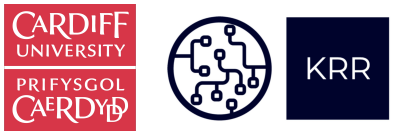
Advances in Logic-Based Entity Resolution: Enhancing ASPEn with Local Merges and Optimality Criteria

Zhiliang (Leon) Xiang, Meghyn Bienvenu, Gianluca Cima,
Víctor Gutiérrez Basulto, Yazmín Ibáñez-García



Cardiff Knowledge Representation & Reasoning (KRR) Group

the LaBRI Research Lab

Sapienza University of Rome

Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	John Smith	32	12345-6	aspirin

Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	John Smith	32	12345-6	aspirin

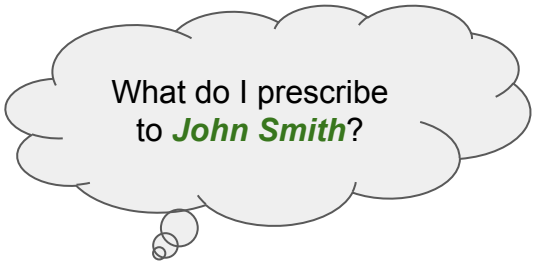
Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	**John Smith**	32	12345-6	aspirin



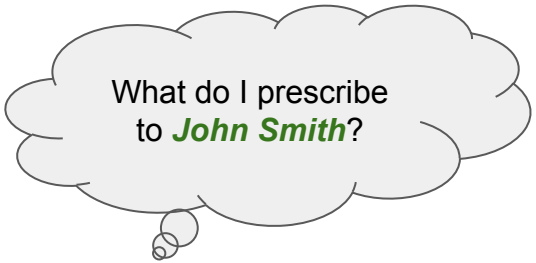
Entity Resolution (ER)

Entity resolution (*ironically goes with different names: deduplication, record linkage, entity matching…*)

- to identify **different constants** representing **the same entity**
- usually in structured/semi-structured data sources, e.g. **database**, knowledge graph

Patient

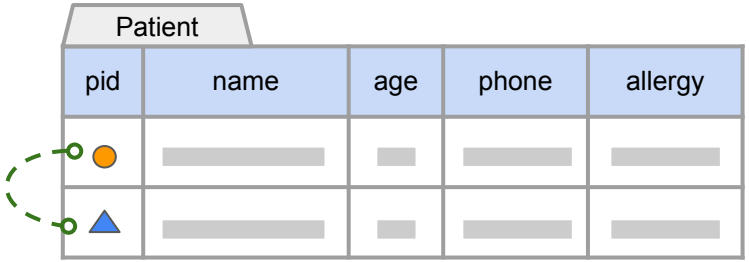
pid	name	age	phone	allergy
p1	J.Smith	32	123456	brufen
p2	John Smith	32	12345-6	aspirin



Critical to **data quality** and **decision making**!

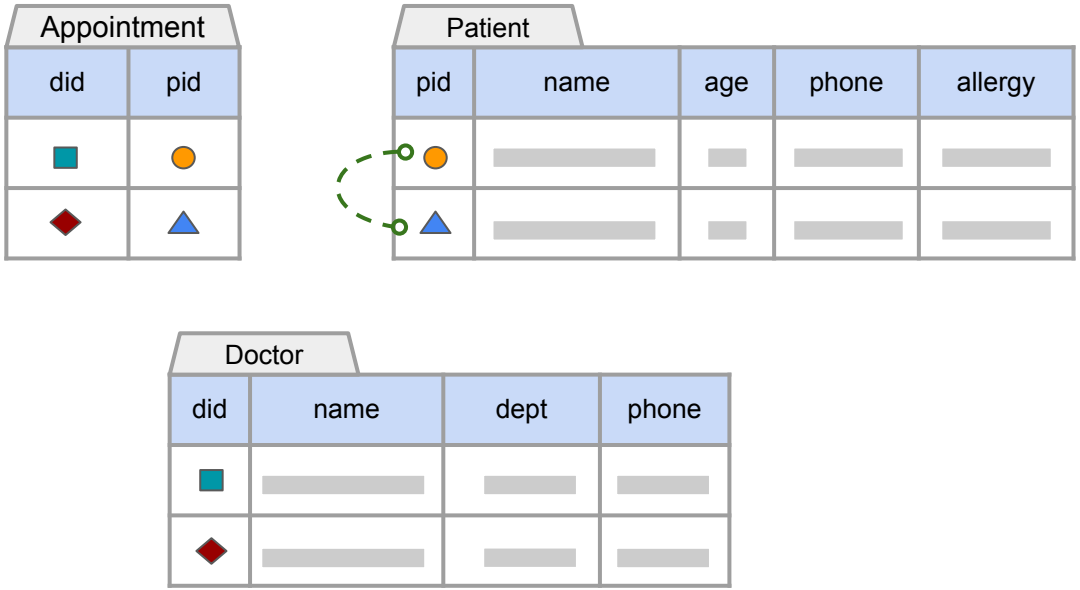
Entity Resolution (ER)

Traditional: within **single-table** or **table pair** (same entity type)



Entity Resolution (ER)

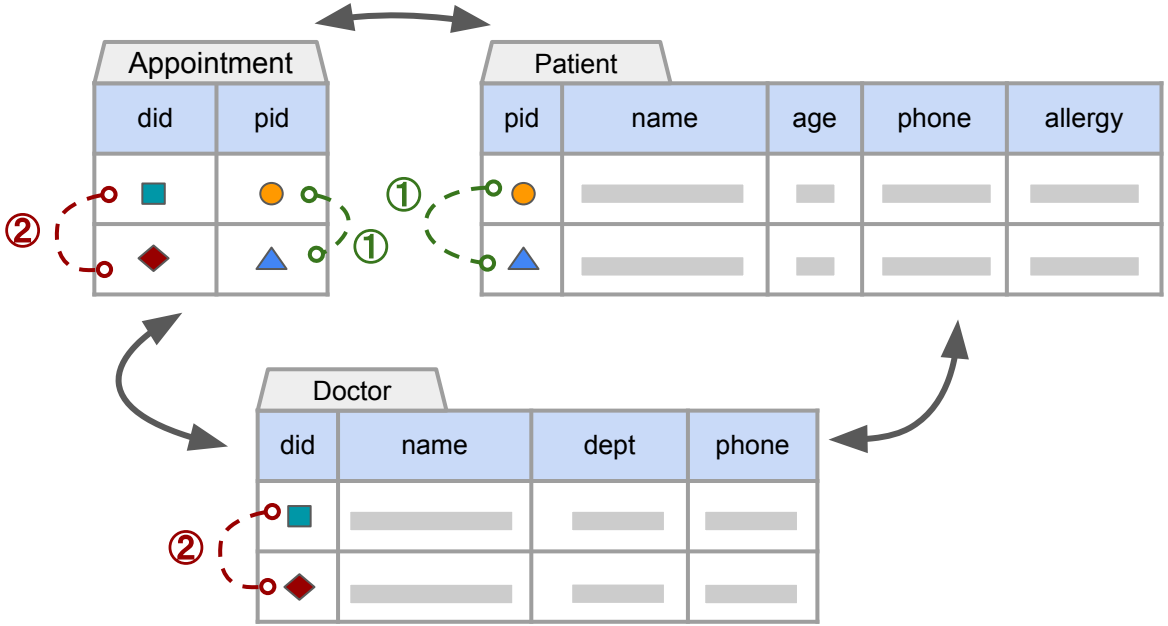
Traditional: within **single-table** or **table pair** (same entity type)



Entity Resolution (ER)

Traditional: within **single-table** or **table pair** (same entity type)

Collective: across **multiple tables**, exploit **inter-dependencies** between tables



Previous work

Many approaches of different foundations: **ML**, **probabilistic**, **rule**

Theoretical framework:

- ***LACE (A Logical Approach to Collective Entity Resolution)*** *[Bienvenu et al. PODS 2022,KR 2023]*

ASP-based Implementation:

- ***ASPEn system*** *[Xiang et al. KR 2024, KR2025]*

Key features:

- **Declarative**, **collective**
- Computes **optimal** ER solutions, **explanations**, brave and (approximate) cautious merge sets etc.
- **Promising performances** on real-life datasets

Previous work

Issues of *ASPEn:*

- Semantics is **global**: merged constants are treated the same **everywhere**

Previous work

Issues of *ASPEn:*

- Semantics is **global**: merged constants are treated the same **everywhere**
  - **Cannot** capture **local resolution**, e.g. some *J. Smith* are *John Smith*, some are *Jerry Smith*

Previous work

Issues of *ASPEn:*

- Semantics is **global**: merged constants are treated the same **everywhere**
  - **Cannot** capture **local resolution**, e.g. some *J. Smith* are *John Smith*, some are *Jerry Smith*
  - Merges may be **blocked** by integrity constraints, e.g. functional dependency (FD)

Previous work

Issues of *ASPEn:*

- Semantics is **global**: merged constants are treated the same **everywhere**
  - **Cannot** capture **local resolution**, e.g. some *J. Smith* are *John Smith*, some are *Jerry Smith*
  - Merges may be **blocked** by integrity constraints, e.g. functional dependency (FD)
- Focuses on set maximal solutions, other natural optimality criteria, can be used to **select preferred solutions**

Previous work

Issues of *ASPEn:*

- Semantics is **global**: merged constants are treated the same **everywhere**
  - **Cannot** capture **local resolution**, e.g. some *J. Smith* are *John Smith*, some are *Jerry Smith*
  - Merges may be **blocked** by integrity constraints, e.g. functional dependency (FD)
- Focuses on set maximal solutions, other natural optimality criteria, can be used to **select preferred solutions**

Goal: enhancing ASPEn with local semantics + alt optimality criteria

LACE framework *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\boldsymbol{n},\boldsymbol{a},\boldsymbol{t},l) \wedge Patient(y,\boldsymbol{n'},\boldsymbol{a},\boldsymbol{t},l') \wedge \boldsymbol{n} \approx \boldsymbol{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\boldsymbol{x},n,\boldsymbol{d},\boldsymbol{t}) \wedge Doctor(\boldsymbol{y},n',\boldsymbol{d},\boldsymbol{t}) \wedge App(\boldsymbol{x},\boldsymbol{p}) \wedge App(\boldsymbol{y},\boldsymbol{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \boldsymbol{n} \neq \boldsymbol{n'} \Rightarrow \bot$

LACE framework *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\boldsymbol{n},\boldsymbol{a},\boldsymbol{t},l) \wedge Patient(y,\boldsymbol{n'},\boldsymbol{a},\boldsymbol{t},l') \wedge \boldsymbol{n} \approx \boldsymbol{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\boldsymbol{x},n,\boldsymbol{d},\boldsymbol{t}) \wedge Doctor(\boldsymbol{y},n',\boldsymbol{d},\boldsymbol{t}) \wedge App(\boldsymbol{x},\boldsymbol{p}) \wedge App(\boldsymbol{y},\boldsymbol{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \boldsymbol{n} \neq \boldsymbol{n'} \Rightarrow \bot$

LACE framework *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\boldsymbol{n},\boldsymbol{a},\boldsymbol{t},l) \wedge Patient(y,\boldsymbol{n'},\boldsymbol{a},\boldsymbol{t},l') \wedge \boldsymbol{n \approx n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\boldsymbol{x},n,\boldsymbol{d},\boldsymbol{t}) \wedge Doctor(\boldsymbol{y},n',\boldsymbol{d},\boldsymbol{t}) \wedge App(\boldsymbol{x},\boldsymbol{p}) \wedge App(\boldsymbol{y},\boldsymbol{p}) \rightsquigarrow Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \boldsymbol{n \neq n'} \Rightarrow \bot$

LACE framework *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

LACE framework *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

LACE framework *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\boldsymbol{n},\boldsymbol{a},\boldsymbol{t},l) \wedge Patient(y,\boldsymbol{n'},\boldsymbol{a},\boldsymbol{t},l') \wedge \boldsymbol{n} \approx \boldsymbol{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\boldsymbol{x},n,\boldsymbol{d},\boldsymbol{t}) \wedge Doctor(\boldsymbol{y},n',\boldsymbol{d},\boldsymbol{t}) \wedge App(\boldsymbol{x},\boldsymbol{p}) \wedge App(\boldsymbol{y},\boldsymbol{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \boldsymbol{n} \neq \boldsymbol{n'} \Rightarrow \bot$

LACE framework *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

EqRel E $E = \{\}$

Global semantics *[Bienvenu et al. PODS 2022]*

Database D

Appointment

did	pid
d1	p1
d2	p2
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \bot$

EqRel E $\quad E = \{\mathbf{(p1,p2)}\}$

Global semantics *[Bienvenu et al. PODS 2022]*

Induced database D_E

Appointment

did	pid
d1	**p1**
d2	**p1**
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \bot$

EqRel E $\quad E = \{\mathbf{(p1,p2)}\}$

Global semantics *[Bienvenu et al. PODS 2022]*

Induced database D_E

Appointment

did	pid
d1	p1
d2	p1
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \bot$

EqRel E $\quad E = \{\mathbf{(p1,p2)}\}$

Global semantics *[Bienvenu et al. PODS 2022]*

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	p3
d3	p4

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\boldsymbol{n},\boldsymbol{a},\boldsymbol{t},l) \wedge Patient(y,\boldsymbol{n'},\boldsymbol{a},\boldsymbol{t},l') \wedge \boldsymbol{n \approx n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\boldsymbol{x},n,\boldsymbol{d},\boldsymbol{t}) \wedge Doctor(\boldsymbol{y},n',\boldsymbol{d},\boldsymbol{t}) \wedge App(\boldsymbol{x},\boldsymbol{p}) \wedge App(\boldsymbol{y},\boldsymbol{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \boldsymbol{n \neq n'} \Rightarrow$

EqRel E $\quad E' = \{(p1,p2), \boldsymbol{(d1,d2)}\}$

Global semantics *[Bienvenu et al. PODS 2022]*

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	**p3**
d3	**p3**

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p3	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d1	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n \approx n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n \neq n'} \Rightarrow \bot$

EqRel E $\quad E = \{(p1,p2),(d1,d2),\mathbf{(p3,p4)}\}$

Global semantics *[Bienvenu et al. PODS 2022]*

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	p3
d3	p3

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p3	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

EqRel E $\quad E = \{(p1,p2),(d1,d2),(p3,p4)\}$ → A *solution* is an E s.t. D_E satisfies **H** and **C**

Issue of global semantics

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	p3
d3	p3

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p3	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim\!> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

Patient tuples with the same pid must not have different names

EqRel E $\quad E = \{(p1,p2), (p3,p4), (d1,d2)\}$ → A *solution* is a E s.t. D_E satisfies **H** and **C**

Issue of global semantics

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	p3
d3	p3

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p3	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} => Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} => \bot$

$Patient(p,\mathbf{n},a,t,l) \wedge Patient(p,\mathbf{n'},a',t,l') \sim> Eq(n,n')$

EqRel E $\quad E = \{(p1,p2), (p3,p4), (d1,d2)\}$

Issue of global semantics

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	p3
d3	p3

Patient

pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p3	J.Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(p,\mathbf{n},a,t,l) \wedge Patient(p,\mathbf{n'},a',t,l') \sim> Eq(n,n')$

EqRel E $\quad E = \{(p1,p2), (p3,p4), (d1,d2), (J.Smith, John Smith), (J.Smith, Jerry Smith)\}$

Issue of global semantics

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	p3
d3	p3

Patient

pid	name	age	phone	allergy
p1	John Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	John Smith	32	1234-5	ibuprofen
p3	John Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(p,\mathbf{n},a,t,l) \wedge Patient(p,\mathbf{n'},a',t,l') \sim> Eq(n,n')$

EqRel E $\quad E = \{(p1,p2), (p3,p4), (d1,d2), (J.Smith, John\ Smith), (J.Smith, Jerry\ Smith)\}$

Names of different patients now cannot be distinguished in the solution!

Issue of global semantics

Induced database D_E

Appointment

did	pid
d2	p1
d2	p1
d3	p3
d3	p3

Patient

pid	name	age	phone	allergy
p1	John Smith	32	12345	brupen
p1	John Smith	32	12345	aspirin
p3	John Smith	32	1234-5	ibuprofen
p3	John Smith	32	6789	ibuprofen

Doctor

did	name	dept	phone
d2	ブラックジャック	surgeon	66677
d2	Black jack	surgeon	66677
d3	Tezuka	skin	77333

Specification L

Hard rule: $Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$

Soft rule: $Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \sim> Eq(x,y)$

Denial constraint: $Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(p,\mathbf{n},a,t,l) \wedge Patient(p,\mathbf{n'},a',t,l') \sim> Eq(n,n')$

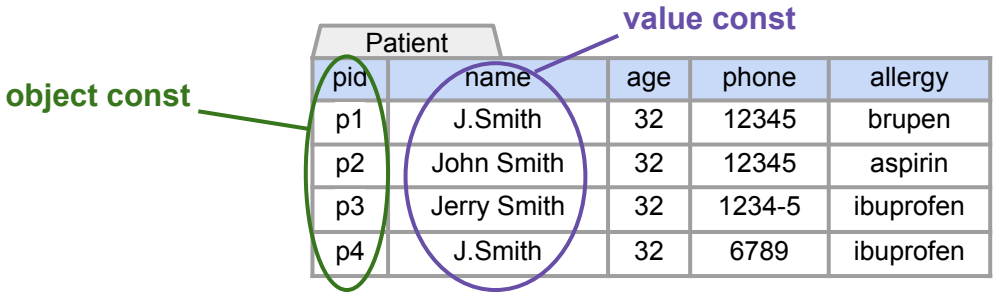
EqRel E

$E = \{(p1,p2), (p3,p4), (d1,d2), (J.Smith, John\ Smith), (J.Smith, Jerry\ Smith)\}$

Need to consider the **context** when merging some of the constants!

Combining global and local semantics *[Bienvenu et al. KR 2023]*

Database D



pid	name	age	phone	allergy
p1	J.Smith	32	12345	brupen
p2	John Smith	32	12345	aspirin
p3	Jerry Smith	32	1234-5	ibuprofen
p4	J.Smith	32	6789	ibuprofen

Specification L

$$Patient(x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow Eq(x,y)$$

$$Doctor(\mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(\mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(\mathbf{x},\mathbf{p}) \wedge App(\mathbf{y},\mathbf{p}) \rightsquigarrow Eq(x,y)$$

$$Patient(p,n,a,t,l) \wedge Patient(p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$$

$$Patient(p,\mathbf{n},a,t,l) \wedge Patient(p,\mathbf{n'},a',t,l') \rightsquigarrow Eq(n,n')$$

Combining global and local semantics *[Bienvenu et al. KR 2023]*

Database D

Patient

tid	pid	name	age	phone	allergy
t1	p1	J.Smith	32	12345	brupen
t2	p2	John Smith	32	12345	aspirin
t3	p3	Jerry Smith	32	1234-5	ibuprofen
t4	p4	J.Smith	32	6789	ibuprofen

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow EqO(x,y)$ **object merge**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow EqO(x,y)$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(i, p,\mathbf{n},a,t,l) \wedge Patient(i', p,\mathbf{n'},a',t,l') \rightsquigarrow EqV(\langle i,2\rangle,\langle i',2\rangle)$ **value merge**

EqRels E,V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle,\langle t_2,2\rangle),(\langle t_3,2\rangle,\langle t_4,2\rangle)\}$

Combining global and local semantics *[Bienvenu et al. KR 2023]*

Database D

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
t1	p1	J.Smith	32	12345	brupen
t2	p2	John Smith	32	12345	aspirin
t3	p3	Jerry Smith	32	1234-5	ibuprofen
t4	p4	J.Smith	32	6789	ibuprofen

Specification L

$Patient(i, x, \mathbf{n}, \mathbf{a}, \mathbf{t}, l) \wedge Patient(i', y, \mathbf{n'}, \mathbf{a}, \mathbf{t}, l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow EqO(x,y)$ **object merge**

$Doctor(i, \mathbf{x}, n, \mathbf{d}, \mathbf{t}) \wedge Doctor(i', \mathbf{y}, n', \mathbf{d}, \mathbf{t}) \wedge App(i'', \mathbf{x}, \mathbf{p}) \wedge App(i''', \mathbf{y}, \mathbf{p}) \rightsquigarrow EqO(x,y)$

$Patient(i, p, n, a, t, l) \wedge Patient(i', p, n', a', t', l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(i, p, \mathbf{n}, a, t, l) \wedge Patient(i', p, \mathbf{n'}, a', t, l') \rightsquigarrow EqV(\langle i,2\rangle, \langle i',2\rangle)$ **value merge**

EqRels E, V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle, \langle t_2,2\rangle), (\langle t_3,2\rangle, \langle t_4,2\rangle)\}$

Combining global and local semantics *[Bienvenu et al. KR 2023]*

Database D

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
t1	p1	J.Smith	32	12345	brupen
t2	p2	John Smith	32	12345	aspirin
t3	p3	Jerry Smith	32	1234-5	ibuprofen
t4	p4	J.Smith	32	6789	ibuprofen

Specification L

$Patient(i, x, \mathbf{n}, \mathbf{a}, \mathbf{t}, l) \wedge Patient(i', y, \mathbf{n'}, \mathbf{a}, \mathbf{t}, l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow EqO(x,y)$ **object merge**

$Doctor(i, \mathbf{x}, n, \mathbf{d}, \mathbf{t}) \wedge Doctor(i', \mathbf{y}, n', \mathbf{d}, \mathbf{t}) \wedge App(i'', \mathbf{x}, \mathbf{p}) \wedge App(i''', \mathbf{y}, \mathbf{p}) \rightsquigarrow EqO(x,y)$

$Patient(i, p, n, a, t, l) \wedge Patient(i', p, n', a', t', l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(i, p, \mathbf{n}, a, t, l) \wedge Patient(i', p, \mathbf{n'}, a', t, l') \rightsquigarrow EqV(\langle i,2\rangle, \langle i',2\rangle)$ **value merge**

EqRels E, V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle, \langle t_2,2\rangle), (\langle t_3,2\rangle, \langle t_4,2\rangle)\}$

Combining global and local semantics *[Bienvenu et al. KR 2023]*

Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow EqO(x,y)$ **object merge**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow EqO(x,y)$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(i, p,\mathbf{n},a,t,l) \wedge Patient(i', p,\mathbf{n'},a',t,l') \rightsquigarrow EqV(\langle i,2\rangle,\langle i',2\rangle)$ **value merge**

EqRels E,V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle,\langle t_2,2\rangle),(\langle t_3,2\rangle,\langle t_4,2\rangle)\}$

replace each **object** constant c with the set $\{c' \mid (c,c') \in E\}$,

replace each **value** constant at the location $\langle t,i\rangle$ with the set $\{ t'[j] \mid (\langle t,i\rangle,\langle t',j\rangle) \in V\}$

Combining global and local semantics *[Bienvenu et al. KR 2023]*

Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow EqO(x,y)$ **object merge**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \sim\!> EqO(x,y)$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(i, p,\mathbf{n},a,t,l) \wedge Patient(i', p,\mathbf{n'},a',t,l') \sim\!> EqV(\langle i,2\rangle,\langle i',2\rangle)$ **value merge**

EqRels E,V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle,\langle t_2,2\rangle),(\langle t_3,2\rangle,\langle t_4,2\rangle)\}$

replace each **object** constant c with the set $\{c' \mid (c,c') \in E\}$,

replace each **value** constant at the location $\langle t,i\rangle$ with the set $\{ t'[j] \mid (\langle t,i\rangle,\langle t',j\rangle) \in V\}$

Combining global and local semantics *[Bienvenu et al. KR 2023]*

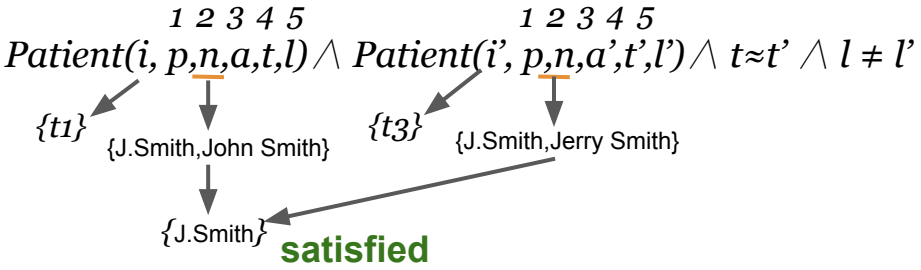
Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets

Query evaluation on $D_{E,V}$



Combining global and local semantics *[Bienvenu et al. KR 2023]*

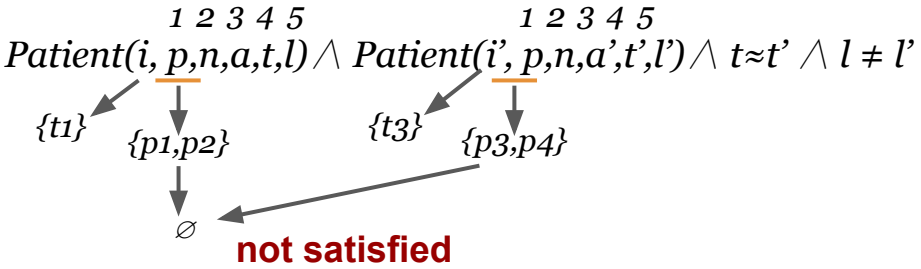
Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets

Query evaluation on $D_{E,V}$



Combining global and local semantics *[Bienvenu et al. KR 2023]*

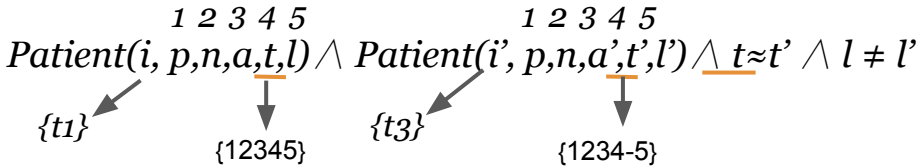
Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets
- For $t \approx t'$, satisfied if $\approx$ holds for two constants from the mapped sets of t and t', resp

Query evaluation on $D_{E,V}$



12345 ≈ 1234-5

satisfied

Combining global and local semantics *[Bienvenu et al. KR 2023]*

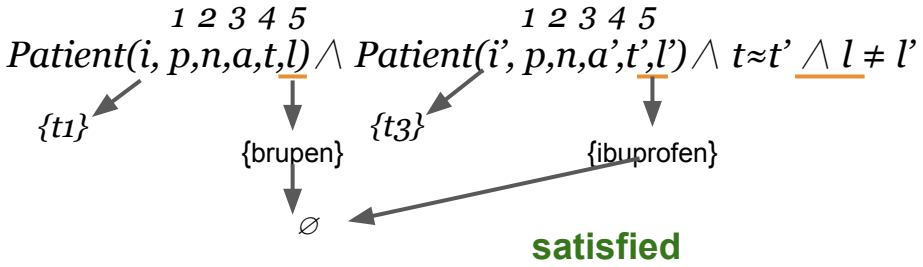
Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

- Same variable mapped to different sets depending on the context (tuple) it occurs
- Final assignment is the intersection of such sets
- For $t \approx t'$, satisfied if $\approx$ holds for two constants from the mapped sets of t and t', resp
- For $l \neq l'$, satisfied if intersection mapped sets of l and l' is empty

Query evaluation on $D_{E,V}$



Combining global and local semantics *[Bienvenu et al. KR 2023]*

Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow EqO(x,y)$ **object merge**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \sim> EqO(x,y)$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(i, p,\mathbf{n},a,t,l) \wedge Patient(i', p,\mathbf{n'},a',t,l') \sim> EqV(\langle i,2\rangle,\langle i',2\rangle)$ **location merge**

EqRels E,V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle,\langle t_2,2\rangle),(\langle t_3,2\rangle,\langle t_4,2\rangle)\}$

↓

A *solution* is an $\langle E,V\rangle$ s.t. $D_{E,V}$ satisfies **H** and **C**

Combining global and local semantics *[Bienvenu et al. KR 2023]*

Induced database $D_{E,V}$

Patient

tid	1:pid	2:name	3:age	4:phone	5:allergy
{t1}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{brupen}
{t2}	{p1,p2}	{J.Smith,John Smith}	{32}	{12345}	{aspirin}
{t3}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{1234-5}	{ibuprofen}
{t4}	{p3,p4}	{J.Smith,Jerry Smith}	{32}	{6789}	{ibuprofen}

Specification L

$Patient(i, x,\mathbf{n},\mathbf{a},\mathbf{t},l) \wedge Patient(i', y,\mathbf{n'},\mathbf{a},\mathbf{t},l') \wedge \mathbf{n} \approx \mathbf{n'} \Rightarrow EqO(x,y)$ **object merge**

$Doctor(i, \mathbf{x},n,\mathbf{d},\mathbf{t}) \wedge Doctor(i', \mathbf{y},n',\mathbf{d},\mathbf{t}) \wedge App(i'', \mathbf{x},\mathbf{p}) \wedge App(i''', \mathbf{y},\mathbf{p}) \rightsquigarrow EqO(x,y)$

$Patient(i, p,n,a,t,l) \wedge Patient(i', p,n',a',t',l') \wedge \mathbf{n} \neq \mathbf{n'} \Rightarrow \bot$

$Patient(i, p,\mathbf{n},a,t,l) \wedge Patient(i', p,\mathbf{n'},a',t,l') \rightsquigarrow EqV(\langle i,2\rangle,\langle i',2\rangle)$ **location merge**

EqRels E,V $\quad E = \{(p_1,p_2), (p_3,p_4), (d_1,d_2)\}$ $\quad V = \{(\langle t_1,2\rangle,\langle t_2,2\rangle),(\langle t_3,2\rangle,\langle t_4,2\rangle)\}$

A *solution* is an $\langle E,V\rangle$ s.t. $D_{E,V}$ satisfies **H** and **C**

Space of solutions → Set inclusion optimal solutions

Construct ASP encoding

- Database tuples as facts, plus `proj(I,P,T)`, the `P`-th position of tuple `I` has value `T`
- `eqo/2, eqv/4` axiomatise as equivalence relations

Construct ASP encoding

- Database tuples as facts, plus `proj(I,P,T)`, the `P`-th position of tuple `I` has value `T`
- `eqo/2, eqv/4` axiomatise as equivalence relations

Patient(i, p,n,a,t,l)∧ Patient(i', p,n',a',t,l')~>EqV(<i,2>,<i',2>)

```
{eqv(I1,2,I2,2),neqv(I1,2,I2,2)}=1:-patient(I',P2,_,_,_,_),
                    patient(I,P1,_,_,_,_),eqo(P1,P2),val(I1,4,T),val(I2,4,T).
```

Construct ASP encoding

- Database tuples as facts, plus `proj(I,P,T)`, the `P`-th position of tuple `I` has value `T`
- `eqo/2`, `eqv/4` axiomatise as equivalence relations

Patient(i, p,n,a,t,l) ∧ *Patient(i', p,n',a',t,l')* **~>*EqV(<i,2>,<i',2>)***

{eqv(I1,2,I2,2),neqv(I1,2,I2,2)}=1 :- *patient(I',P2,_,_,_,_),*
patient(I,P1,_,_,_,_), eqo(P1,P2),val(I1,4,T),val(I2,4,T).

➔ Encode **soft rules** using **choice head**

Construct ASP encoding

- Database tuples as facts, plus `proj(I,P,T)`, the `P`-th position of tuple `I` has value `T`
- `eqo/2, eqv/4` axiomatise as equivalence relations

$Patient(i, \mathbf{p}, n, a, t, l) \wedge Patient(i', \mathbf{p}, n', a', t, l') \rightsquigarrow EqV(\langle i,2\rangle, \langle i',2\rangle)$

```
{eqv(I1,2,I2,2),neqv(I1,2,I2,2)}=1:-patient(I',P2,_,_,_,_),
                    patient(I,P1,_,_,_,_),eqo(P1,P2),val(I1,4,T),val(I2,4,T).
```

- Encode **soft rules** using **choice head**
- Replace **join variable**

  ➔ on **objects** by ***eqo*-atom with distinguished variables**

Construct ASP encoding

- Database tuples as facts, plus `proj(I,P,T)`, the `P`-th position of tuple `I` has value `T`
- `eqo/2, eqv/4` axiomatise as equivalence relations

*Patient(**i**, p,n,a,**t**,l) ∧ Patient(**i'**, p,n',a',**t**,l')~>EqV(<i,2>,<i',2>)*

```
{eqv(I1,2,I2,2),neqv(I1,2,I2,2)}=1:-patient(I',P2,_,_,_,_),
                    patient(I,P1,_,_,_,_),eqo(P1,P2),val(I1,4,T),val(I2,4,T).
```

- Encode **soft rules** using **choice head**
- Replace **join variable**
  - on **objects** by ***eqo*-atom with distinguished variables**

  ➔ on **values** by **intersection of value sets** associated to "locations"
    - `val(I1,4,T2):-eqv(I1,4,I2,4),proj(I2,4,T2).` (induced DB)

Construct ASP encoding

- Database tuples as facts, plus `proj(I,P,T)`, the `P`-th position of tuple `I` has value `T`
- `eqo/2, eqv/4` axiomatise as equivalence relations

*Patient(i, p,n,a,**t**,l) ∧ Patient(i', p,n',a',**t'**,l'), **t≈t'** ~>EqV(<i,2>,<i',2>)*

```
{eqv(I1,2,I2,2),neqv(I1,2,I2,2)}=1:-patient(I',P2,_,_,_,_),eqo(P1,P2),
  patient(I,P1,_,_,_,_),val(I1,4,T1),val(I2,4,T2),sim(T1,T2,S),S>90.
```

- Encode **soft rules** using **choice head**
- Replace **join variable**
  - on **objects** by ***eqo*-atom with distinguished variables**
  - on **values** by **intersection of value sets** associated to “locations”
- **Similarity atom**

  ➔ expanded with an **extra argument** denoting **threshold**

Construct ASP encoding

- Database tuples as facts, plus `proj(I,P,T)`, the `P`-th position of tuple `I` has value `T`
- `eqo/2, eqv/4` axiomatise as equivalence relations

*Patient(i, p,n,a,**t**,l) ∧ Patient(i', p,n',a',**t'**,l'), **t≈t'** ~>EqV(<i,2>,<i',2>)*

```
{eqv(I1,2,I2,2),neqv(I1,2,I2,2)}=1:-patient(I',P2,_,_,_,_),eqo(P1,P2),
  patient(I,P1,_,_,_,_),val(I1,4,T1),val(I2,4,T2),sim(T1,T2,S),S>90.
```

- Encode **soft rules** using **choice head**
- Replace **join variable**
  - on **objects** by ***eqo*-atom with distinguished variables**
  - on **values** by **intersection of value sets** associated to "locations"
- **Similarity atom**
  - expanded with an **extra argument** denoting **threshold**
  - ➔ evaluated on a pair of **fresh variables** associated to value sets

Optimality Criteria

Set of all triples such that each pair $p1,p2$ (of object or value merge) satisfies rule $r \in L$ in $D_{E,V}$.

actP*(D, E, V, L)* E.g *(p1,p2, r)*

Optimality Criteria

Set of all triples such that each pair $p = p1,p2$ (of object or value merge) satisfies rule $r \in L$ in $D_{E,V}$.

$\mathbf{actP}(D, E, V, L)$ E.g $(p1,p2, r)$

4 target sets:

$\mathbf{EQ}(E,V) = E \cup V$

$\mathbf{SUP}(E,V) = \{(p,r) \in \mathbf{actP} \mid p \in E \cup V\}$

$\mathbf{ABS}(E,V) = \{p \mid (p,r) \in \mathbf{actP}, p \notin E \cup V\}$

$\mathbf{VIO}(E,V) = \{(p,r) \in \mathbf{actP} \mid p \notin E \cup V\}$

Optimality Criteria

Set of all triples such that each pair $p = p1,p2$ (of object or value merge) satisfies rule $r \in L$ in $D_{E,V}$.

$\mathbf{actP}(D, E, V, L)$ E.g $(p1,p2, r)$

Optimise **Set**/**Cardinality**:

$\mathbf{max\ EQ}(E,V) = E \cup V$

$\mathbf{max\ SUP}(E,V) = \{(p,r) \in \mathbf{actP} \mid p \in E \cup V\}$

$\mathbf{min\ ABS}(E,V) = \{p \mid (p,r) \in \mathbf{actP}, p \notin E \cup V\}$

$\mathbf{min\ VIO}(E,V) = \{(p,r) \in \mathbf{actP} \mid p \notin E \cup V\}$

The more merges (of objects and values) the better

Optimality Criteria

Set of all triples such that each pair $p = p1,p2$ (of object or value merge) satisfies rule $r \in L$ in $D_{E,V}$.

$\mathbf{actP}(D, E, V, L)$ E.g $(p1,p2, r)$

Optimise **Set**/**Cardinality**:

$\mathbf{max\ EQ}(E,V) = E \cup V$ The more merges (of objects and values) the better

$\mathbf{max\ SUP}(E,V) = \{(p,r) \in \mathbf{actP} \mid p \in E \cup V\}$ The more rules supporting the merges the better

$\mathbf{min\ ABS}(E,V) = \{p \mid (p,r) \in \mathbf{actP}, p \notin E \cup V\}$

$\mathbf{min\ VIO}(E,V) = \{(p,r) \in \mathbf{actP} \mid p \notin E \cup V\}$

Optimality Criteria

Set of all triples such that each pair $p = p1,p2$ (of object or value merge) satisfies rule $r \in L$ in $D_{E,V}$.

$\boldsymbol{actP}(D, E, V, L)$ E.g $(p1,p2, r)$

Optimise **Set**/**Cardinality**:

$\boldsymbol{max\ EQ}(E,V) = E \cup V$	The more merges (of objects and values) the better
$\boldsymbol{max\ SUP}(E,V) = \{(p,r) \in \boldsymbol{actP} \mid p \in E \cup V\}$	The more rules supporting the merges the better
$\boldsymbol{min\ ABS}(E,V) = \{p \mid (p,r) \in \boldsymbol{actP}, p \notin E \cup V\}$	The less unincluded active merges the better
$\boldsymbol{min\ VIO}(E,V) = \{(p,r) \in \boldsymbol{actP} \mid p \notin E \cup V\}$	

Optimality Criteria

Set of all triples such that each pair $p = p1,p2$ (of object or value merge) satisfies rule $r \in L$ in $D_{E,V}$.

$\textbf{actP}(D, E, V, L)$ E.g $(p1,p2, r)$

Optimise **Set**/**Cardinality**:

$\textbf{max EQ}(E,V) = E \cup V$	The more merges (of objects and values) the better
$\textbf{max SUP}(E,V) = \{(p,r) \in \textbf{actP} \mid p \in E \cup V\}$	The more rules supporting the merges the better
$\textbf{min ABS}(E,V) = \{p \mid (p,r) \in \textbf{actP}, p \notin E \cup V\}$	The less unincluded active merges the better
$\textbf{min VIO}(E,V) = \{(p,r) \in \textbf{actP} \mid p \notin E \cup V\}$	The less violations of soft rules the better

Optimality Criteria

Set of all triples such that each pair $p = p1,p2$ (of object or value merge) satisfies rule $r \in L$ in $D_{E,V}$.

$\boldsymbol{actP}(D, E, V, L)$ E.g $(p1,p2, r)$

Optimise **Set**/**Cardinality**:

$\boldsymbol{max\ EQ}(E,V) = E \cup V$	The more merges (of objects and values) the better
$\boldsymbol{max\ SUP}(E,V) = \{(p,r) \in \boldsymbol{actP} \mid p \in E \cup V\}$	The more rules supporting the merges the better
$\boldsymbol{min\ ABS}(E,V) = \{p \mid (p,r) \in \boldsymbol{actP}, p \notin E \cup V\}$	The less unincluded active merges the better
$\boldsymbol{min\ VIO}(E,V) = \{(p,r) \in \boldsymbol{actP} \mid p \notin E \cup V\}$	The less violations of soft rules the better

maxE and **maxS** are **the same under set optim**, so 7 diff'nt criteria overall

Computing Optimal Solutions

For a non-ground atom of a target set

- **Set**:
  - Set-inclusion preference (asprin)
  - Domain heuristic (heur) e.g. `#heuristic neqo(X,Y). [1, False]`
- **Cardinality** :
  - Weight preference
  - Weighted constraint (wc), e.g. `#minimize{1@1,X,Y,R:neqo(X,Y,R)}`
  - wc + heur

Experimental setup

Datasets:

- 6 **multi-table datasets** IMDB (movie), (**synthetic duplicates**) MUSIC, Pokemon
- Baselines ASPEn, 2 rule-based ER systems Magellan, JedAI

Name	#Rec	#Rel	#At	#Ref	#Dup	#Const
Imdb	30k	5	22	4	6k	64k
ImdbC	30k	5	22	4	6k	66k
Mu	41k	11	72	12	15k	156k
MuC	41k	11	72	12	15k	160k
MuCC	41k	11	72	12	15k	166k
Poke	240k	20	104	20	4k	349k

Table 4: Dataset Statistics. #-columns represent the number of records, relations, attributes, referential constraints and duplicates, respectively.

Performance

Method	Data	F_1	(P / R)	t_o	t_g	t_s	#DC	Data	F_1	(P / R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80** 78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77** 71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40 95.67	18.78	N/A	N/A	0/5		97.49	99.40 95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39 **99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36 94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39 **99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39 **98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63 82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51** 38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46 59.30	105.06	N/A	N/A	0/23		32.75	73.95 21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45** **95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86 **88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45** **95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17 **88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62** 38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97 29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47 19.75	101.47	N/A	N/A	0/23		2.1	1.08 46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83 65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71 69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5 **88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73** **74.03**	11,880	496.14	48.52	10/10

Performance

Method	Data	F_1	(P / R)	t_o	t_g	t_s	#DC	Data	F_1	(P / R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80** 78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77** 71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40 95.67	18.78	N/A	N/A	0/5		97.49	99.40 95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39 **99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36 94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39 **99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39 **98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63 82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51** 38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46 59.30	105.06	N/A	N/A	0/23		32.75	73.95 21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45** **95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86 **88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45** **95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17 **88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62** 38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97 29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47 19.75	101.47	N/A	N/A	0/23		2.1	1.08 46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83 65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71 69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5 **88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73** **74.03**	11,880	496.14	48.52	10/10

Performance

Method	Data	F_1	(P / R)	t_o	t_g	t_s	#DC	Data	F_1	(P / R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80** 78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77** 71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40 95.67	18.78	N/A	N/A	0/5		97.49	99.40 95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39 **99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36 94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39 **99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39 **98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63 82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51** 38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46 59.30	105.06	N/A	N/A	0/23		32.75	73.95 21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45** **95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86 **88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45** **95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17 **88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62** 38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97 29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47 19.75	101.47	N/A	N/A	0/23		2.1	1.08 46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83 65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71 69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5 **88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73** **74.03**	11,880	496.14	48.52	10/10

Performance

Method	Data	F_1	(P / R)	t_o	t_g	t_s	#DC	Data	F_1	(P / R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80** 78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77** 71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40 95.67	18.78	N/A	N/A	0/5		97.49	99.40 95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39 **99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36 94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39 **99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39 **98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63 82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51** 38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46 59.30	105.06	N/A	N/A	0/23		32.75	73.95 21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45** **95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86 **88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45** **95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17 **88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62** 38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97 29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47 19.75	101.47	N/A	N/A	0/23		2.1	1.08 46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83 65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71 69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5 **88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73** **74.03**	11,880	496.14	48.52	10/10

Performance

Method	Data	F_1	(P / R)	t_o	t_g	t_s	#DC	Data	F_1	(P / R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80** 78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77** 71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40 95.67	18.78	N/A	N/A	0/5		97.49	99.40 95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39 **99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36 94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39 **99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39 **98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63 82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51** 38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46 59.30	105.06	N/A	N/A	0/23		32.75	73.95 21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45** **95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86 **88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45** **95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17 **88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62** 38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97 29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47 19.75	101.47	N/A	N/A	0/23		2.1	1.08 46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83 65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71 69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5 **88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73** **74.03**	11,880	496.14	48.52	10/10

Performance

Method	Data	F_1	(P / R)	t_o	t_g	t_s	#DC	Data	F_1	(P / R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80** 78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77** 71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40 95.67	18.78	N/A	N/A	0/5		97.49	99.40 95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39 **99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36 94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39 **99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39 **98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63 82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51** 38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46 59.30	105.06	N/A	N/A	0/23		32.75	73.95 21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45** **95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86 **88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45** **95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17 **88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62** 38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97 29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47 19.75	101.47	N/A	N/A	0/23		2.1	1.08 46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83 65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71 69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5 **88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73** **74.03**	11,880	496.14	48.52	10/10

Performance

Method	Data	F_1	(P / R)	t_o	t_g	t_s	#DC	Data	F_1	(P / R)	t_o	t_g	t_s	#DC
Magellan	*Imdb*	88.09	**99.80** 78.83	**3.89**	N/A	N/A	0/5	*ImdbC*	83.05	**99.77** 71.12	**3.11**	N/A	N/A	0/5
JedAI		97.49	99.40 95.67	18.78	N/A	N/A	0/5		97.49	99.40 95.67	18.67	N/A	N/A	0/5
ASPEN		**99.27**	99.39 **99.14**	610.49	**11.51**	**0.082**	1/5		96.99	99.36 94.73	757.17	**11.75**	**0.088**	1/5
ASPEN$^+$		**99.27**	99.39 **99.14**	86.24	13.02	0.096	3/5		**99.13**	99.39 **98.87**	94.85	18.71	0.69	3/5
Magellan	*Mu*	89.78	98.63 82.38	**64.83**	N/A	N/A	0/23	*MuC*	55.54	**97.51** 38.83	**66.87**	N/A	N/A	0/23
JedAI		70.67	87.46 59.30	105.06	N/A	N/A	0/23		32.75	73.95 21.02	101.02	N/A	N/A	0/23
ASPEN		**97.64**	**99.45** **95.89**	666.01	**1.78**	**0.20**	21/23		90.31	91.86 **88.81**	695.76	**20.15**	**3.57**	17/23
ASPEN$^+$		**97.64**	**99.45** **95.89**	665.65	1.82	0.21	23/23		**90.46**	92.17 **88.81**	770.44	72.81	12.16	23/23
Magellan	*MuCC*	55.48	**97.62** 38.75	**64.81**	N/A	N/A	0/23	*Poke*	7.01	3.97 29.74	260.96	N/A	N/A	0/10
JedAI		31.04	72.47 19.75	101.47	N/A	N/A	0/23		2.1	1.08 46.56	**23.46**	N/A	N/A	0/10
ASPEN		71.18	77.83 65.58	718.38	**21.01**	**10.47**	16/23		81.78	99.71 69.31	4,454	**311.37**	**0.91**	10/10
ASPEN$^+$		**88.85**	89.5 **88.21**	993.31	97.51	23.02	23/23		**84.98**	**99.73** **74.03**	11,880	496.14	48.52	10/10

Qualitative Case

Place

tid	pid	name	type	address	coordinate
t1	p1	Kunsthalle	district	Kindikty	48.20 81.63
t2	p2	Kunstha1le	district	Kindikty dist.,	48.20-81.63

sr1: same {type}, similar {name} then could be same {pid}
d1: same {pid} must not have diff'nt {coordinate}

Ablum

tid	rid	artist	name	barcode	language
t3	r1	a1	chante les poetes	48.20 81.63	FR
t4	r2	a2	chanteLes poetes'	48.20-81.63	Fr

hr1: same {artist, language}, similar {name} then must be same {rid}
d2: same {rid} must not have diff'nt {barcode}

Qualitative Case

Place

tid	pid	name	type	address	coordinate
t1	p1	Kunsthalle	district	Kindikty	**48.20 81.63**
t2	p2	Kunstha1le	district	Kindikty dist.,	**48.20-81.63**

(p1,p2) is **blocked** without merging here

sr1: same {type}, similar {name} then could be same {pid}
d1: same {pid} must not have diff'nt {coordinate}

Ablum

tid	rid	artist	name	barcode	language
t3	r1	a1	chante les poetes	48.20 81.63	FR
t4	r2	a2	chanteLes poetes'	48.20-81.63	Fr

hr1: same {artist, language}, similar {name} then must be same {rid}
d2: same {rid} must not have diff'nt {barcode}

Qualitative Case

Place

tid	pid	name	type	address	coordinate
t1	p1	Kunsthalle	district	Kindikty	48.20 81.63
t2	p2	Kunstha1le	district	Kindikty dist.,	48.20-81.63

(p1,p2) is **blocked** without merging here

sr1: same {type}, similar {name} then could be same {pid}
d1: same {pid} must not have diff'nt {coordinate}

Ablum

tid	rid	artist	name	barcode	language
t3	r1	a1	chante les poetes	**48.20 8.163**	FR
t4	r2	a2	chanteLes poetes'	**48.20-81.63**	Fr

(r1,r2) leads to **no solution** without merging here

hr1: same {artist, language}, similar {name} then must be same {rid}
d2: same {rid} must not have diff'nt {barcode}

Qualitative Case

Place

tid	pid	name	type	address	coordinate
t1	p1	Kunsthalle	district	Kindikty	**48.20 81.63**
t2	p2	Kunstha1le	district	Kindikty dist.,	48.20-81.63

sr1: same {type}, similar {name} then could be same {pid}
d1: same {pid} must not have diff'nt {coordinate}

merge **has to be local** here, otherwise two diff'nt coordinates are **wrongly merged** !

Ablum

tid	rid	artist	name	barcode	language
t3	r1	a1	chante les poetes	**48.20 8.163**	FR
t4	r2	a2	chanteLes poetes'	48.20-81.63	Fr

hr1: same {artist, language}, similar {name} then must be same {rid}
d2: same {rid} must not have diff'nt {barcode}

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

set-optimisation

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.51	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.91	**99.73**	73.91	49.11	2	1.15

cardinality-optimisation

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

set-optimisation

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.51	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.91	**99.73**	73.91	49.11	2	1.15

cardinality-optimisation

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

set-optimisation

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.51	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.91	**99.73**	73.91	49.11	2	1.15

cardinality-optimisation

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

set-optimisation

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.5	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.9	**99.73**	73.91	49.11	2	1.15

cardinality-optimisation

Comparing optimality criteria

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxES/SS	90.50	92.20	**88.86**	**12.16**	50	1.86
	minAS	**91.88**	**95.11**	**88.86**	12.8	50	**1.7**
	minVS	91.7	94.73	88.85	12.44	50	1.78
MuCC	maxES/SS	88.85	89.5	**88.21**	23.02	50	1.67
	minAS	89.62	91.11	88.18	23.01	50	**1.54**
	minVS	**90.13**	**92.2**	88.15	**21.61**	50	2.48
Poke	maxES/SS	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	minAS	83.83	**99.73**	72.3	51.58	50	0.06
	minVS	84.62	**99.73**	73.48	56.87	50	**0.01**

Solving with **single** threads

Data	Method	$\overline{F}_1$	($\overline{P}$ /	$\overline{R}$)	t_s^1	#e	t_s^n
MuC	maxEC	90.51	92.20	**88.9**	35.09	50	**2.53**
	maxSC	90.52	92.21	**88.9**	**30.1**	16	12.09
	minAC	91.4	94.05	**88.9**	84.69	50	2.61
	minVC	**92.01**	**95.35**	88.89	48.97	50	5.66
MuCC	maxEC	88.83	89.45	**88.21**	66.71	1	N/A
	maxSC	88.85	89.50	**88.21**	**52.8**	2	629.56
	minAC	89.51	90.93	88.14	92.97	50	**2.12**
	minVC	**89.74**	**91.36**	88.18	67.82	50	8.42
Poke	maxEC	**84.98**	**99.73**	**74.03**	**48.52**	1	N/A
	maxSC	**84.98**	**99.73**	**74.03**	49.1	1	N/A
	minAC	84.80	**99.73**	73.75	49.23	50	**0.037**
	minVC	84.91	**99.73**	73.91	49.11	2	1.15

Solving with **36** threads!

Conclusion

- Introduced **ASPEn+**, an **ASP-based system for collective ER**, first of its kind that supports both **global** and **local** merges
- Defined **seven optimality criteria** for **preferred** ER solutions, and provided **complexity analyses**.
- Conducted extensive experiments, achieving **superior accuracy** on complex, **real-world** datasets, demonstrating the practical benefits of **local semantics** and **flexible optimisation**