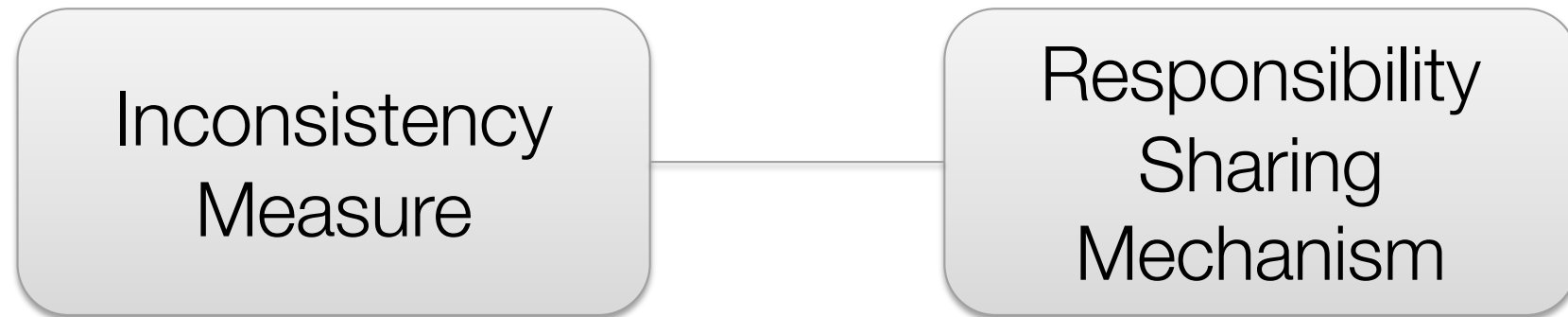
Algorithm via *minimum-cost maximum flow*

[Carmeli-Grohe-K-Livshits-Tibi21]

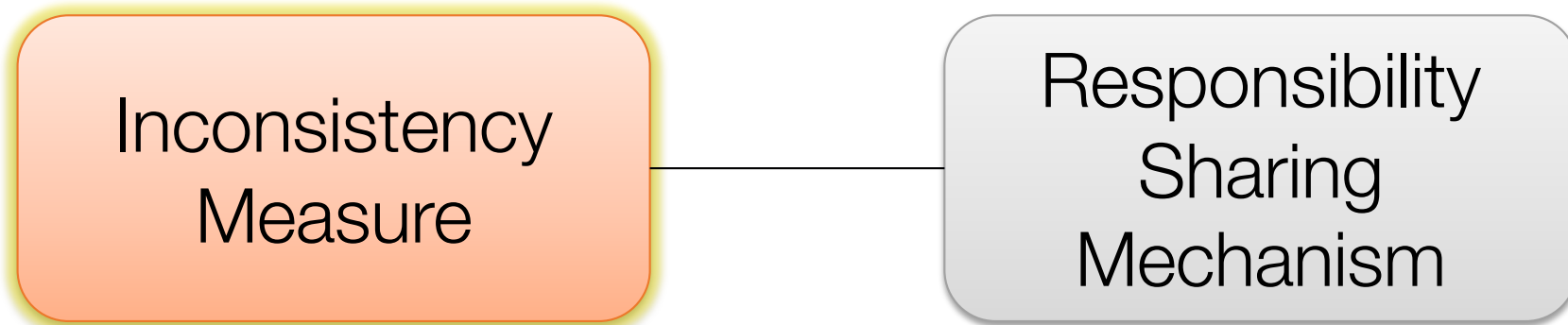
Outline

1. Inconsistency Measures via Repairs
2. Repair Counting
3. Repair Optimization
4. Responsibility to Inconsistency **We are here**

Responsibility Attribution Requires 2 Parts



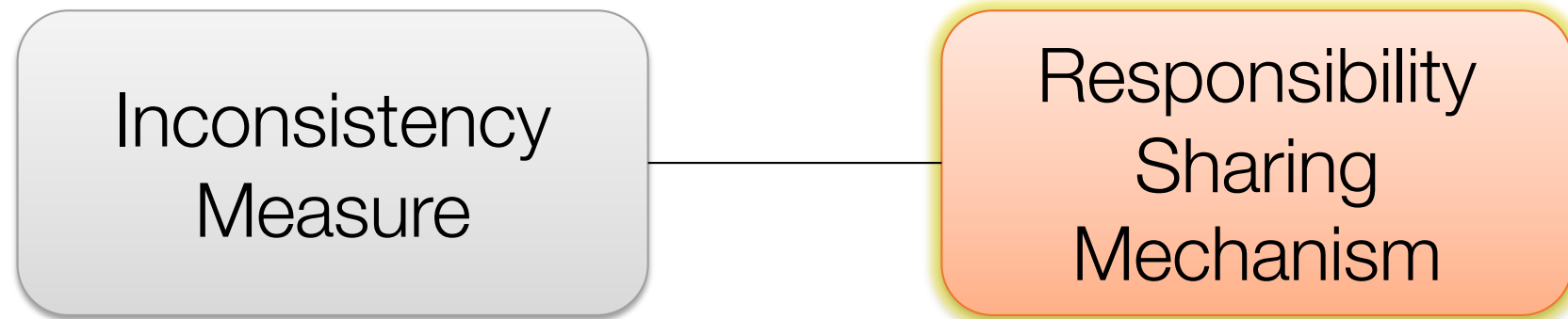
Responsibility Attribution Requires 2 Parts



Basic Inconsistency Measures

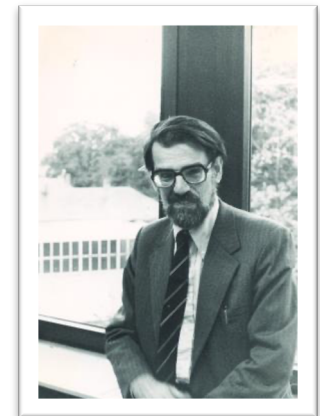
- **Drastic**: 1 or 0 (inconsistent or consistent)
 - [Thimm 2017]
- **#violations** (i.e., minimal inconsistent subsets)
 - [Hunter & Konieczny 2008]
- **#problematic** tuples (i.e., tuples in violations)
 - [Grant & Hunter 2011]
- **#repairs**: number of maximal consistent subsets
 - [Grant & Hunter 2011]
- Minimal #tuples to delete to attain consistency (**cardinality repair**)
 - [Grant and Hunter 2013], [Bertossi 2018]

Responsibility Attribution Requires 2 Parts

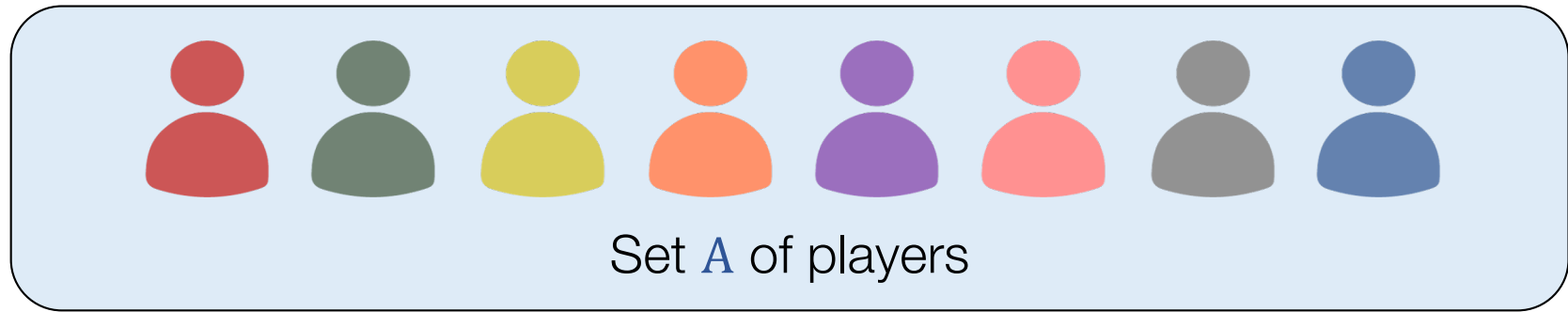


The Shapley Value

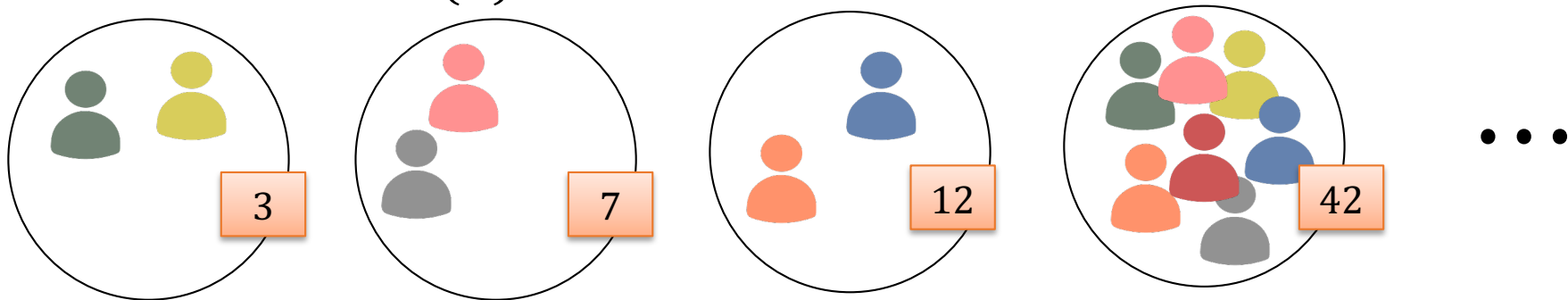
- A widely known profit-sharing formula in cooperative game theory by Shapley
 - [L.S. Shapley: *Stochastic Games*, 1953]
- Theoretical justification: **unique modulo rationality desiderata**
- Applied in various areas:
 - Pollution responsibility in environmental management
 - Influence measurement in social network analysis
 - Identifying candidate autism genes
 - Bargaining foundations in economics
 - Takeover corporate rights in law
 - Local explanations in machine learning
 - Answer explanation for DB queries



Shapley Definition



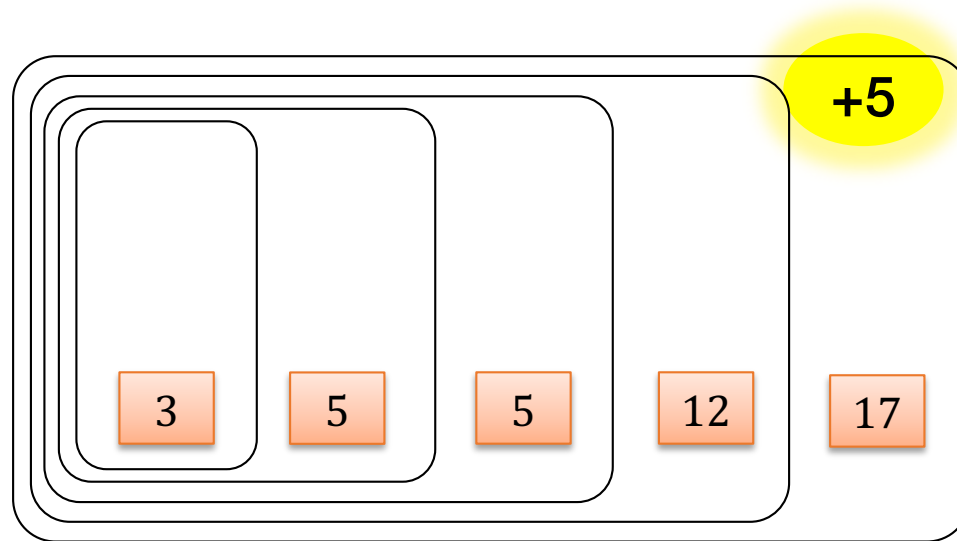
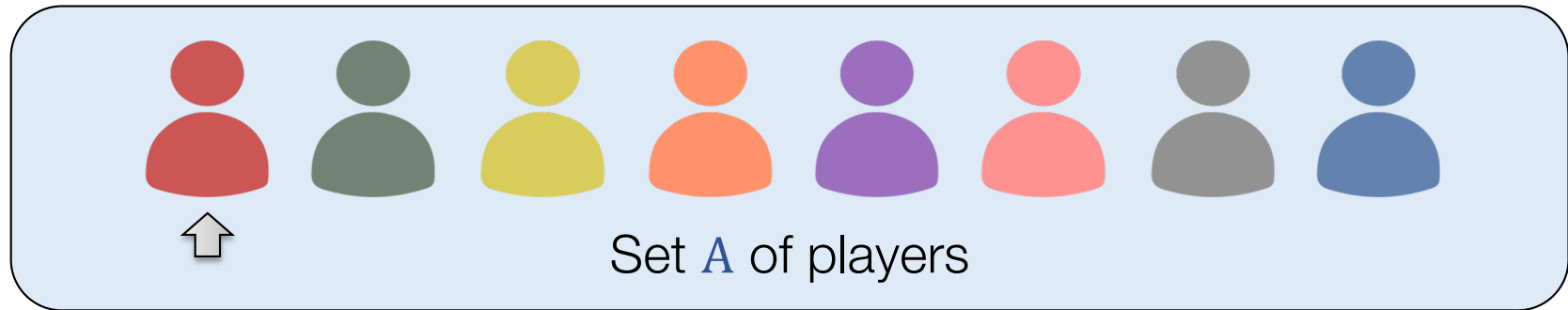
Wealth function $v: \mathcal{P}(A) \rightarrow \mathbb{R}$



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

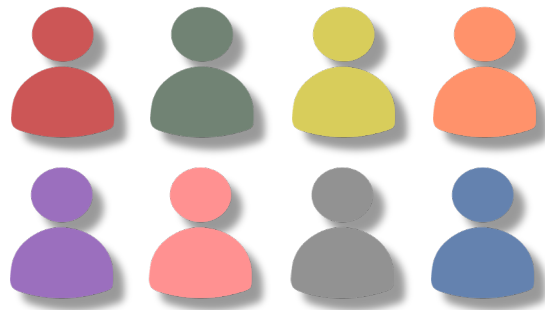


$$\text{Shapley}(A, v, a) = \sum_{B \subseteq A \setminus \{a\}} \frac{|B|! (|A| - |B| - 1)!}{|A|!} (v(B \cup \{a\}) - v(B))$$

Shapley value: expected delta

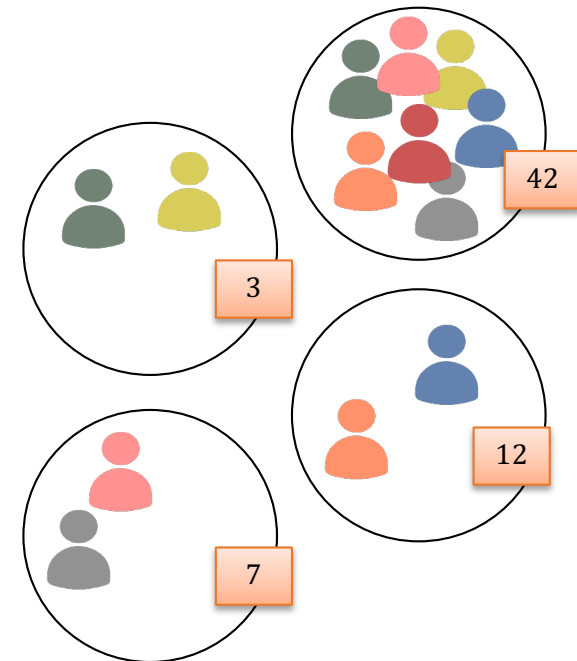
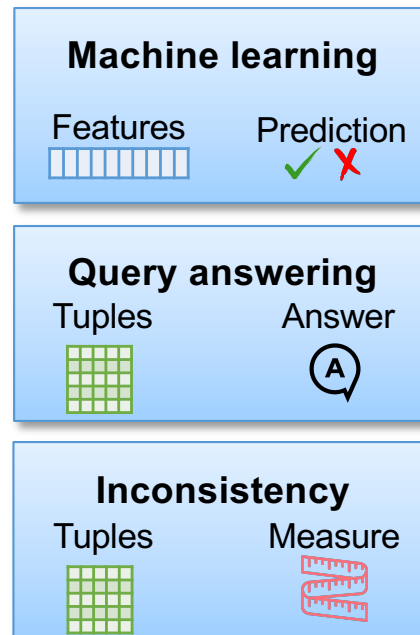
Instantiations of the Shapley Value

Set A of players



How to share the wealth
among the players?

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



Computational Complexity

Measure	lhs chain	No lhs chain, tractable c-repair	other
drastic	PTIME	FP ^{#P} -complete	
#repairs	PTIME	FP ^{#P} -complete	
card. repair	PTIME	Open	NP-hard
#violations	PTIME		
#problematic	PTIME		

Computational Complexity + Approximation

Measure	lhs chain	No lhs chain, tractable c-repair	other
drastic	PTIME	FP ^{#P} -complete	
<i>approx</i>		FPRAS	
#repairs	PTIME	FP ^{#P} -complete	
<i>approx</i>		Open	
card. repair	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

Concluding Remarks

- Various ways of measuring inconsistency amount to combinatorial problems over database repairs
- With inconsistency measures, we can attribute responsibility to inconsistency via mechanisms from cooperative game theory (e.g., Shapley, Banzhaf)
- We have a detailed picture of the computational complexity for FDs
- Largely open: other types of constraints, soft constraints, update operations (not just delete)

Thank you!