

# Combining Entity Resolution and Query Answering in Ontologies

Domenico Lembo



SAPIENZA  
UNIVERSITÀ DI ROMA

15<sup>th</sup> International Conference on Knowledge Discovery,  
Knowledge Engineering and Knowledge Management  
(IC3K 2023)

November 13–15, 2023, Rome, Italy

Problem of recognizing **different representations** of the **same real-world object** (i.e. entity)

Complex task, because of:

- Numerous names, definitions, identifiers for the same object
- Different "objects" having the same name `[height=2.2cm]figures/Jordan.pdf`
- Errors in the data
- Missing values
- Abbreviations/Encodings
- ....

# Entity Resolution (ER)

Problem of recognizing **different representations** of the **same real-world object** (i.e. entity)

Complex task, because of:

- Numerous names, definitions, identifiers for the same object
- Different “objects” having the same name
- Errors in the data
- Missing values



[height=2.2cm]figures/Jordan.pdf

- Abbreviations/Encodings
- ....

# Entity Resolution (ER)

Problem of recognizing **different representations** of the **same real-world object** (i.e. entity)

Complex task, because of:

- Numerous names, definitions, identifiers for the same object



- Different “objects” having the same name



Michael Jordan  
(NBA player)



Michael Jordan  
(Actor)



Michael Jordan  
(Scientist)

- Errors in the data

- Missing values

- Abbreviations/Encodings

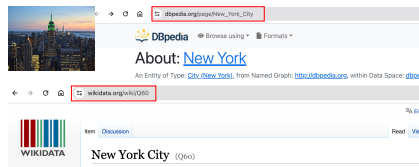
- ....

# Entity Resolution (ER)

Problem of recognizing **different representations** of the **same real-world object** (i.e. entity)

Complex task, because of:

- Numerous names, definitions, identifiers for the same object



The image shows two browser windows. The top window is DBpedia, displaying the page 'About: New York' with the URL 'dbpedia.org/page/New\_York\_City' highlighted in a red box. The bottom window is Wikidata, displaying the page 'New York City (Q60)' with the URL 'wikidata.org/wiki/Q60' highlighted in a red box.

- Different “objects” having the same name



Michael Jordan  
(NBA player)



Michael Jordan  
(Actor)



Michael Jordan  
(Scientist)

- Errors in the data
- Missing values

- Abbreviations/Encodings
- ....

# ER crucial in many data management tasks

- Relational Databases
- Data Integration
- Data Federation
- Data Warehousing
- Data Exchange
- Linked Data
- Ontology-based Data Access (aka Virtual Knowledge Graphs)
- .....

It is a central challenge when [preparing data](#) to make them suitable for analysis

# Alternative names for ER

Within the realm of relational databases, Entity Resolution (ER) conventionally centered on matching records by comparing the likeness of their attributes.

ER is thus often referred to as **Record Linkage** [Newcombe et al., 1959, Fellegi and Sunter, 1969].

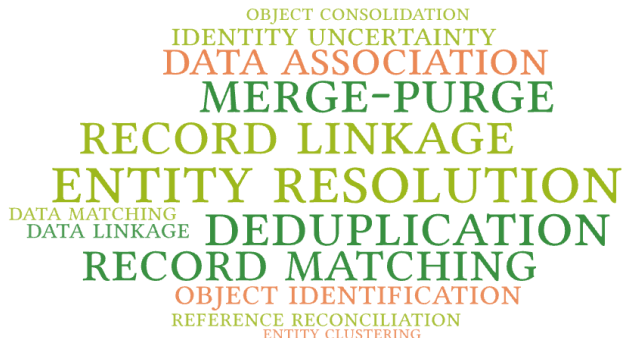
A plethora of alternative names for this problem has been proposed (even though with some nuances).

# Alternative names for ER

Within the realm of relational databases, Entity Resolution (ER) conventionally centered on matching records by comparing the likeness of their attributes.

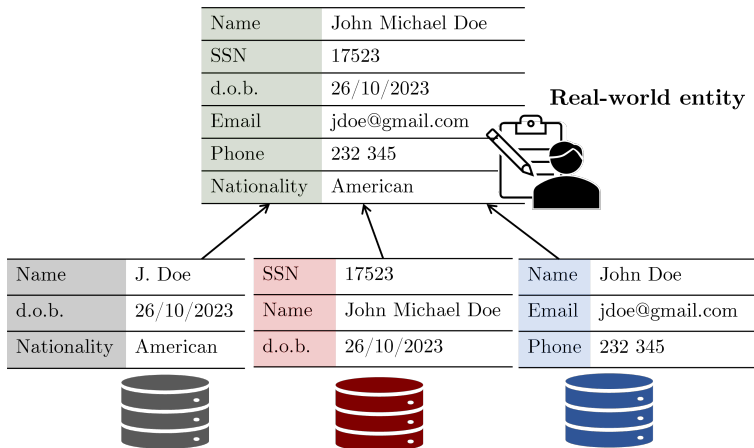
ER is thus often referred to as **Record Linkage** [Newcombe et al., 1959, Fellegi and Sunter, 1969].

A plethora of alternative names for this problem has been proposed (even though with some nuances).



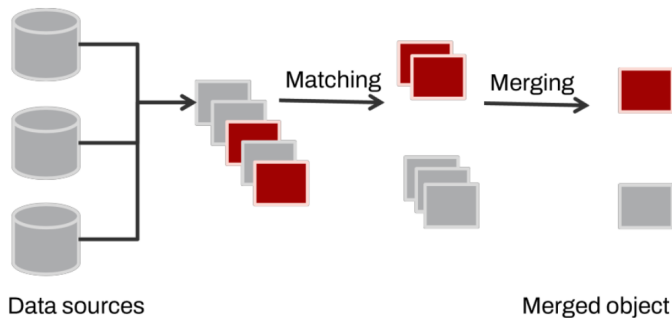


# Example in a relational context



# Matching & Merging

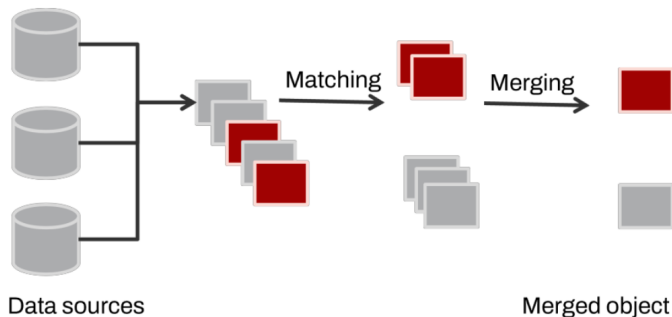
Despite the different names, the common idea is that when two different representations refer to the same real-world object (i.e. **match**), they are then **merged** into a common representation [Singla and Domingos, 2006, Benjelloun et al., 2009]



*For a while, let us focus on data records as those depicted above (even though what follows applies even to different data formats)*

# Matching & Merging

Despite the different names, the common idea is that when two different representations refer to the same real-world object (i.e. **match**), they are then **merged** into a common representation [Singla and Domingos, 2006, Benjelloun et al., 2009]



*For a while, let us focus on data records as those depicted above (even though what follows applies even to different data formats)*

Following [Papadakis et al., 2021], we may start mentioning:

- **Distance-based methods**, through string similarity functions [Guha et al., 2000, Guha et al., 2004, Chaudhuri et al., 2007]
- **Probabilistic Methods**, essentially designed around the formulation given by [Fellegi and Sunter, 1969], which aims at identifying **pairs of records that match**, **pairs of records that do no match**, as well as pairs that possibly match [Singla and Domingos, 2006, Bhattacharya and Getoor, 2007, Wu et al., 2020]
- **Supervised methods**, where ER is defined as a binary classification problem, where the output for each pair of record is a match or non-match decision [Köpcke et al., 2010]
- **Unsupervised methods**, e.g. through clustering techniques [Elfeky et al., 2002, Chaudhuri et al., 2005]
- Since the number of comparisons is typically very high, **Blocking Strategies** have been proposed to avoid unnecessary comparisons between records (see [Christen, 2012] for a survey)

# Rule-based approaches

A disadvantage of the methods mentioned so far is their limited **interpretability**

In particular, they are not able to incorporate (expressive) **constraints** that arise naturally in many settings [Arasu et al., 2009]

This issue calls for specialized techniques that directly formulate **rules** expressing the conditions under which a match occurs.

- Various forms of **Entity Resolution rules** have been proposed (e.g. [Arasu et al., 2009, Li et al., 2015, Bienvenu et al., 2022]), often of incomparable expressive power
- E.g., **Matching Dependencies** [Fan, 2008, Bertossi et al., 2013] are rules of the form

$$R_1[\mathbf{X}_1] \approx R_2[\mathbf{X}_2] \rightarrow R_1[\mathbf{Y}_1] \doteq R_2[\mathbf{Y}_2]$$

saying that if the projection  $p_1$  of a fact in  $R_1$  on attributes  $\mathbf{X}_1$  is similar to the projection  $p_2$  of a fact in  $R_2$  over  $\mathbf{X}_2$ , then the values in  $\mathbf{Y}_1$  of  $p_1$  and the values in  $\mathbf{Y}_2$  of  $p_2$  must be merged.

Another (orthogonal) limit of certain proposals (as in [Fellegi and Sunter, 1969]) is that they consider all pairs of candidate matches as **independent** (and identically distributed).

A generalization of this problem considers **a set of records that are related**, i.e., such relationships are taken into account to empower the ER process [Singla and Domingos, 2006, Arasu et al., 2009, Kouki et al., 2019, Bienvenu et al., 2022].

As a consequence, collective ER causes that some matching decisions depend on other matching decisions.

*Intuition: if two movies are the same, their directors are the same, which in turn may lead to conclude that other pairs of movies by the same directors should be matched.*

Once records match, they can be merged in various different ways

	<b>Name</b>	<b>Phone</b>	<b>E-mail</b>
$r_1$	{JohnDoe}	{235-2635}	{jdoe@yahoo}
$r_2$	{J.Doe}	{234-4358}	
$r_3$	{JohnD.}	{234-4358}	{jdoe@yahoo}

Once records match, they can be merged in various different ways

	<b>Name</b>	<b>Phone</b>	<b>E-mail</b>
$r_1$	{JohnDoe}	{235-2635}	{jdoe@yahoo}
$r_2$	{J.Doe}	{234-4358}	
$r_3$	{JohnD.}	{234-4358}	{jdoe@yahoo}
$r_4$	{John Doe}	{234-4358, 235-2635}	{jdoe@yahoo}

Names are combined into a “normalized” representative, whereas set-unions is performed on phone numbers and emails [Benjelloun et al., 2009]



# General Properties for Match and Merge Functions

[Benjelloun et al., 2009] proposes natural and general properties (ICAR) for match and merge functions

As notation, " $r \approx r'$ " says that  $r$  matches with  $r'$  and  $\langle r, r' \rangle$  indicates the result of merging  $r$  and  $r'$ :

(Idempotence)  $\forall r : r \approx r$  and  $\langle r, r \rangle = r$ .

(Commutativity)  $\forall r_1, r_2 : r_1 \approx r_2$  if and only if  $r_2 \approx r_1$  and if  $r_1 \approx r_2$  then  $\langle r_1, r_2 \rangle = \langle r_2, r_1 \rangle$ .

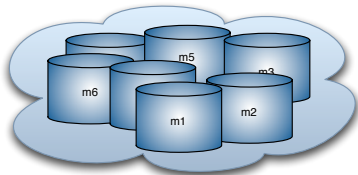
(Associativity)  $\forall r_1, r_2, r_3$  such that  $\langle r_1, \langle r_2, r_3 \rangle \rangle$  and  $\langle \langle r_1, r_2 \rangle, r_3 \rangle$  exist, then  $\langle r_1, \langle r_2, r_3 \rangle \rangle = \langle \langle r_1, r_2 \rangle, r_3 \rangle$ .

(Representativity) If  $r_3 = \langle r_1, r_2 \rangle$  then for any  $r_4$  such that  $r_1 \approx r_4$ , we also have  $r_3 \approx r_4$ .

# Query Answering

Query answering has not always been explicitly considered in ER frameworks

However, not all ER approaches produce **one** clean database: this may depend on the **order in which match and merges are applied** [Bertossi et al., 2013], or because of **some non-determinism of the ER semantics** [Bienvenu et al., 2022], or because of **incompleteness in the data** [Fagin et al., 2023]



Thus, it might be not clear which database (i.e. model) to choose as the result of the ER process

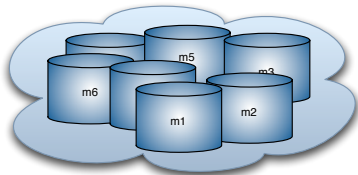
Of course, this situation heavily affect the notion of **query answering!**

A mathematically rigorous approach in these cases is to cast in the framework the notion of certain answers (or even possible answers) [Bienvenu et al., 2022, Fagin et al., 2023]

# Query Answering

Query answering has not always been explicitly considered in ER frameworks

However, not all ER approaches produce **one** clean database: this may depend on the **order in which match and merges are applied** [Bertossi et al., 2013], or because of **some non-determinism of the ER semantics** [Bienvenu et al., 2022], or because of **incompleteness in the data** [Fagin et al., 2023]



Thus, it might be not clear which database (i.e. model) to choose as the result of the ER process

Of course, this situation heavily affect the notion of **query answering!**

A mathematically rigorous approach in these cases is to cast in the framework the notion of **certain answers** (or even possible answers) [Bienvenu et al., 2022, Fagin et al., 2023]

In Record Linkage approaches, entities are not always explicitly represented, rather **a record is a set of attribute-value pairs representing entity properties**

In Ontologies (e.g., Knowledge Graphs), entities are denoted through special constants (e.g., URIs)

*Name(Doe<sub>1</sub>, John Doe)*

In this context, ER can be defined as the problem of determining, for each pair  $(c_1, c_2)$  of constants, whether they represent the same real-world entity, and can thus be merged [Bienvenu et al., 2022]

Additionally, the terminological component of an ontology (TBox) comprises axioms that can be used for deduction processes

In **rule-based ER over ontologies**, a major challenge is thus understanding the **interaction of the TBox with ER rules**, as well as how to **compute answers to queries** over the TBox, **in combination with ER**

**Note:** *from now on, entity IDs are simply called **entities!** We use instead **objects** or **individuals** to refer to the real-world items denoted by entities*

In Record Linkage approaches, entities are not always explicitly represented, rather **a record is a set of attribute-value pairs representing entity properties**

In Ontologies (e.g., Knowledge Graphs), entities are denoted through special constants (e.g., URIs)

*Name(Doe<sub>1</sub>, John Doe)*

In this context, ER can be defined as the problem of **determining, for each pair  $(c_1, c_2)$  of constants, whether they represent the same real-world entity**, and can thus be merged [Bienvenu et al., 2022]

Additionally, the terminological component of an ontology (TBox) comprises axioms that can be used for deduction processes

In **rule-based ER over ontologies**, a major challenge is thus understanding the **interaction of the TBox with ER rules**, as well as how to **compute answers to queries** over the TBox, **in combination with ER**

**Note:** *from now on, entity IDs are simply called **entities!** We use instead **objects** or **individuals** to refer to the real-world items denoted by entities*

In Record Linkage approaches, entities are not always explicitly represented, rather **a record is a set of attribute-value pairs representing entity properties**

In Ontologies (e.g., Knowledge Graphs), entities are denoted through special constants (e.g., URIs)

*Name(Doe<sub>1</sub>, John Doe)*

In this context, ER can be defined as the problem of **determining, for each pair  $(c_1, c_2)$  of constants, whether they represent the same real-world entity**, and can thus be merged [Bienvenu et al., 2022]

Additionally, the terminological component of an ontology (TBox) comprises axioms that can be used for deduction processes

In **rule-based ER over ontologies**, a major challenge is thus understanding the **interaction of the TBox with ER rules**, as well as how to **compute answers to queries** over the TBox, **in combination with ER**

*Note: from now on, entity IDs are simply called **entities!** We use instead **objects** or **individuals** to refer to the real-world items denoted by entities*

In Record Linkage approaches, entities are not always explicitly represented, rather **a record is a set of attribute-value pairs representing entity properties**

In Ontologies (e.g., Knowledge Graphs), entities are denoted through special constants (e.g., URIs)

*Name(Doe<sub>1</sub>, John Doe)*

In this context, ER can be defined as the problem of **determining, for each pair  $(c_1, c_2)$  of constants, whether they represent the same real-world entity**, and can thus be merged [Bienvenu et al., 2022]

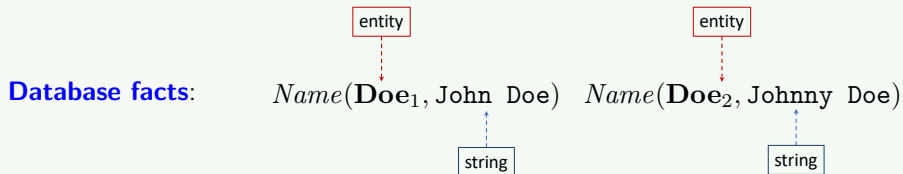
Additionally, the terminological component of an ontology (TBox) comprises axioms that can be used for deduction processes

In **rule-based ER over ontologies**, a major challenge is thus understanding the **interaction of the TBox with ER rules**, as well as how to **compute answers to queries** over the TBox, **in combination with ER**

**Note:** *from now on, entity IDs are simply called **entities!** We use instead **objects** or **individuals** to refer to the real-world items denoted by entities*

# ER in Ontologies: Matching entities may call for matching values

## Example



**ER Rule:** Entities associated to names with Jaccard similarity above 0.7 are denoting the same individual

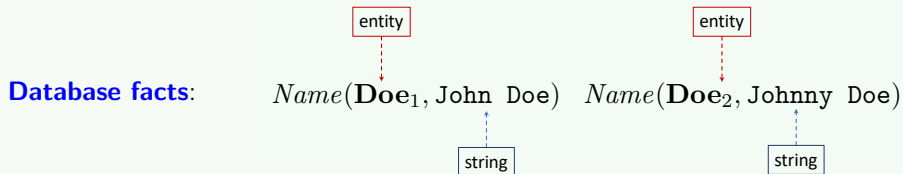
Jaccard similarity of John Doe and Johnny Doe is 0.8  $\rightarrow$   $\mathbf{Doe}_1$  and  $\mathbf{Doe}_2$  denote the same individual

If we also want every individual to have only one name  $\rightarrow$  Inconsistency caused by values.



# ER in Ontologies: Matching entities may call for matching values

## Example



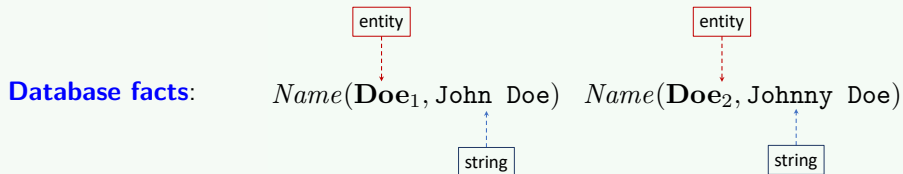
**ER Rule:** Entities associated to names with Jaccard similarity above 0.7 are denoting the same individual

Jaccard similarity of John Doe and Johnny Doe is 0.8  $\rightarrow$  **Doe<sub>1</sub>** and **Doe<sub>2</sub>** denote the same individual

If we also want every individual to have only one name  $\rightarrow$  Inconsistency caused by values.

# ER in Ontologies: Matching entities may call for matching values

## Example



**ER Rule:** Entities associated to names with Jaccard similarity above 0.7 are denoting the same individual

Jaccard similarity of John Doe and Johnny Doe is 0.8  $\rightarrow$  **Doe<sub>1</sub>** and **Doe<sub>2</sub>** denote the same individual

If we also want every individual to have only one name  $\rightarrow$  **Inconsistency caused by values.**

We consider the recent framework proposed in [Fagin et al., 2023], which:

- Proposes a **new collective approach** for ER in ontologies with  **$n$ -ary predicates each of whose arguments ranges over entities or ranges over values**
- Considers ontologies where ground atoms are coupled with tuple-generating dependencies (**TGDs**) and equality-generating dependencies (**EGDs**).

TGDs and EGDs can express axioms used in Description Logics or Datalog<sup>+/-</sup>, as well as ER rules

- Defines a declarative semantics that **merges entities globally** and **merges values locally**
- Defines the notion of **certain answers for conjunctive queries** (CQs) in this framework
- Provides a procedure for **computing the certain answers to CQs**, based on a tailored **never-failing chase** technique

# Example

Consider the following ontology  $\mathcal{O} = (\mathcal{T}, \mathbf{D})$

TBox  $\mathcal{T}$

( $r_1$ )  $Name(p_1, n_1) \wedge Name(p_2, n_2) \wedge JaccSim(n_1, n_2, 0.7) \rightarrow p_1 = p_2$

( $r_2$ )  $Name(p, n_1) \wedge Name(p, n_2) \rightarrow n_1 = n_2$

( $r_3$ )  $HPhone(p, f_1) \wedge HPhone(p, f_2) \rightarrow f_1 = f_2$

( $r_4$ )  $HPhone(p_1, f) \wedge HPhone(p_2, f) \rightarrow SameHouse(p_1, p_2)$

$p_1, p_2, p$  are **entity-variables**, i.e., occur in predicate arguments ranging over, whereas  $n_1, n_2, f_1, f_2, f$  are **value-variables**, i.e., occur in predicate arguments ranging over values

Database  $\mathbf{D}$

John Doe  $\xleftarrow{Name}$  **Doe<sub>1</sub>**  $\xrightarrow{HPhone}$  358

Johnny Doe  $\xleftarrow{Name}$  **Doe<sub>2</sub>**  $\xrightarrow{HPhone}$  635

Mary Doe  $\xleftarrow{Name}$  **Doe<sub>3</sub>**  $\xrightarrow{HPhone}$  358

Model we look for

# Example

Consider the following ontology  $\mathcal{O} = (\mathcal{T}, \mathbf{D})$

TBox  $\mathcal{T}$

( $r_1$ )  $Name(p_1, n_1) \wedge Name(p_2, n_2) \wedge JaccSim(n_1, n_2, 0.7) \rightarrow p_1 = p_2$

( $r_2$ )  $Name(p, n_1) \wedge Name(p, n_2) \rightarrow n_1 = n_2$

( $r_3$ )  $HPhone(p, f_1) \wedge HPhone(p, f_2) \rightarrow f_1 = f_2$

( $r_4$ )  $HPhone(p_1, f) \wedge HPhone(p_2, f) \rightarrow SameHouse(p_1, p_2)$

$p_1, p_2, p$  are **entity-variables**, i.e., occur in predicate arguments ranging over, whereas  $n_1, n_2, f_1, f_2, f$  are **value-variables**, i.e., occur in predicate arguments ranging over values

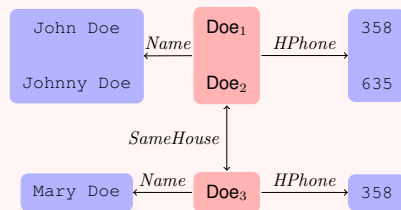
Database  $\mathbf{D}$

John Doe  $\xleftarrow{Name}$  **Doe<sub>1</sub>**  $\xrightarrow{HPhone}$  358

Johnny Doe  $\xleftarrow{Name}$  **Doe<sub>2</sub>**  $\xrightarrow{HPhone}$  635

Mary Doe  $\xleftarrow{Name}$  **Doe<sub>3</sub>**  $\xrightarrow{HPhone}$  358

Model we look for



# Semantics (informal)

To this aim, [Fagin et al., 2023] defines a new semantics such that:

- models, called **solutions**, use **equivalence classes of entities**, like  $[\mathbf{Doe}_1, \mathbf{Doe}_2]$ , and **sets of values**, like  $\{358, 635\}$
- TGDs and EGDs satisfaction and query evaluation are revised so that
  - (a) joins occur when there is a **non-empty intersection** of sets
  - (b) **TGDs propagate such intersection and queries return such intersection**. E.g., if we substitute  $(r_4)$  with

$$HPhone(p_1, f) \wedge HPhone(p_2, f) \rightarrow SameHouse(p_1, p_2, f)$$

solutions contain  $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\})$

If we ask for telephone number in common between  $\mathbf{Doe}_1$  and  $\mathbf{Doe}_3$ , e.g., we issue the query:

$$q(x) : \neg HPhone(\mathbf{Doe}_1, x) \wedge HPhone(\mathbf{Doe}_3, x)$$

we obtain  $\{358\}$  as answer

- (c) EGDs are satisfied when the **the sets to be equated are the same set**

## Semantics – Instances

We have countably infinite many **entity-nulls** and **value-nulls** (note that we consider generic TGDs)

Equivalence classes of entities and entity nulls are called **E-sets**

Non-empty sets of values and value nulls are called **V-sets**

### Definition (Instance)

An *instance*  $\mathcal{I}$  for an ontology  $\mathcal{O}$  w.r.t. an equivalence relation  $\sim$  is a set of facts  $P(T_1, \dots, T_n)$  such that  $P$  is a  $n$ -ary predicate in  $\mathcal{O}$ , and each  $T_i$  is either an **E-set** w.r.t.  $\sim$  or a **V-set**

### Example

- $(d_1)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe, Johnny Doe}\})$
- $(d_2)$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$
- $(d_3)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$
- $(d_4)$   $HPhone([\mathbf{Doe}_3], \{358\})$
- $(d_5)$   $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3])$
- $(d_6)$   $SameHouse([\mathbf{Doe}_3], [\mathbf{Doe}_1, \mathbf{Doe}_2])$

## Semantics – Instances

We have countably infinite many **entity-nulls** and **value-nulls** (note that we consider generic TGDs)

Equivalence classes of entities and entity nulls are called **E-sets**

Non-empty sets of values and value nulls are called **V-sets**

### Definition (Instance)

An *instance*  $\mathcal{I}$  for an ontology  $\mathcal{O}$  w.r.t. an equivalence relation  $\sim$  is a set of facts  $P(T_1, \dots, T_n)$  such that  $P$  is a  $n$ -ary predicate in  $\mathcal{O}$ , and each  $T_i$  is either an **E-set** w.r.t.  $\sim$  or a **V-set**

### Example

- $(d_1)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe, Johnny Doe}\})$
- $(d_2)$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$
- $(d_3)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$
- $(d_4)$   $HPhone([\mathbf{Doe}_3], \{358\})$
- $(d_5)$   $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3])$
- $(d_6)$   $SameHouse([\mathbf{Doe}_3], [\mathbf{Doe}_1, \mathbf{Doe}_2])$



## Definition (Assignment)

An *assignment from a conjunction of atoms  $\phi(\mathbf{x})$  to an instance  $\mathcal{I}$*  is a mapping  $\mu$  from the variables and values in  $\phi(\mathbf{x})$  to E-sets and V-sets of  $\mathcal{I}$  such that

- each entity-variable is mapped to an E-set
- **different occurrences of a value-variable** are mapped to V-sets with **non-empty intersection**
- each value  $v$  is mapped to a V-set  $V$ , such that  $v \in V$
- for each atom  $P(x, e, v)$  of  $\phi(\mathbf{x})$ , where  $x$  is a variable,  $e$  is an entity and  $v$  is a value,  $\mathcal{I}$  contains a fact of the form  $P(\mu(x), [e], \mu(v))$ ,

## Example

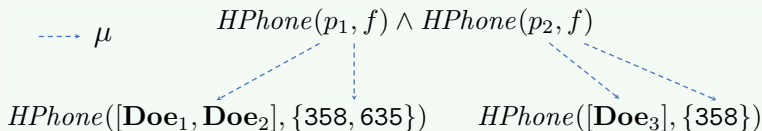
# Semantics – assignments

## Definition (Assignment)

An *assignment from a conjunction of atoms  $\phi(\mathbf{x})$  to an instance  $\mathcal{I}$*  is a mapping  $\mu$  from the variables and values in  $\phi(\mathbf{x})$  to E-sets and V-sets of  $\mathcal{I}$  such that

- each entity-variable is mapped to an E-set
- **different occurrences of a value-variable** are mapped to V-sets with **non-empty intersection**
- each value  $v$  is mapped to a V-set  $V$ , such that  $v \in V$
- for each atom  $P(x, e, v)$  of  $\phi(\mathbf{x})$ , where  $x$  is a variable,  $e$  is an entity and  $v$  is a value,  $\mathcal{I}$  contains a fact of the form  $P(\mu(x), [e], \mu(v))$ ,

## Example



# Semantics of TGDs

An instance  $\mathcal{I}$  for an ontology  $\mathcal{O}$  satisfies a TGD  $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y}))$  if for each assignment  $\mu$  from  $\phi(\mathbf{x})$  to  $\mathcal{I}$  there is an assignment  $\mu'$  from  $\psi(\mathbf{x}, \mathbf{y})$  to  $\mathcal{I}$  such that, for each  $x$  in  $\mathbf{x}$

- $\mu(x) = \mu'(x)$ , if  $x$  is an entity-variable
- the intersection of V-sets assigned by  $\mu$  to each occurrence of  $x$  in  $\phi(\mathbf{x})$  is contained in the intersection of V-sets assigned by  $\mu'$  to each occurrence of  $x$  in  $\psi(\mathbf{x}, \mathbf{y})$ , if  $x$  is a value-variable

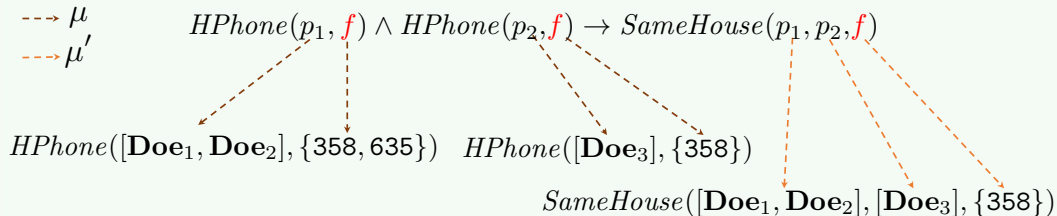
## Example

# Semantics of TGDs

An instance  $\mathcal{I}$  for an ontology  $\mathcal{O}$  satisfies a TGD  $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y}))$  if for each assignment  $\mu$  from  $\phi(\mathbf{x})$  to  $\mathcal{I}$  there is an assignment  $\mu'$  from  $\psi(\mathbf{x}, \mathbf{y})$  to  $\mathcal{I}$  such that, for each  $x$  in  $\mathbf{x}$

- $\mu(x) = \mu'(x)$ , if  $x$  is an entity-variable
- the intersection of V-sets assigned by  $\mu$  to each occurrence of  $x$  in  $\phi(\mathbf{x})$  is contained in the intersection of V-sets assigned by  $\mu'$  to each occurrence of  $x$  in  $\psi(\mathbf{x}, \mathbf{y})$ , if  $x$  is a value-variable

## Example



An instance  $\mathcal{I}$  for an ontology  $\mathcal{O}$  satisfies an EGD  $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow y = z)$  if each assignment  $\mu$  from  $\phi(\mathbf{x})$  to  $\mathcal{I}$  is such that  $\mu(y) = \mu(z)$  (i.e., all occurrences of  $y$  and  $z$  must be assigned with the same set)

## Example





# Solution of an Ontology

An instance  $\mathcal{I}$  is a **solution** for an ontology  $\mathcal{O} = (\mathcal{T}, \mathbf{D})$  if

- $\mathcal{I}$  satisfies all TGDs and EGDs in  $\mathcal{T}$
- for each ground atom in  $P(c_1, \dots, c_n) \in \mathbf{D}$  if there is a fact  $P(T_1, \dots, T_n)$  in  $\mathcal{I}$  such that  $c_i \in T_i$ , for all  $1 \leq i \leq n$ .

## Database $\mathbf{D}$

- $(g_1)$  *Name*(**Doe**<sub>1</sub>, John Doe)
- $(g_2)$  *Name*(**Doe**<sub>2</sub>, Johnny Doe)
- $(g_3)$  *HPhone*(**Doe**<sub>1</sub>, 358)
- $(g_4)$  *HPhone*(**Doe**<sub>2</sub>, 635)
- $(g_5)$  *Name*(**Doe**<sub>3</sub>, Mary Doe)
- $(g_6)$  *HPhone*(**Doe**<sub>3</sub>, 358)

## Solution

- $(d_1)$  *Name*([**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], {John Doe, Johnny Doe})
- $(d_2)$  *HPhone*([**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], {358, 635})
- $(d_3)$  *Name*([**Doe**<sub>3</sub>], {Mary Doe})
- $(d_4)$  *HPhone*([**Doe**<sub>3</sub>], {358})
- $(d_5)$  *SameHouse*([**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], [**Doe**<sub>3</sub>], {358})
- $(d_6)$  *SameHouse*([**Doe**<sub>3</sub>], [**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], {358})



# Queries

The *answer*  $q^{\mathcal{I}}$  to a CQ  $q(x_1, \dots, x_n)$  on an instance  $\mathcal{I}$  is the set of all tuples  $\langle T_1, \dots, T_n \rangle$  such that there is an assignment  $\mu$  from  $q$  to  $\mathcal{I}$  for which

- $T_i = \mu(x_i)$ , if  $x_i$  is an entity-variable
- $T_i$ , is the intersection of the V-sets assigned to the various occurrences of  $x_i$ , if  $x_i$  is a value-variable

## Example

$$q(x) : \neg HPhone(\mathbf{Doe}_1, x) \wedge HPhone(\mathbf{Doe}_3, x)$$

$$HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\}) \quad HPhone([\mathbf{Doe}_3], \{358\})$$

$$q^{\mathcal{I}} = \{\langle \{358\} \rangle\}$$

## Certain Answers

Let  $\mathbf{T} = \langle T_1, \dots, T_n \rangle$  and  $\mathbf{T}' = \langle T'_1, \dots, T'_n \rangle$  be two tuples of E-sets and V-sets, we say that  $\mathbf{T}'$  *dominates*  $\mathbf{T}$ , denoted  $\mathbf{T} \leq \mathbf{T}'$ , if  $T_i \subseteq T'_i$ , for all  $1 \leq i \leq n$ .

### Definition (certain-answers)

A null-free tuple  $\mathbf{T}$  of E-sets and V-sets is a *certain answer* to a CQ  $q$  w.r.t. an ontology  $\mathcal{O}$  if

- 1 for every solution  $\mathcal{I}$  for  $\mathcal{O}$ , there is a tuple  $\mathbf{T}' \in q^{\mathcal{I}}$  such that  $\mathbf{T} \leq \mathbf{T}'$
- 2 there is no null-free tuple  $\mathbf{T}'$  that satisfies (1) and  $\mathbf{T} < \mathbf{T}'$

We write  $\text{cert}(q, \mathcal{O})$  for the set of certain answers to  $q$  w.r.t.  $\mathcal{O}$

### Example

Given  $q(x) : -\text{HPhone}(\mathbf{Doe}_1, x)$ , then  $\text{cert}(q, \mathcal{O}) = \{\{\{358, 635\}\}\}$ .

Note that  $\{\{358\}\}$  and  $\{\{635\}\}$  are not certain answers because do not enjoy Condition 2

## Certain Answers

Let  $\mathbf{T} = \langle T_1, \dots, T_n \rangle$  and  $\mathbf{T}' = \langle T'_1, \dots, T'_n \rangle$  be two tuples of E-sets and V-sets, we say that  $\mathbf{T}'$  *dominates*  $\mathbf{T}$ , denoted  $\mathbf{T} \leq \mathbf{T}'$ , if  $T_i \subseteq T'_i$ , for all  $1 \leq i \leq n$ .

### Definition (certain-answers)

A null-free tuple  $\mathbf{T}$  of E-sets and V-sets is a *certain answer* to a CQ  $q$  w.r.t. an ontology  $\mathcal{O}$  if

- 1 for every solution  $\mathcal{I}$  for  $\mathcal{O}$ , there is a tuple  $\mathbf{T}' \in q^{\mathcal{I}}$  such that  $\mathbf{T} \leq \mathbf{T}'$
- 2 there is no null-free tuple  $\mathbf{T}'$  that satisfies (1) and  $\mathbf{T} < \mathbf{T}'$

We write  $\text{cert}(q, \mathcal{O})$  for the set of certain answers to  $q$  w.r.t.  $\mathcal{O}$

### Example

Given  $q(x) : -\text{HPhone}(\mathbf{Doe}_1, x)$ , then  $\text{cert}(q, \mathcal{O}) = \{\{\{358, 635\}\}\}$ .

Note that  $\{\{358\}\}$  and  $\{\{635\}\}$  are not certain answers because do not enjoy Condition 2

# Homomorphism

For an instance  $\mathcal{I}$ ,  $under(\mathcal{I})$  is the set of entities, entity-nulls, value and value-nulls in  $\mathcal{I}$ .

Let  $\mathcal{I}_1$  and  $\mathcal{I}_2$  be two instances. An **homomorphism**  $h : \mathcal{I}_1 \rightarrow \mathcal{I}_2$  is a structure-preserving mapping from  $under(\mathcal{I}_1)$  to  $under(\mathcal{I}_2)$

More formally:

## Definition (homomorphism)

- $h$  maps entities and values to themselves
- $h$  maps each entity-null to an entity or an entity-null, and each value-null to a value or a value-null
- for each  $P(T_1, \dots, T_n)$  in  $\mathcal{I}_1$ , there is  $P(U_1, \dots, U_n)$  in  $\mathcal{I}_2$  such that  $h(T_i) \subseteq U_i$  for each  $i$

## Definition (universal solution)

A solution  $\mathcal{U}$  for an ontology  $\mathcal{O}$  is *universal* if, for every solution  $\mathcal{I}$  for  $\mathcal{O}$ , there is a homomorphism  $h : \mathcal{U} \rightarrow \mathcal{I}$

*The solution we have used so far for our ongoing example is a universal solution.*

## Theorem

Let  $q$  be a CQ, let  $\mathcal{O}$  be an ontology, and let  $\mathcal{U}$  be a universal solution for  $\mathcal{O}$ . Then  $\text{cert}(q, \mathcal{O}) = q^{\mathcal{U}} \downarrow^{\rho}$ .

The operator  $\downarrow^{\rho}$  first (i) eliminates nulls occurring in  $q^{\mathcal{U}}$ , then (ii) eliminates tuples that are dominated by other tuples.

## Definition (universal solution)

A solution  $\mathcal{U}$  for an ontology  $\mathcal{O}$  is *universal* if, for every solution  $\mathcal{I}$  for  $\mathcal{O}$ , there is a homomorphism  $h : \mathcal{U} \rightarrow \mathcal{I}$

*The solution we have used so far for our ongoing example is a universal solution.*

## Theorem

Let  $q$  be a CQ, let  $\mathcal{O}$  be an ontology, and let  $\mathcal{U}$  be a universal solution for  $\mathcal{O}$ . Then  $\text{cert}(q, \mathcal{O}) = q^{\mathcal{U}} \downarrow^{\rho}$ .

The operator  $\downarrow^{\rho}$  first (i) eliminates nulls occurring in  $q^{\mathcal{U}}$ , then (ii) eliminates tuples that are dominated by other tuples.

## Definition (universal solution)

A solution  $\mathcal{U}$  for an ontology  $\mathcal{O}$  is *universal* if, for every solution  $\mathcal{I}$  for  $\mathcal{O}$ , there is a homomorphism  $h : \mathcal{U} \rightarrow \mathcal{I}$

*The solution we have used so far for our ongoing example is a universal solution.*

## Theorem

Let  $q$  be a CQ, let  $\mathcal{O}$  be an ontology, and let  $\mathcal{U}$  be a universal solution for  $\mathcal{O}$ . Then  $\text{cert}(q, \mathcal{O}) = q^{\mathcal{U}} \downarrow^{\rho}$ .

The operator  $\downarrow^{\rho}$  first (i) eliminates nulls occurring in  $q^{\mathcal{U}}$ , then (ii) eliminates tuples that are dominated by other tuples.

## Example

Let's consider the TBox

$$\begin{aligned}(r_3) \quad & HPhone(p, f_1) \wedge HPhone(p, f_2) \rightarrow f_1 = f_2 \\(r_5) \quad & HPhone(p, f) \rightarrow IsHPhoneOf(f, p)\end{aligned}$$

and the database **D** consisting of

$$\begin{aligned}HPhone(\mathbf{Doe}_1, 358) \\HPhone(\mathbf{Doe}_1, 635)\end{aligned}$$

The following instance  $\mathcal{U}$  is a universal solution

$$\begin{aligned}HPhone([\mathbf{Doe}_1], \{358, 635\}) \\IsHPhoneOf(\{358, 635\}, [\mathbf{Doe}_1])\end{aligned}$$

The set of certain answers to  $q(x, y) : \neg IsHPhoneOf(x, y)$  is  $cert(q, \mathcal{O}) = \{\{\{358, 635\}\}, [\mathbf{Doe}_1]\}$

Note that in this case we do not need to apply  $\downarrow^\rho$  to  $q^{\mathcal{U}}$



# Example

## Example

Let's consider the TBox

$$(r_3) \quad HPhone(p, f_1) \wedge HPhone(p, f_2) \rightarrow f_1 = f_2$$

$$(r_5) \quad HPhone(p, f) \rightarrow IsHPhoneOf(f, p)$$

and the database **D** consisting of

$$HPhone(\mathbf{Doe}_1, 358)$$

$$HPhone(\mathbf{Doe}_1, 635)$$

Even the following  $\mathcal{U}'$  is a universal solution, where  $n_1$  and  $n_2$  are value-nulls.

$$HPhone([\mathbf{Doe}_1], \{358, 635, n_1\})$$

$$IsHPhoneOf(\{358, 635, n_1\}, [\mathbf{Doe}_1])$$

$$IsHPhoneOf(\{358, n_2\}, [\mathbf{Doe}_1])$$

In this case:

$$q^{\mathcal{U}'} = \{\langle \{358, 635, n_1\}, [\mathbf{Doe}_1] \rangle, \langle \{358, n_2\}, [\mathbf{Doe}_1] \rangle\}$$

$$q^{\mathcal{U}'} \downarrow = \{\langle \{358, 635\}, [\mathbf{Doe}_1] \rangle, \langle \{358\}, [\mathbf{Doe}_1] \rangle\}$$

$$q^{\mathcal{U}'} \downarrow^\rho = \{\langle \{358, 635\}, [\mathbf{Doe}_1] \rangle = cert(q, \mathcal{O})\}$$

# Computing a Universal Solution

We use a tailored chase procedure that we show through an example

## Database **D**

$(g_1)$  *Name*(**Doe**<sub>1</sub>, John Doe)  
 $(g_2)$  *Name*(**Doe**<sub>2</sub>, Johnny Doe)  
 $(g_3)$  *HPhone*(**Doe**<sub>1</sub>, 358)  
 $(g_4)$  *HPhone*(**Doe**<sub>2</sub>, 635)  
 $(g_5)$  *Name*(**Doe**<sub>3</sub>, Mary Doe)  
 $(g_6)$  *HPhone*(**Doe**<sub>3</sub>, 358)

→

Starting instance  $\mathcal{I}_0$

## e-EGD chase rule

$(r_1) : \textit{Name}(p_1, n_1) \wedge \textit{Name}(p_2, n_2) \wedge \textit{JaccSim}(n_1, n_2, 0.7) \rightarrow p_1 = p_2$  is applicable because of  $(d_1), (d_2)$   
 $\mathcal{I}_1$  is obtained from  $\mathcal{I}_0$  by replacing **everywhere** [**Doe**<sub>1</sub>] and [**Doe**<sub>2</sub>] with their **union** [**Doe**<sub>1</sub>, **Doe**<sub>2</sub>].

$(d_7)$  *Name*([**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], {John Doe})     $(d_8)$  *Name*([**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], {Johnny Doe})  
 $(d_9)$  *HPhone*([**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], {358})     $(d_{10})$  *HPhone*([**Doe**<sub>1</sub>, **Doe**<sub>2</sub>], {635})  
 $(d_5)$  *Name*([**Doe**<sub>3</sub>], {Mary Doe})     $(d_6)$  *HPhone*([**Doe**<sub>3</sub>], {358})

# Computing a Universal Solution

We use a tailored chase procedure that we show through an example

## Database $D$

$(g_1)$   $Name(\mathbf{Doe}_1, \text{John Doe})$   
 $(g_2)$   $Name(\mathbf{Doe}_2, \text{Johnny Doe})$   
 $(g_3)$   $HPhone(\mathbf{Doe}_1, 358)$   
 $(g_4)$   $HPhone(\mathbf{Doe}_2, 635)$   
 $(g_5)$   $Name(\mathbf{Doe}_3, \text{Mary Doe})$   
 $(g_6)$   $HPhone(\mathbf{Doe}_3, 358)$

→

## Starting instance $\mathcal{I}_0$

$(d_1)$   $Name([\mathbf{Doe}_1], \{\text{John Doe}\})$   
 $(d_2)$   $Name([\mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_3)$   $HPhone([\mathbf{Doe}_1], \{358\})$   
 $(d_4)$   $HPhone([\mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$   
 $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

## e-EGD chase rule

$(r_1) : Name(p_1, n_1) \wedge Name(p_2, n_2) \wedge JaccSim(n_1, n_2, 0.7) \rightarrow p_1 = p_2$  is applicable because of  $(d_1), (d_2)$   
 $\mathcal{I}_1$  is obtained from  $\mathcal{I}_0$  by replacing **everywhere**  $[\mathbf{Doe}_1]$  and  $[\mathbf{Doe}_2]$  with their **union**  $[\mathbf{Doe}_1, \mathbf{Doe}_2]$ .

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$   $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_9)$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358\})$   $(d_{10})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$   $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

# Computing a Universal Solution

We use a tailored chase procedure that we show through an example

## Database $D$

$(g_1)$   $Name(\mathbf{Doe}_1, \text{John Doe})$   
 $(g_2)$   $Name(\mathbf{Doe}_2, \text{Johnny Doe})$   
 $(g_3)$   $HPhone(\mathbf{Doe}_1, 358)$   
 $(g_4)$   $HPhone(\mathbf{Doe}_2, 635)$   
 $(g_5)$   $Name(\mathbf{Doe}_3, \text{Mary Doe})$   
 $(g_6)$   $HPhone(\mathbf{Doe}_3, 358)$

→

## Starting instance $\mathcal{I}_0$

$(d_1)$   $Name([\mathbf{Doe}_1], \{\text{John Doe}\})$   
 $(d_2)$   $Name([\mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_3)$   $HPhone([\mathbf{Doe}_1], \{358\})$   
 $(d_4)$   $HPhone([\mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$   
 $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

## e-EGD chase rule

$(r_1) : Name(p_1, n_1) \wedge Name(p_2, n_2) \wedge JaccSim(n_1, n_2, 0.7) \rightarrow p_1 = p_2$  is applicable because of  $(d_1), (d_2)$   
 $\mathcal{I}_1$  is obtained from  $\mathcal{I}_0$  by replacing **everywhere**  $[\mathbf{Doe}_1]$  and  $[\mathbf{Doe}_2]$  with their **union**  $[\mathbf{Doe}_1, \mathbf{Doe}_2]$ .

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$   $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_9)$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358\})$   $(d_{10})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$   $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

# Computing a Universal Solution

We use a tailored chase procedure that we show through an example

## Database $D$

$(g_1)$   $Name(\mathbf{Doe}_1, \text{John Doe})$   
 $(g_2)$   $Name(\mathbf{Doe}_2, \text{Johnny Doe})$   
 $(g_3)$   $HPhone(\mathbf{Doe}_1, 358)$   
 $(g_4)$   $HPhone(\mathbf{Doe}_2, 635)$   
 $(g_5)$   $Name(\mathbf{Doe}_3, \text{Mary Doe})$   
 $(g_6)$   $HPhone(\mathbf{Doe}_3, 358)$

→

## Starting instance $\mathcal{I}_0$

$(d_1)$   $Name([\mathbf{Doe}_1], \{\text{John Doe}\})$   
 $(d_2)$   $Name([\mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_3)$   $HPhone([\mathbf{Doe}_1], \{358\})$   
 $(d_4)$   $HPhone([\mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$   
 $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

## e-EGD chase rule

$(r_1) : Name(p_1, n_1) \wedge Name(p_2, n_2) \wedge JaccSim(n_1, n_2, 0.7) \rightarrow p_1 = p_2$  is applicable because of  $(d_1), (d_2)$   
 $\mathcal{I}_1$  is obtained from  $\mathcal{I}_0$  by replacing **everywhere**  $[\mathbf{Doe}_1]$  and  $[\mathbf{Doe}_2]$  with their **union**  $[\mathbf{Doe}_1, \mathbf{Doe}_2]$ .

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$   $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_9)$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358\})$   $(d_{10})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$   $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

# Computing a Universal Solution

instance  $\mathcal{I}_1$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$     $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_9)$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358\})$     $(d_{10})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$     $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

v-EGD chase rule

$(r_3) : HPhone(p, f_1) \wedge HPhone(p, f_2) \rightarrow f_1 = f_2$  is applicable because of  $(d_9), (d_{10})$   
 $\mathcal{I}_2$  is obtained from  $\mathcal{I}_1$  by replacing in  $d_9$  and  $d_{10}$   $\{358\}$  and  $\{635\}$  with their **union**  $\{358, 635\}$ .

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$     $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$     $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

# Computing a Universal Solution

instance  $\mathcal{I}_1$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$     $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_9)$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358\})$     $(d_{10})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$     $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

v-EGD chase rule

$(r_3) : HPhone(p, f_1) \wedge HPhone(p, f_2) \rightarrow f_1 = f_2$  is applicable because of  $(d_9), (d_{10})$   
 $\mathcal{I}_2$  is obtained from  $\mathcal{I}_1$  by replacing in  $d_9$  and  $d_{10}$   $\{358\}$  and  $\{635\}$  with their **union**  $\{358, 635\}$ .

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$     $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$     $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

# Computing a Universal Solution

instance  $\mathcal{I}_2$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

TGD chase rule

$(r'_4) : HPhone(p_1, f) \wedge HPhone(p_2, f) \rightarrow SameHouse(p_1, p_2, f)$  is applicable because of  $(d_{11}), (d_6)$ .  
 $\mathcal{I}_3$  is obtained by **adding**  $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\})$  to  $\mathcal{I}_2$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$   
 $(d_{12})$   $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\})$



# Computing a Universal Solution

instance  $\mathcal{I}_2$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

TGD chase rule

$(r'_4) : HPhone(p_1, f) \wedge HPhone(p_2, f) \rightarrow SameHouse(p_1, p_2, f)$  is applicable because of  $(d_{11}), (d_6)$ .  
 $\mathcal{I}_3$  is obtained by **adding**  $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\})$  to  $\mathcal{I}_2$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$   
 $(d_{12})$   $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\})$

# Introducing nulls because of a TGD chase rule

instance  $\mathcal{I}_2$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358\ 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

TGD chase rule

$(r''_4) : HPhone(p_1, f) \wedge HPhone(p_2, f) \rightarrow \exists a. SameHouse(p_1, p_2, f, a)$  is applicable because of  $(d_{11}), (d_6)$ .

$\mathcal{I}'_3$  is obtained by **adding**  $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\}, \{n\})$  to  $\mathcal{I}_2$ , with  $n$  a fresh value-null

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$   
 $(d_{12})$   $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\}, \{n\})$

# Introducing nulls because of a TGD chase rule

instance  $\mathcal{I}_2$

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358\ 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$

TGD chase rule

$(r''_4) : HPhone(p_1, f) \wedge HPhone(p_2, f) \rightarrow \exists a. SameHouse(p_1, p_2, f, a)$  is applicable because of  $(d_{11}), (d_6)$ .

$\mathcal{I}'_3$  is obtained by **adding**  $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\}, \{n\})$  to  $\mathcal{I}_2$ , with  $n$  a fresh value-null

$(d_7)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{John Doe}\})$      $(d_8)$   $Name([\mathbf{Doe}_1, \mathbf{Doe}_2], \{\text{Johnny Doe}\})$   
 $(d_{11})$   $HPhone([\mathbf{Doe}_1, \mathbf{Doe}_2], \{358, 635\})$   
 $(d_5)$   $Name([\mathbf{Doe}_3], \{\text{Mary Doe}\})$      $(d_6)$   $HPhone([\mathbf{Doe}_3], \{358\})$   
 $(d_{12})$   $SameHouse([\mathbf{Doe}_1, \mathbf{Doe}_2], [\mathbf{Doe}_3], \{358\}, \{n\})$

## Terminating chase procedure

The chase procedure terminates when it produces an **instance for which no rule in the TBox is applicable**. One such sequence  $\sigma = \mathcal{I}_0, \mathcal{I}_1, \dots, \mathcal{I}_k$  is called a **finite chase** and the **result of the chase,  $\text{chase}(\mathcal{O}, \sigma)$ , coincides with  $\mathcal{I}_k$**

### Theorem

If  $\mathcal{O}$  is an ontology and  $\sigma$  is a finite chase for  $\mathcal{O}$ , then  $\text{chase}(\mathcal{O}, \sigma)$  is a universal solution for  $\mathcal{O}$ .

*An easy case in which the chase terminates is when all TGDs in the TBox are **full** (no existentially quantified variables in the head of TGDs)*

Identifying sufficient conditions for (all instance) chase termination is left for future study

# Non-terminating chase procedure

Obviously, the chase may not terminate. A classical definition [Beeri and Vardi, 1984] for the result of the chase for an infinite chase sequence  $\sigma = \mathcal{I}_0, \mathcal{I}_1, \dots$  in the presence of TGDs and EGDs is:

$$\text{chase}(\mathcal{O}, \sigma) = \{f \mid \text{there is some } i \geq 0 \text{ such that } f \in \mathcal{I}_j \text{ for each } j \geq i\}.$$

## Example

Let  $\mathcal{O} = \langle \mathcal{T}, \mathbf{D} \rangle$ , where  $\mathbf{D} = \{P(1, 2)\}$  and  $\mathcal{T}$  consists of the two rules

$$(r_1) \quad P(x, y) \rightarrow \exists z. P(y, z) \qquad (r_2) \quad P(x, y) \wedge P(y, z) \rightarrow y = z$$

We repeatedly apply the above rules in the following order:  $r_1, r_1, r_2, r_1, r_2, r_1, r_2, r_1 \dots$

$$\begin{aligned} \mathcal{I}_0 &= \{P(\{1\}, \{2\})\} \\ \mathcal{I}_1 &= \{P(\{1\}, \{2\}), P(\{2\}, \{n_1\})\} \\ \mathcal{I}_2 &= \{P(\{1\}, \{2\}), P(\{2\}, \{n_1\}), P(\{n_1\}, \{n_2\})\} \\ \mathcal{I}_3 &= \{P(\{1\}, \{2, n_1\}), P(\{2, n_1\}, \{2, n_1\}), P(\{n_1\}, \{n_2\})\} \\ &\dots \\ \mathcal{I}_7 &= \{P(\{1\}, \{2, n_1, n_2\}), P(\{2, n_1\}, \{2, n_1, n_2\}), P(\{2, n_1, n_2\}, \{2, n_1, n_2\}), P(\{n_2\}, \{n_3\}), \\ &\quad P(\{n_3\}, \{n_4\})\} \\ &\dots \end{aligned}$$

Thus  $\text{chase}(\mathcal{O}, \sigma) = \emptyset$ . but if we apply  $r_1$  and then  $r_2$  we have a finite chase sequence such that  $\text{chase}(\mathcal{O}, \sigma) = \{P(\{1\}, \{2, n_1\}), P(\{2, n_1\}, \{2, n_1\})\}$

# Non-terminating chase procedure

Obviously, the chase may not terminate. A classical definition [Beeri and Vardi, 1984] for the result of the chase for an infinite chase sequence  $\sigma = \mathcal{I}_0, \mathcal{I}_1, \dots$  in the presence of TGDs and EGDs is:

$$\text{chase}(\mathcal{O}, \sigma) = \{f \mid \text{there is some } i \geq 0 \text{ such that } f \in \mathcal{I}_j \text{ for each } j \geq i\}.$$

## Example

Let  $\mathcal{O} = \langle \mathcal{T}, \mathbf{D} \rangle$ , where  $\mathbf{D} = \{P(1,2)\}$  and  $\mathcal{T}$  consists of the two rules

$$(r_1) \quad P(x, y) \rightarrow \exists z. P(y, z) \qquad (r_2) \quad P(x, y) \wedge P(y, z) \rightarrow y = z$$

We repeatedly apply the above rules in the following order:  $r_1, r_1, r_2, r_1, r_2, r_1, r_2, r_1 \dots$

$$\begin{aligned} \mathcal{I}_0 &= \{P(\{1\}, \{2\})\} \\ \mathcal{I}_1 &= \{P(\{1\}, \{2\}), P(\{2\}, \{n_1\})\} \\ \mathcal{I}_2 &= \{P(\{1\}, \{2\}), P(\{2\}, \{n_1\}), P(\{n_1\}, \{n_2\})\} \\ \mathcal{I}_3 &= \{P(\{1\}, \{2, n_1\}), P(\{2, n_1\}, \{2, n_1\}), P(\{n_1\}, \{n_2\})\} \\ &\dots \\ \mathcal{I}_7 &= \{P(\{1\}, \{2, n_1, n_2\}), P(\{2, n_1\}, \{2, n_1, n_2\}), P(\{2, n_1, n_2\}, \{2, n_1, n_2\}), P(\{n_2\}, \{n_3\}), \\ &\quad P(\{n_3\}, \{n_4\})\} \\ &\dots \end{aligned}$$

Thus  $\text{chase}(\mathcal{O}, \sigma) = \emptyset$ . but if we apply  $r_1$  and then  $r_2$  we have a finite chase sequence such that

$$\text{chase}(\mathcal{O}, \sigma) = \{P(\{1\}, \{2, n_1\}), P(\{2, n_1\}, \{2, n_1\})\}$$

# ICAR properties

Consider an entity-EGD  $\phi(\mathbf{x}) \rightarrow y = z$  and an instance  $\mathcal{I}$ : if there is an assignment  $\mu$  from  $\phi(\mathbf{x})$  to  $\mathcal{I}$  we can say that  $\mu(y)$  matches with  $\mu(z)$  (i.e.  $\mu(y) \approx \mu(z)$  )

Merges are then obtained through set-union (i.e.  $\langle \mu(y), \mu(z) \rangle = \mu(y) \cup \mu(z)$ ).

Our match and merge functions satisfy the following properties defined in [Benjelloun et al., 2009]

(Idempotence)  $\forall r : r \approx r$  and  $\langle r, r \rangle = r$ .

*An equivalence class always matches itself (even without rules), and  $E \cup E = E$*

(Associativity)  $\forall r_1, r_2, r_3$  such that  $\langle r_1, \langle r_2, r_3 \rangle \rangle$  and  $\langle \langle r_1, r_2 \rangle, r_3 \rangle$  exist, then  $\langle r_1, \langle r_2, r_3 \rangle \rangle = \langle \langle r_1, r_2 \rangle, r_3 \rangle$ .

*Of course, the union function is associative*

(Representativity) If  $r_3 = \langle r_1, r_2 \rangle$  then for any  $r_4$  such that  $r_1 \approx r_4$ , we also have  $r_3 \approx r_4$ .

*Consider an entity-EGD  $R(y, x) \wedge S(z, x) \rightarrow y = z$  and two facts  $R([e_1], \{1, 2\})$ ,  $S([e_4], \{2, 3\})$  of  $\mathcal{I}$ . There is an assignment  $\mu$  from the rule to  $\mathcal{I}$  such that  $\mu(y) = [e_1]$  and  $\mu(z) = [e_4]$ , i.e.  $[e_1] \approx [e_4]$ . Assume that  $[e_1]$  is merged with  $[e_2]$  because of another entity-EGD, i.e.  $\langle [e_1], [e_2] \rangle = [e_1, e_2]$ . This means that the above facts become  $R([e_1, e_2], \{1, 2\})$ ,  $S([e_4], \{2, 3\})$ . Therefore  $\langle [e_1, e_2], [e_4] \rangle$ .*

(Commutativity)  $\forall r_1, r_2 : r_1 \approx r_2$  if and only if  $r_2 \approx r_1$  and if  $r_1 \approx r_2$  then  $\langle r_1, r_2 \rangle = \langle r_2, r_1 \rangle$ .

$\mu(y) \approx \mu(z)$  does not imply that there is  $\mu'$  such that  $\mu'(z) \approx \mu'(y)$ .

$$R(y, x) \wedge S(z, x) \rightarrow y = z$$

the existence of  $R([e_1], \{1, 2\}), S([e_2], \{2, 3\})$  does not imply the existence of  $R([e_2], U_1), S([e_1], U_2)$  with  $U_1 \cap U_2 \neq \emptyset$ .

Satisfying the IAR properties is enough to guarantee that the application order of rules is not relevant.



# LACE+: Combining Global and Local Merges in Logic-based Entity Resolution

The paper [Bienvenu et al., 2023] presents the LACE+ framework, which extends the work in [Bienvenu et al., 2022] to consider local and global merges in the same spirit of [Fagin et al., 2023]

The two frameworks have been developed independently

## *Major differences*

- In LACE+ (as in LACE) there is a distinction between hard and soft entity resolution rules
- LACE+ combines ER rules with **denial constraints**, while our framework combines ER rules with TGDs
- [Bienvenu et al., 2022, Bienvenu et al., 2023] considers the complexity of various problems such as the existence of solutions and deciding whether a merge is certain or not, while [Fagin et al., 2023] focuses on the chase and on the query answering problem

## Main Problems left open in [Fagin et al., 2023]

- identifying a “good” notion of the result of the chase when the chase procedure does not terminate (applying a classical definition of the infinite chase by [Beeri and Vardi, 1984] does not work in our case)
- identifying structural conditions on the tgds and the egds of the TBox that **guarantee termination** of the chase procedure, possibly **in polynomial time** and, thus, yield tractable conjunctive query answering
- Enriching the framework with other kinds of axioms, e.g., denial constraints, as in the work by [Bienvenu et al., 2022, Bienvenu et al., 2023]

*Though the contribution is so far mainly conceptual, the framework presented in [Fagin et al., 2023] makes it possible to infuse ER into various areas, such as data exchange, data integration, ontology-based Data Access, in a principled way.*

- [Arasu et al., 2009] Arasu, A., Ré, C., and Suciu, D. (2009).  
Large-scale deduplication with constraints using dedupalog.  
*In Proc. of the 25th IEEE Int. Conf. on Data Engineering (ICDE 2009)*, pages 952–963.
- [Beeri and Vardi, 1984] Beeri, C. and Vardi, M. Y. (1984).  
A proof procedure for data dependencies.  
*J. of the ACM*, 31(4):718–741.
- [Benjelloun et al., 2009] Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S. E., and Widom, J. (2009).  
Swoosh: a generic approach to entity resolution.  
*Very Large Database J.*, 18(1):255–276.
- [Bertossi et al., 2013] Bertossi, L. E., Kolahi, S., and Lakshmanan, L. V. S. (2013).  
Data cleaning and query answering with matching dependencies and matching functions.  
*Theoretical Computer Science*, 52(3):441–482.

- [Bhattacharya and Getoor, 2007] Bhattacharya, I. and Getoor, L. (2007).  
Collective entity resolution in relational data.  
*ACM Trans. Knowl. Discov. Data*, 1(1):5.
- [Bienvenu et al., 2022] Bienvenu, M., Cima, G., and Gutiérrez-Basulto, V. (2022).  
LACE: A logical approach to collective entity resolution.  
In *Proc. of the 41st ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2022)*, pages 379–391.
- [Bienvenu et al., 2023] Bienvenu, M., Cima, G., Gutiérrez-Basulto, V., and Ibáñez-García, Y. (2023).  
Combining global and local merges in logic-based entity resolution.  
In *Proc. of the 20th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2023)*.
- [Chaudhuri et al., 2005] Chaudhuri, S., Ganti, V., and Motwani, R. (2005).  
Robust identification of fuzzy duplicates.  
In *Proc. of the 21st IEEE Int. Conf. on Data Engineering (ICDE 2005)*, pages 865–876.

- [Chaudhuri et al., 2007] Chaudhuri, S., Sarma, A. D., Ganti, V., and Kaushik, R. (2007).  
Leveraging aggregate constraints for deduplication.  
*In Proc. of the 26th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2007)*, pages 437–448.
- [Christen, 2012] Christen, P. (2012).  
A survey of indexing techniques for scalable record linkage and deduplication.  
*IEEE Trans. on Knowledge and Data Engineering*, 24(9):1537–1555.
- [Elfeky et al., 2002] Elfeky, M. G., Elmagarmid, A. K., and Verykios, V. S. (2002).  
TAILOR: A record linkage tool box.  
*In Proc. of the 18th IEEE Int. Conf. on Data Engineering (ICDE 2002)*, pages 17–28.
- [Fagin et al., 2023] Fagin, R., Kolaitis, P. G., Lembo, D., Popa, L., and Scafoglieri, F. (2023).  
A framework for combining entity resolution and query answering in knowledge bases.  
*In Proc. of the 20th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2023)*, pages 229–239.

[Fan, 2008] Fan, W. (2008).

Dependencies revisited for improving data quality.

In *Proc. of the 27th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2008)*, pages 159–170.

[Fellegi and Sunter, 1969] Fellegi, I. and Sunter, A. (1969).

A theory for record linkage.

*J. American Statistical Association*, 64:1183—1210.

[Guha et al., 2004] Guha, S., Koudas, N., Marathe, A., and Srivastava, D. (2004).

Merging the results of approximate match operations.

In *Proc. of the 30th Int. Conf. on Very Large Data Bases (VLDB 2004)*, pages 636–647.

[Guha et al., 2000] Guha, S., Rastogi, R., and Shim, K. (2000).

ROCK: A robust clustering algorithm for categorical attributes.

*Inf. Syst.*, 25(5):345–366.

- [Köpcke et al., 2010] Köpcke, H., Thor, A., and Rahm, E. (2010).  
Evaluation of entity resolution approaches on real-world match problems.  
*Proc. of the 36th Int. Conf. on Very Large Data Bases (VLDB 2010)*, 3(1):484–493.
- [Kouki et al., 2019] Kouki, P., Pujara, J., Marcum, C., Koehly, L. M., and Getoor, L. (2019).  
Collective entity resolution in multi-relational familial networks.  
*Knowl. Inf. Syst.*, 61(3):1547–1581.
- [Li et al., 2015] Li, L., Li, J., and Gao, H. (2015).  
Rule-based method for entity resolution.  
*IEEE Trans. on Knowledge and Data Engineering*, 27(1):250–263.
- [Newcombe et al., 1959] Newcombe, H., Kennedy, J., Axford, S., and James, A. (1959).  
Automatic linkage of vital records.  
*Science*, 130:954–959.

[Papadakis et al., 2021] Papadakis, G., Ioannou, E., Thanos, E., and Palpanas, T. (2021).

*The Four Generations of Entity Resolution.*

Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

[Singla and Domingos, 2006] Singla, P. and Domingos, P. M. (2006).

Entity resolution with markov logic.

In *6th IEEE Int. Conf. on Data Mining (ICDM 2006)*, pages 572—582.

[Wu et al., 2020] Wu, R., Chaba, S., Sawlani, S., Chu, X., and Thirumuruganathan, S. (2020).

Zeroer: Entity resolution using zero labeled examples.

In Maier, D., Pottinger, R., Doan, A., Tan, W., Alawini, A., and Ngo, H. Q., editors, *Proc. of the 2020 Int. Conf. on Management of Data (SIGMOD 2020)*, pages 1149–1164. ACM.