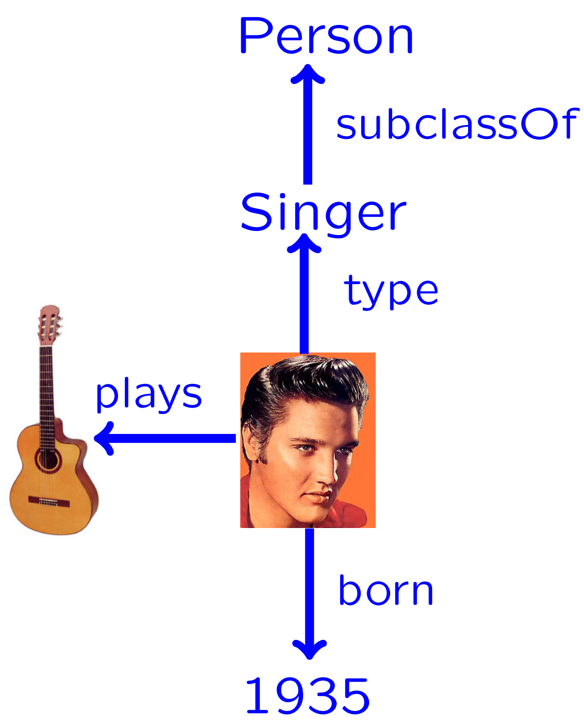
PARIS: Probabilistic Alignment of Relations, Instances, and Schema

Fabian M. Suchanek (Max Planck Institute for Informatics)
Serge Abiteboul (INRIA Saclay, Webdam team)
Pierre Senellart (TIcom ParisTech)

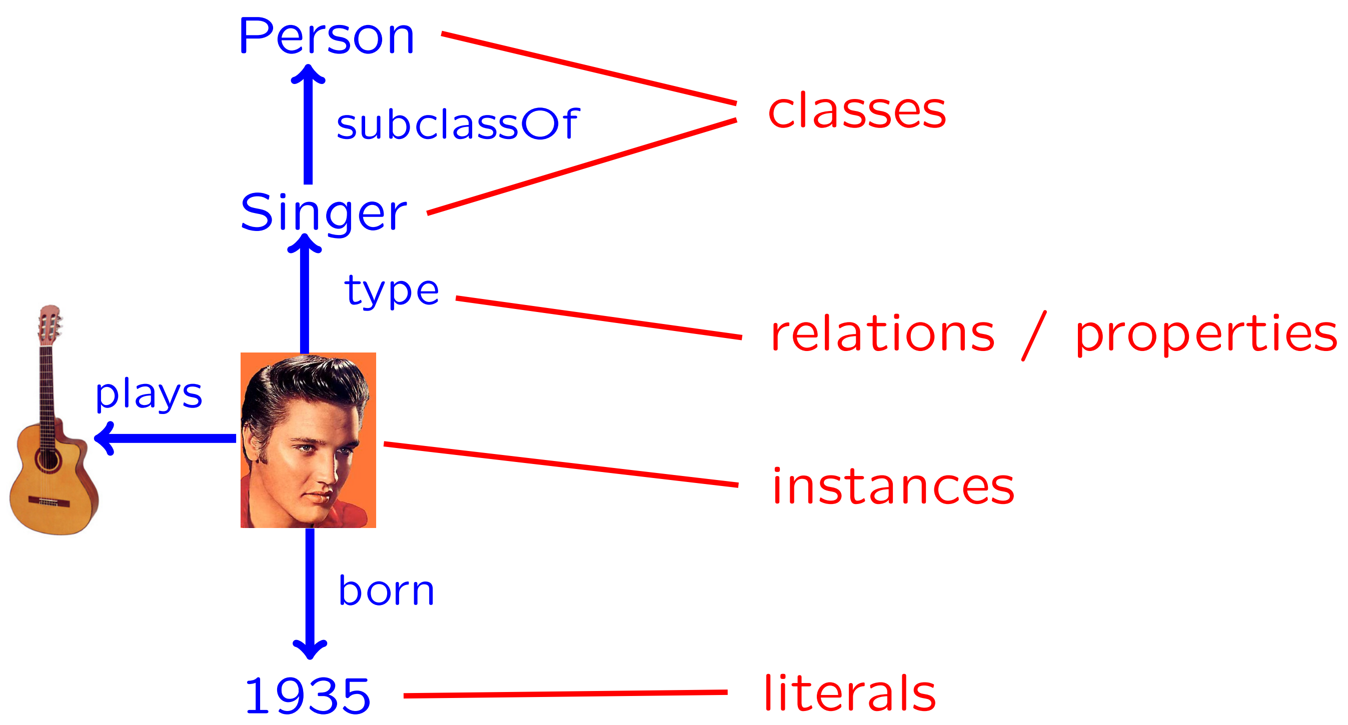
RDF Ontologies

An RDF ontology can be seen as a graph of entities



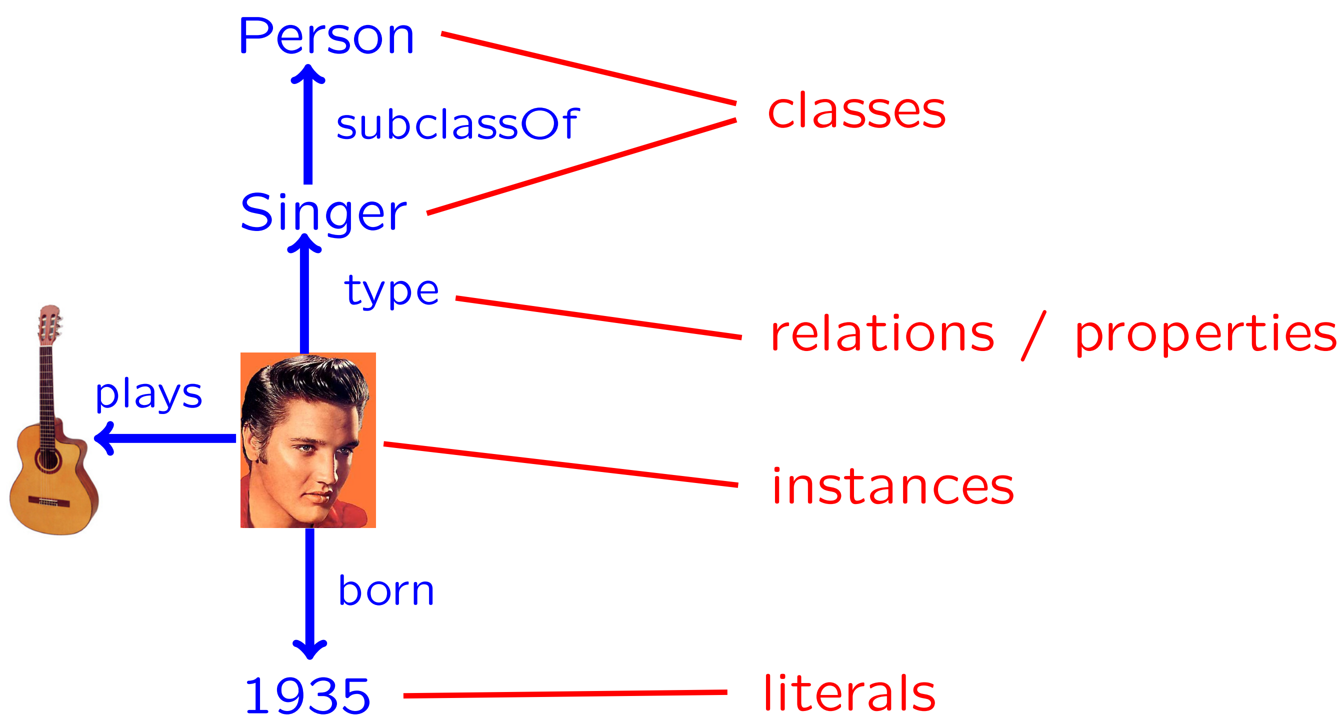
RDF Ontologies

An RDF ontology can be seen as a graph of entities



RDF Ontologies

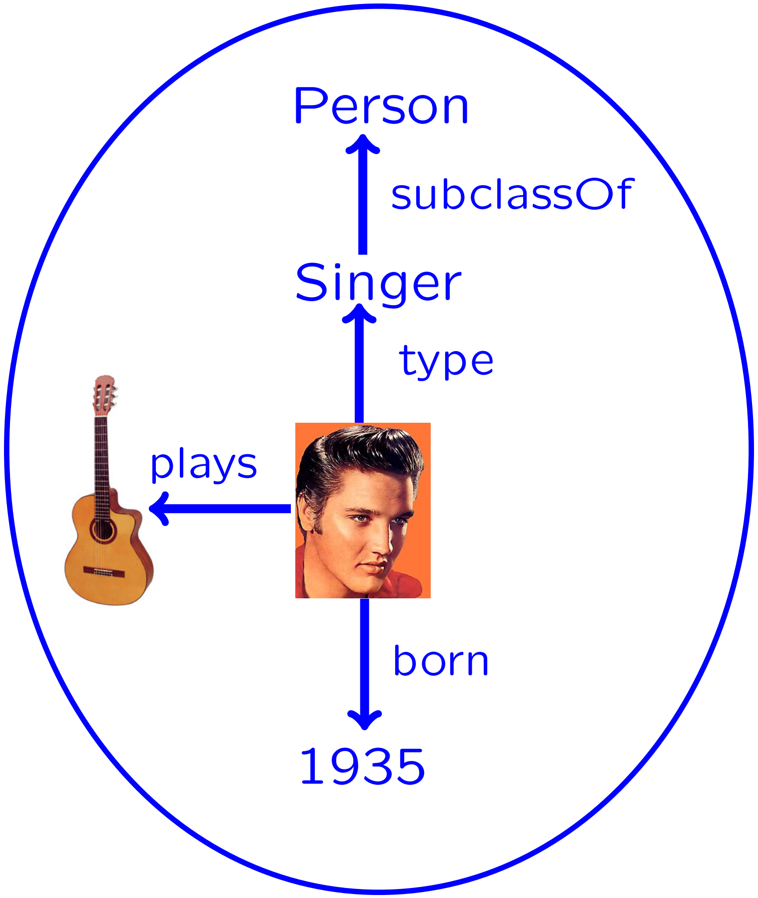
An RDF ontology can be seen as a graph of entities



Ontologies serve all kinds of purposes:
intelligent search, QA, machine translation...

Existing Ontologies

There are literally hundreds of ontologies on the Web



Freebase

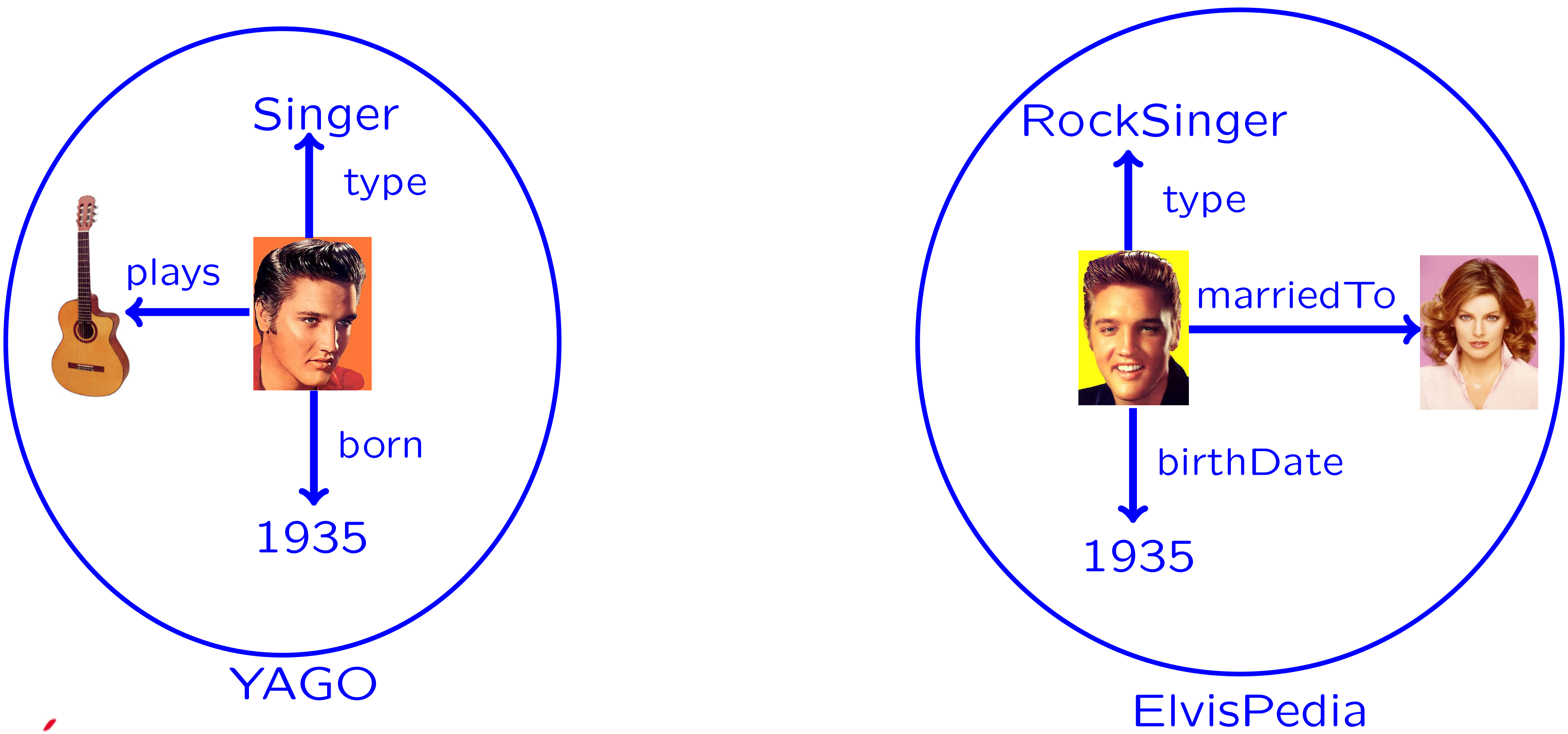
DBpedia

YAGO

UniProt

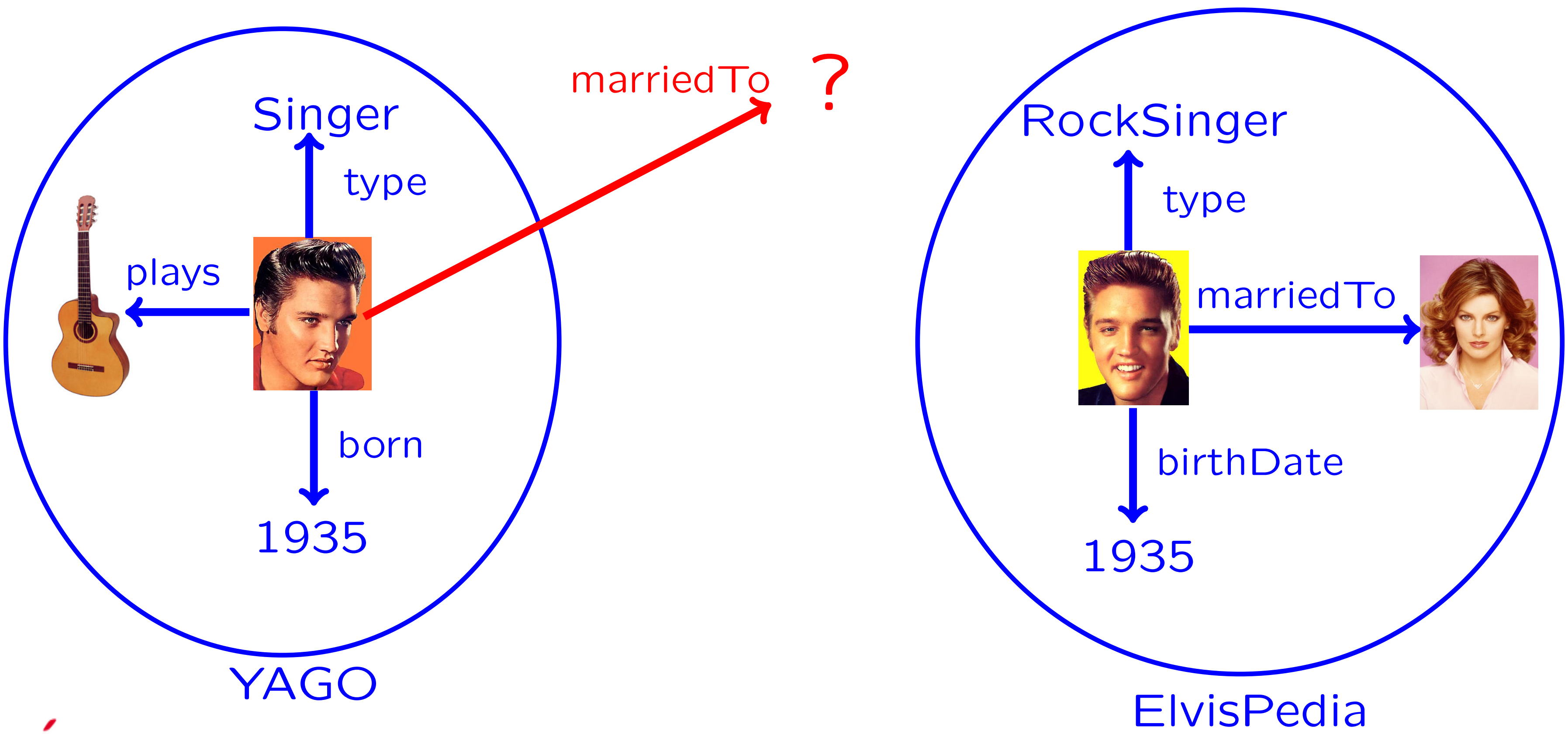
Overlapping Data

Many ontologies contain similar
or overlapping entities and facts



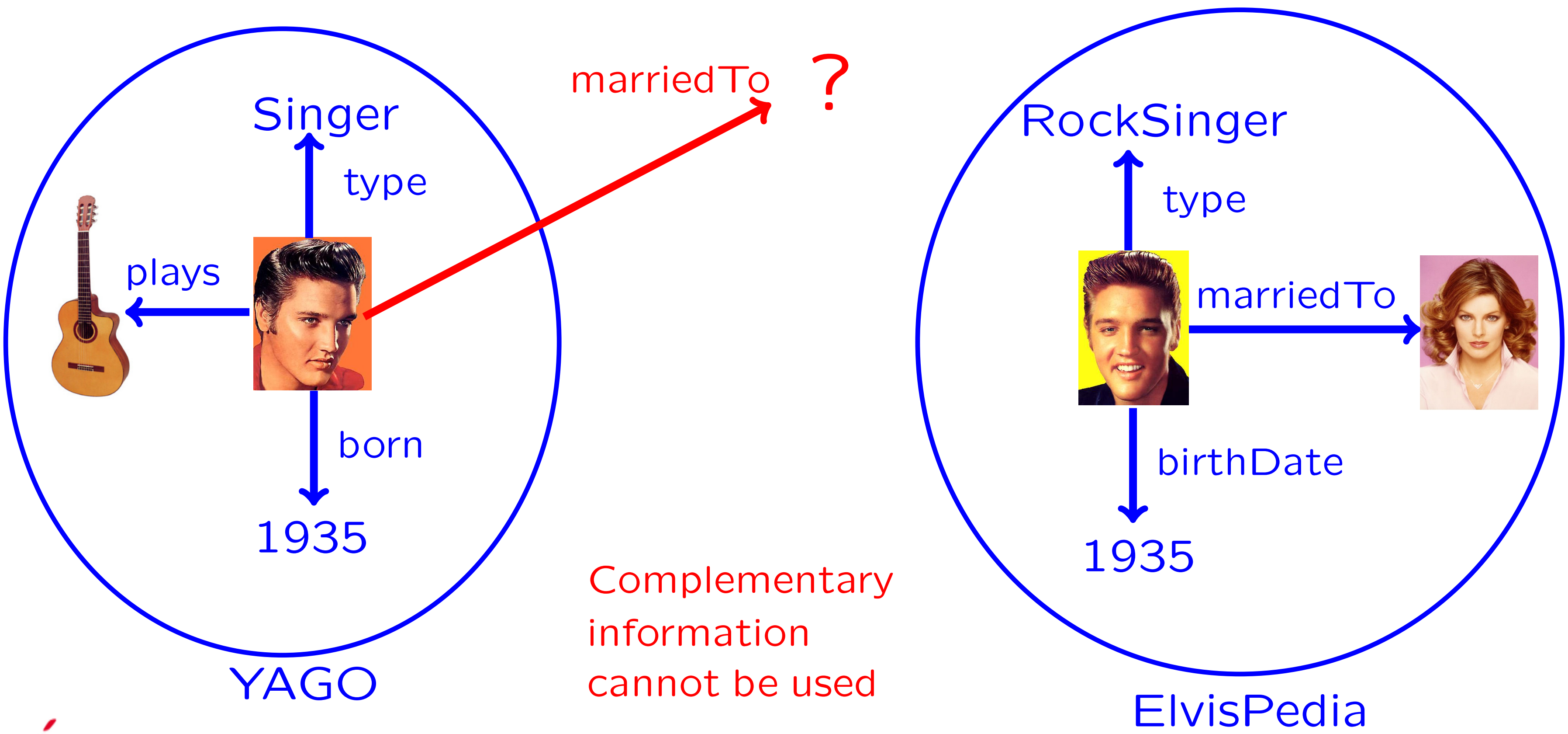
Problem: Elvis is lonely

Who is the spouse
of the guitar player?

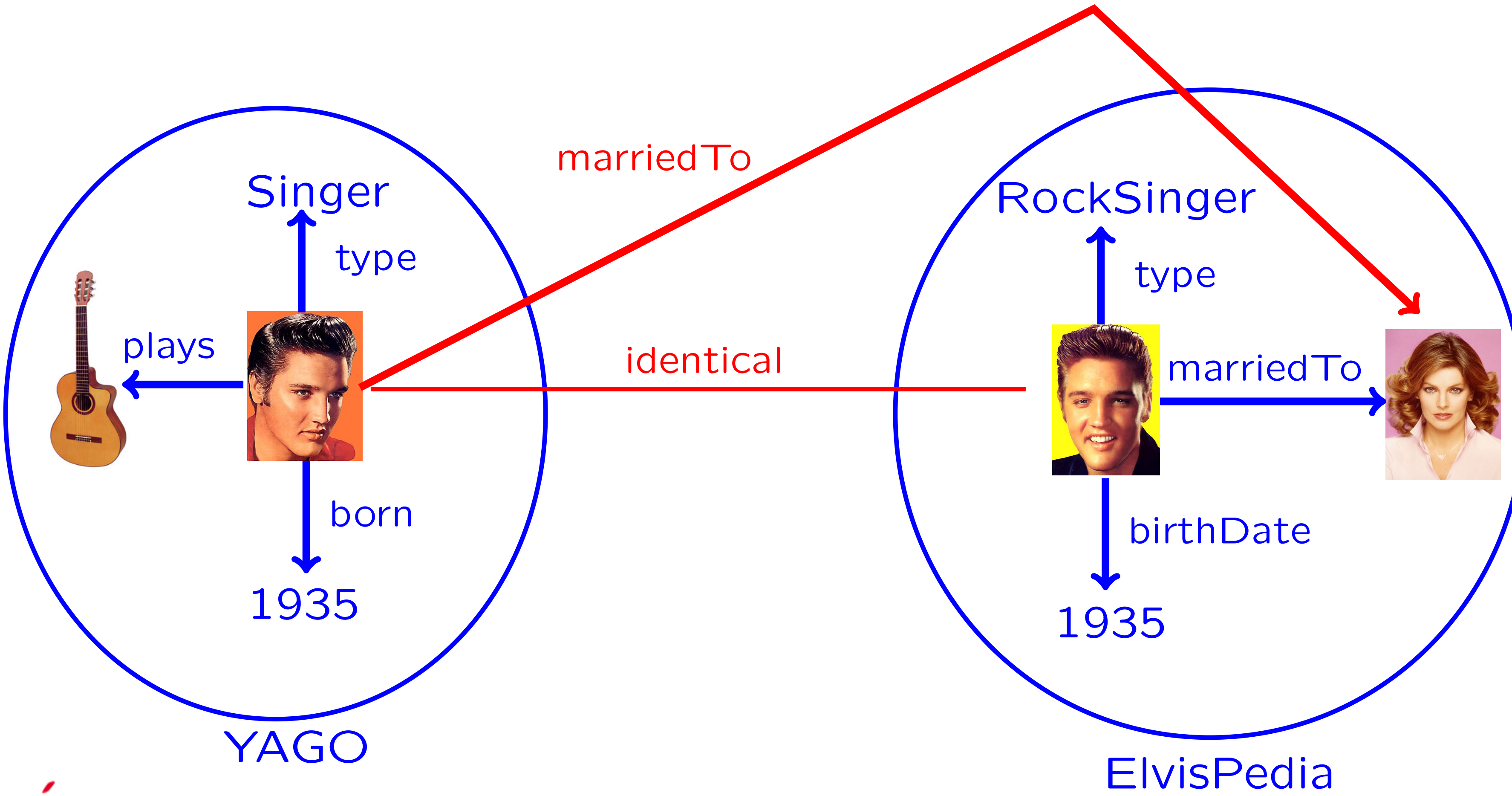


Problem: Elvis is lonely

Who is the spouse
of the guitar player?

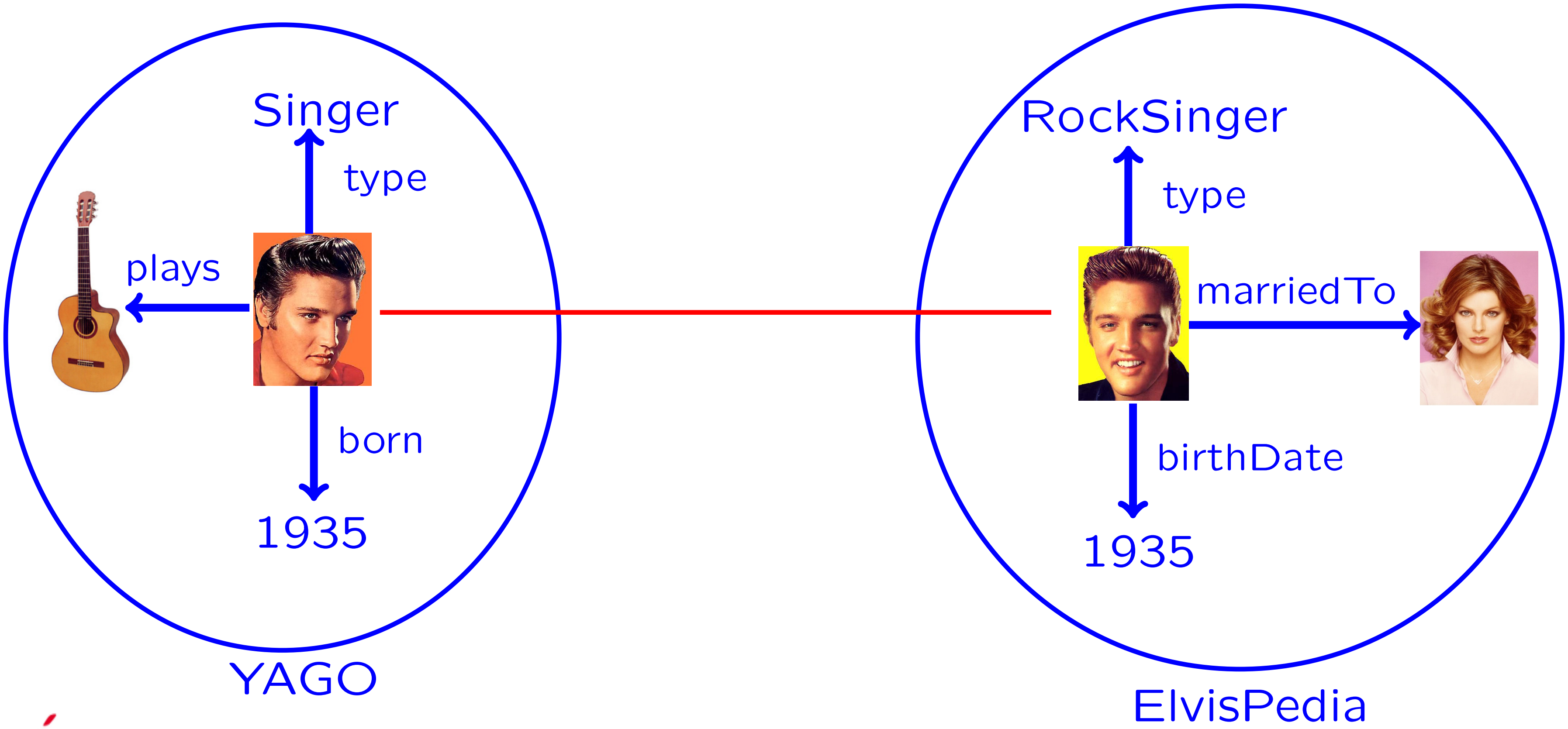


Solution: Unify Entities



Goal: Merging Ontologies

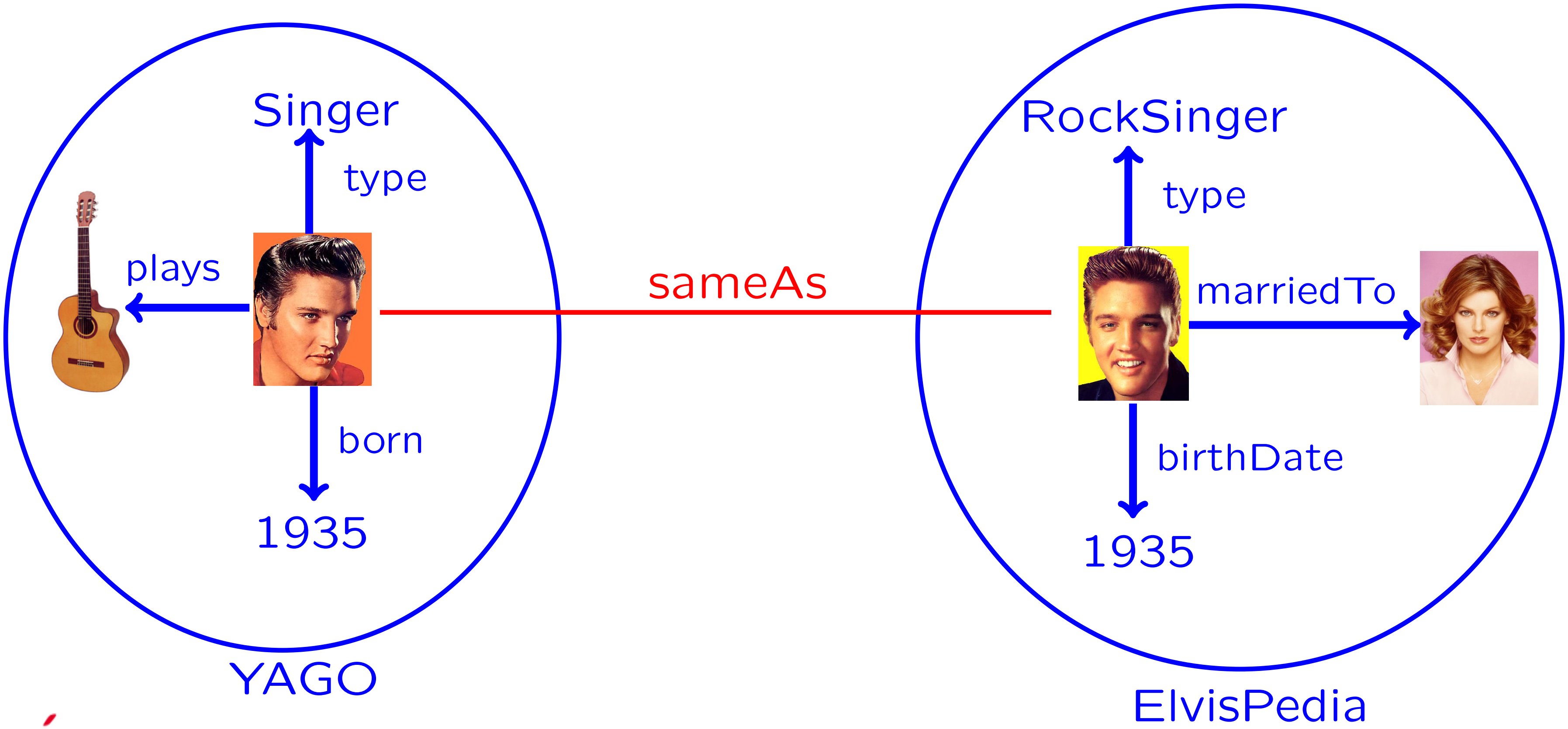
To merge two ontologies, we have to identify



Goal: Merging Ontologies

To merge two ontologies, we have to identify

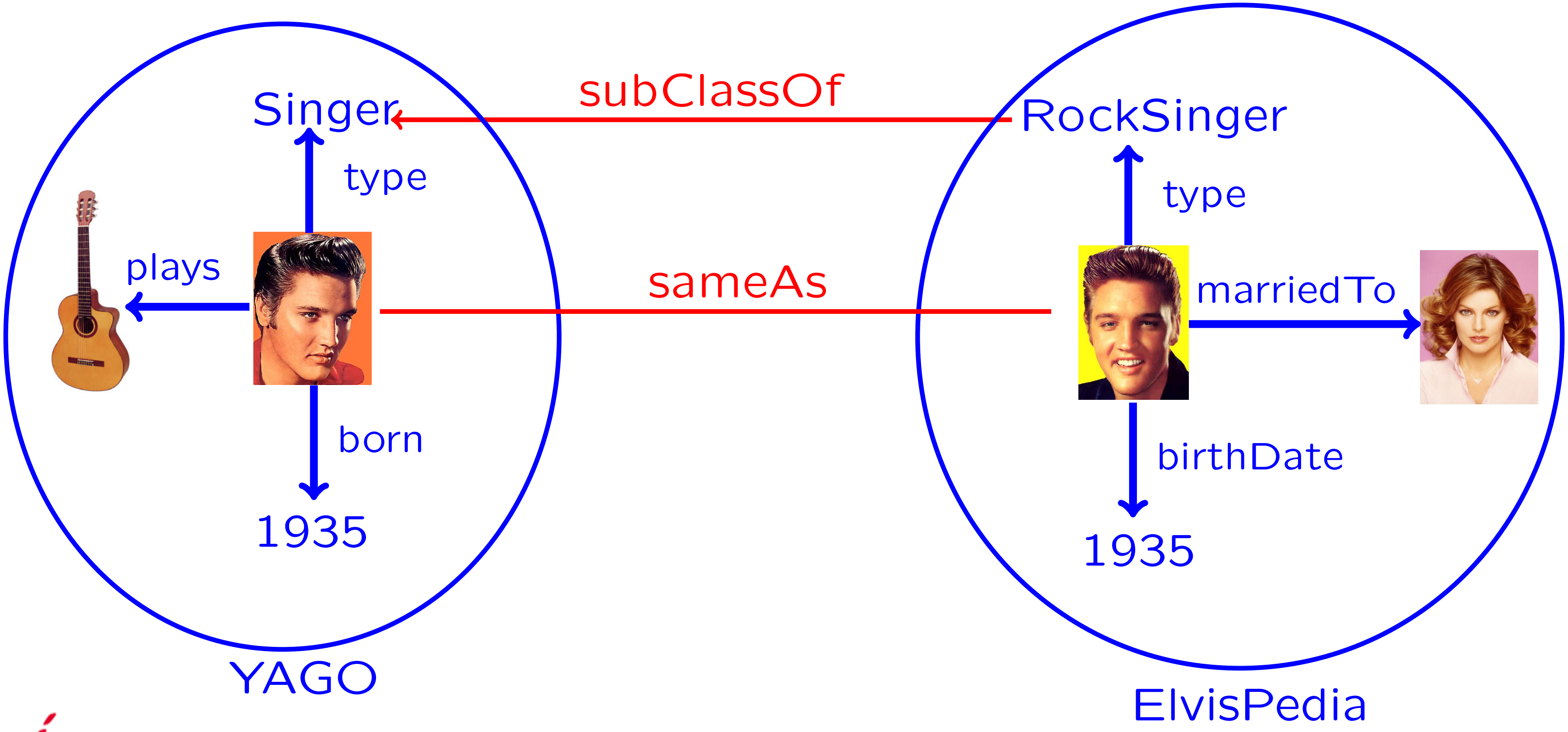
- equivalent instances



Goal: Merging Ontologies

To merge two ontologies, we have to identify

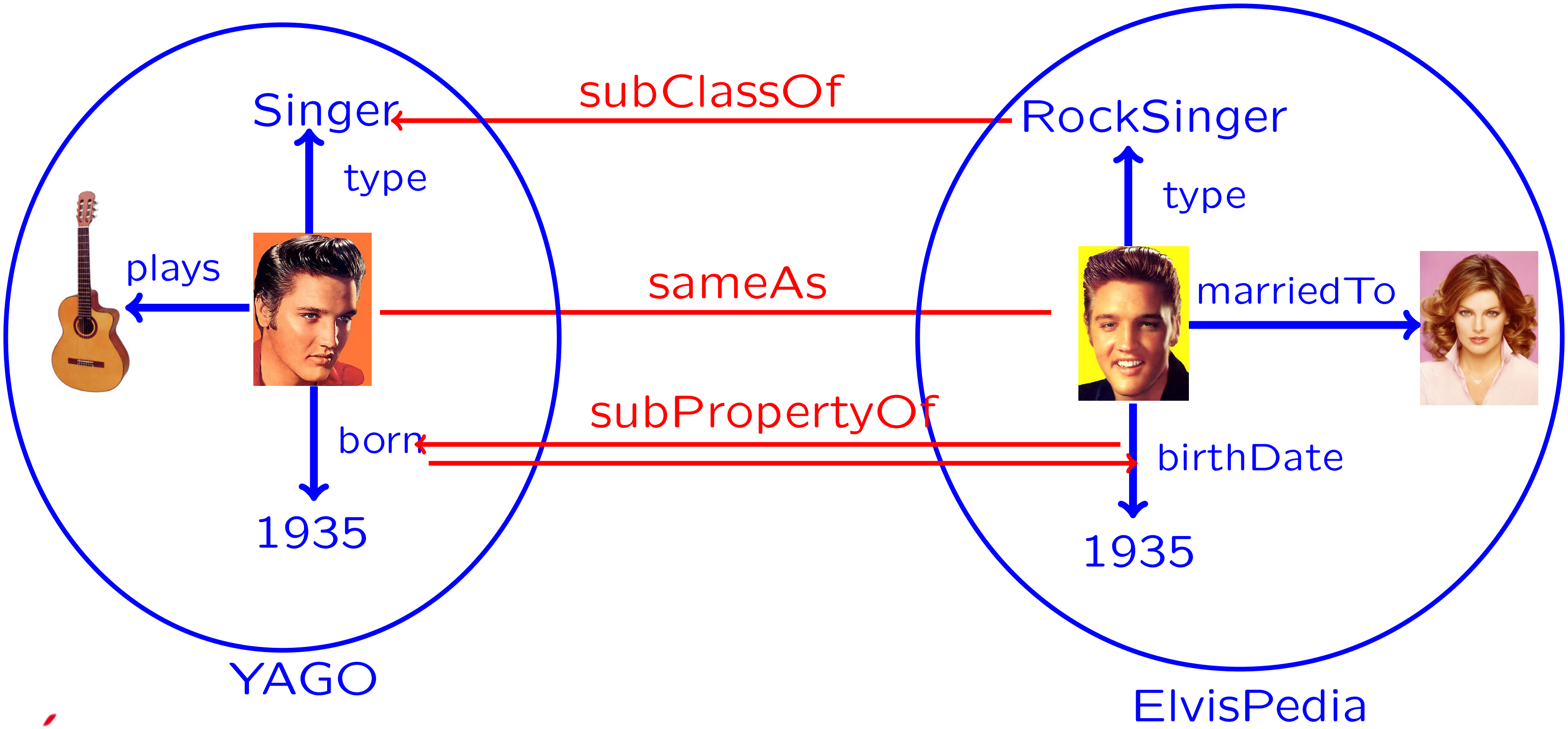
- equivalent instances
- equivalent or subsuming classes



Goal: Merging Ontologies

To merge two ontologies, we have to identify

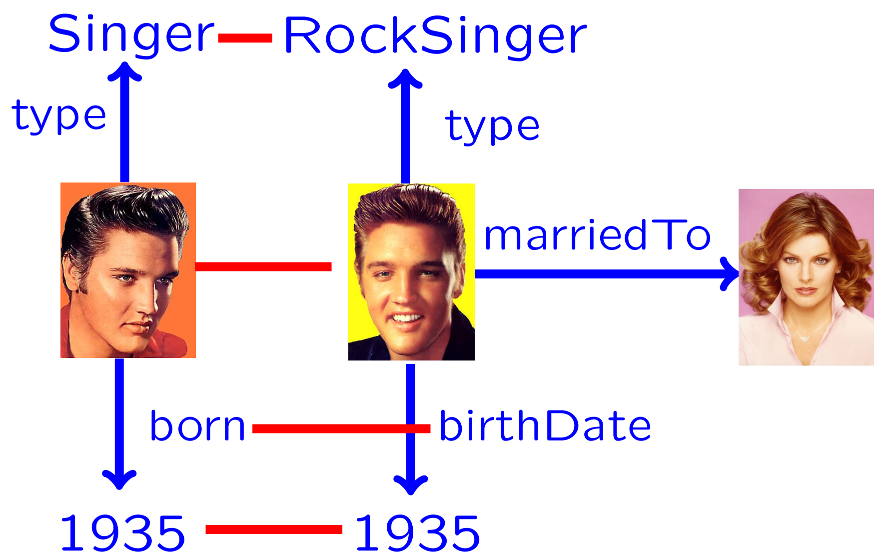
- equivalent instances
- equivalent or subsuming classes
- equivalent or subsuming relations



Previous work

Previous work

- uses hard logical constraints, which may be inadequate
- requires parameter tuning
- has not been tried on large ontologies

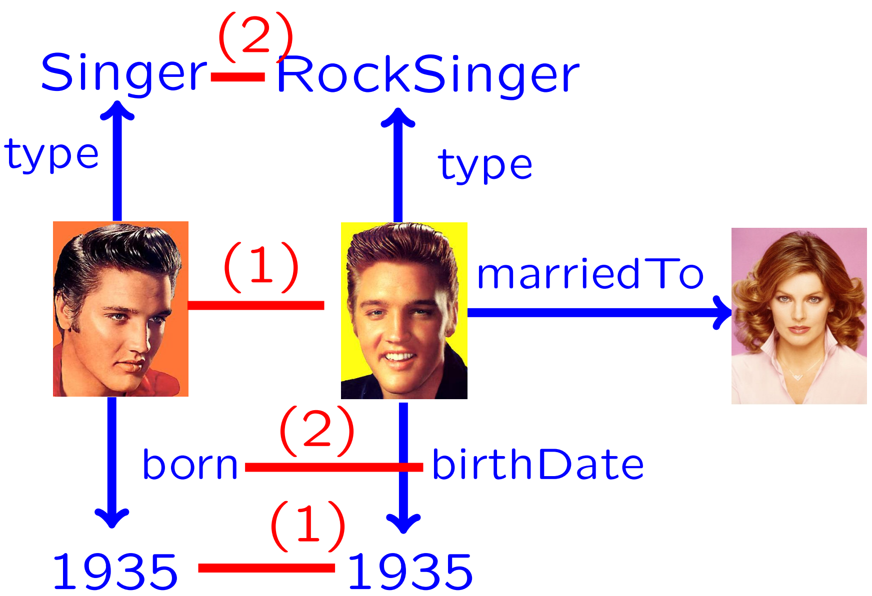


[Gracia 2009, Jean-Mary 2009, Isaac 2007, Aumueller 2005, Wang 2008, Noessner 2010, Sais 2007/2009, Arasu 2009, Volz 2009, Bhattacharya 2007, Hogan 2007/2010, Hu 2011, Li 2009, Udrea 2007, and more]

Previous work

Previous work
- uses hard logical constraints, which may be inadequate
- requires parameter tuning
- has not been tried on large ontologies

- has mostly focused on
  (1) instance matching or
  (2) schema alignment,
  ... but not both

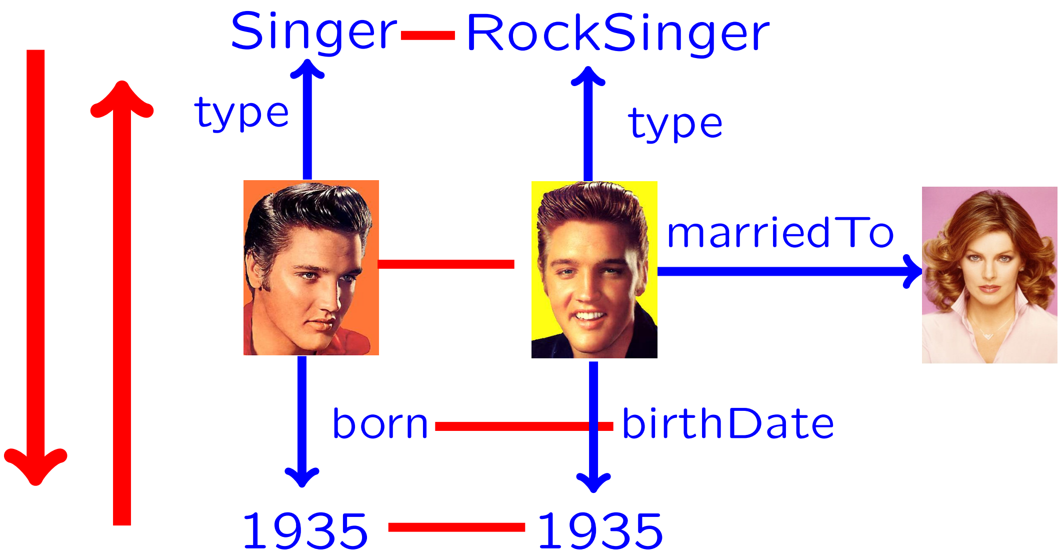


[Gracia 2009, Jean-Mary 2009, Isaac 2007, Aumueller 2005, Wang 2008, Noessner 2010, Sais 2007/2009, Arasu 2009, Volz 2009, Bhattacharya 2007, Hogan 2007/2010, Hu 2011, Li 2009, Udrea 2007, and more]

PARIS: Aligning Everything At Once

There is a synergy between equality of instances, properties and classes!

=> Compute all together!
PARIS



Probabilistic Model

We chose a probabilistic model

$Pr(c_1 \subseteq c_2) =$

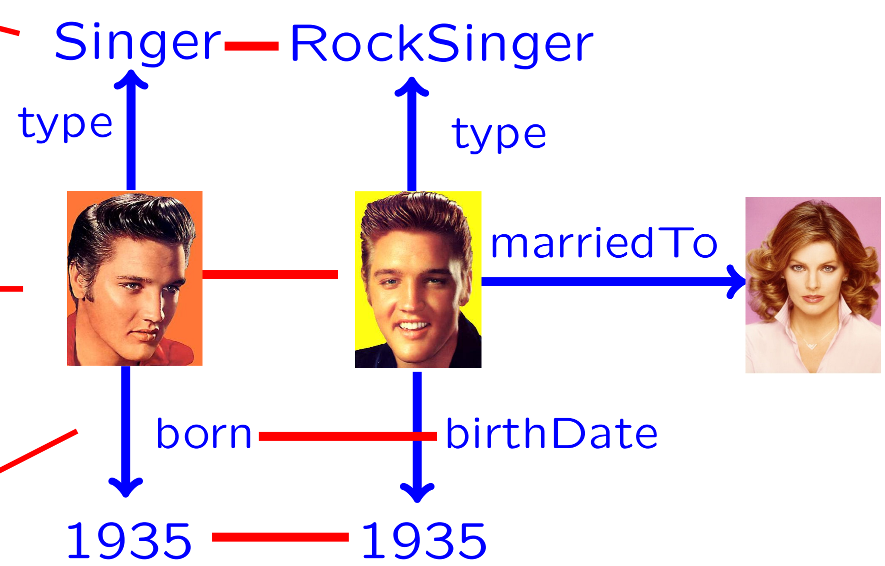
the probability that c_1
is a sub-class of c_2

$Pr(x \equiv y) =$

the probability that x=y

$Pr(p_1 \subseteq p_2) =$

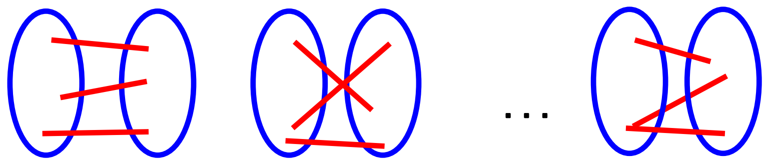
the probability that p_1
is a sub-property of p_2



Probabilistic Model

Worlds:

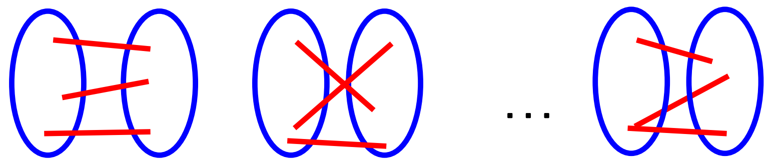
w_1 w_2 ... w_n



All possible alignments between the ontologies

Probabilistic Model

Worlds: w_1 w_2 ... w_n



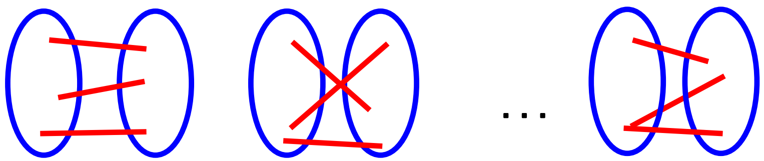
All possible alignments between the ontologies

Probabilities: $Pr(w_1)$ $Pr(w_2)$ ... $Pr(w_n)$

$\sum_i Pr(w_i) = 1$

Probabilistic Model

Worlds: w_1 w_2 $\ldots$ w_n



All possible alignments between the ontologies

Probabilities: $Pr(w_1)$ $Pr(w_2)$ $\ldots$ $Pr(w_n)$

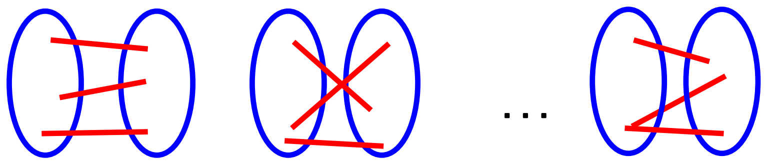
$\sum_i Pr(w_i) = 1$

Events: $e^1_{42} \equiv e^2_{42}$

We care here mainly about equality and subsumption events.
Each event can be true or false in a particular world.

Probabilistic Model

Worlds: w_1 w_2 $\ldots$ w_n



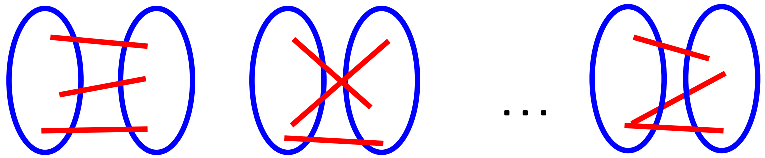
Probabilities: $Pr(w_1)$ $Pr(w_2)$ $\ldots$ $Pr(w_n)$

Marginals: $Pr(e^1_{42} \equiv e^2_{42}\ ) = Pr(w_1) + Pr(w_9)$

The marginal probability of an event
is given by the sum of the probabilities
of the worlds where the event holds.

Probabilistic Model

Worlds: w_1 w_2 $\ldots$ w_n



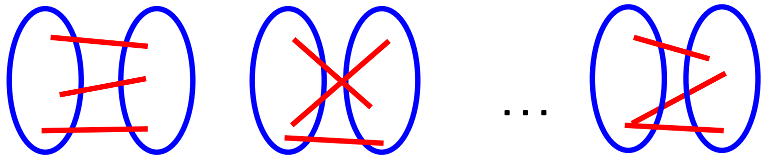
We do not care about these values.

Probabilities: $Pr(w_1)$ $Pr(w_2)$ $\ldots$ $Pr(w_n)$

Marginals: $Pr(e^1_{42} \equiv e^2_{42}) = Pr(w_1) + Pr(w_9)$

Probabilistic Model

Worlds: w_1 w_2 $\ldots$ w_n



Probabilities: $Pr(w_1)$ $Pr(w_2)$ $\ldots$ $Pr(w_n)$

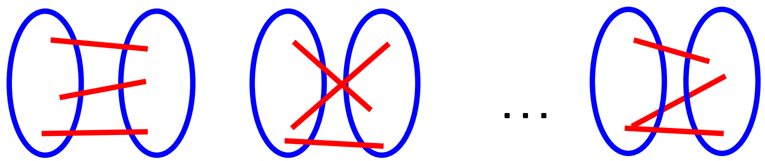
We do not care about these values.

Marginals: $Pr(e^1_{42} \equiv e^2_{42} \;) = Pr(w_1) + Pr(w_9)$

We only care about these marginals!

Probabilistic Model

Worlds: w_1 w_2 $\dots$ w_n



Probabilities: $Pr(w_1)$ $Pr(w_2)$ $\dots$ $Pr(w_n)$

Marginals: $Pr(e^1_{42} \equiv e^2_{42}\ \ ) = Pr(w_1) + Pr(w_9)$

We only care about these marginals!

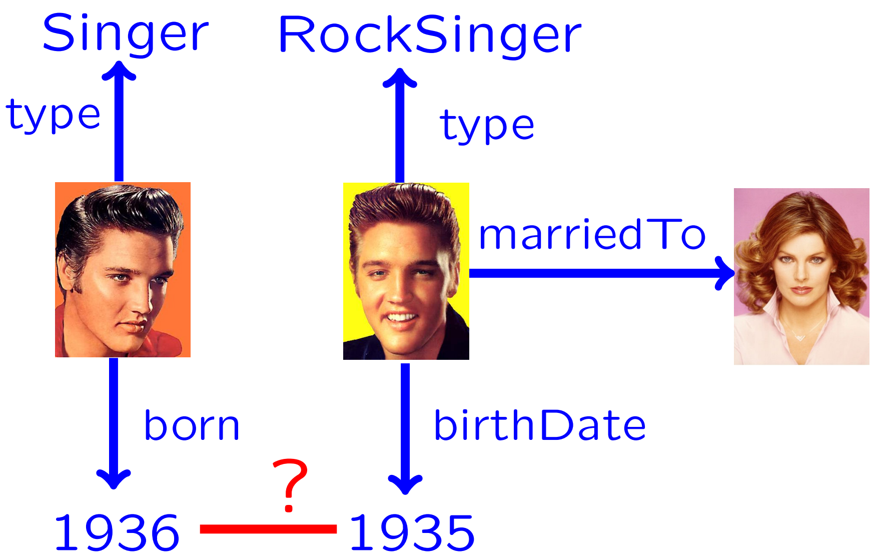
We are interested in marginals that fulfill certain properties.

For any set of marginals, there exists a probability distribution.

(it is the product measure)

Literals

The probability that two literals are equal shall reflect the likelihood that the two literals are intended to refer to the same thing.

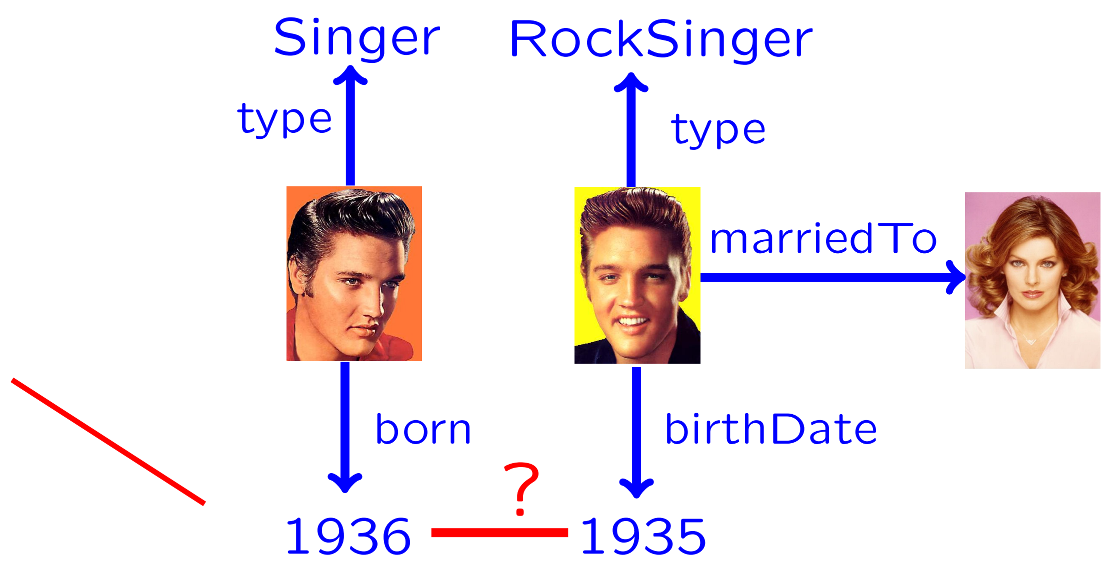


Literals

The probability that two literals are equal shall reflect the likelihood that the two literals are intended to refer to the same thing.

$$Pr(x \equiv y) =$$

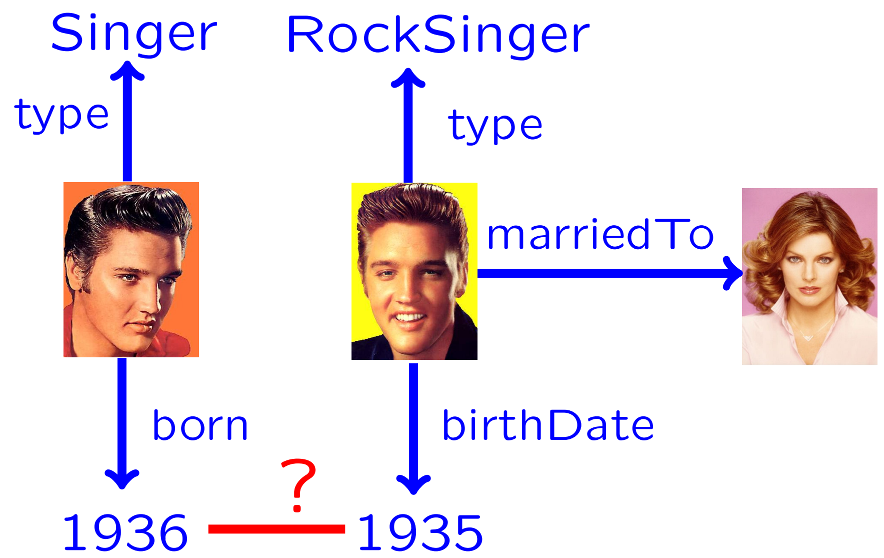
- for strings:
  string distance
- for numbers:
  numeric distance
- for other literals:
  domain-specific



Literals

$$Pr(x \equiv y) =$$

- for strings:
  string distance
- for numbers:
  numeric distance
- for other literals:
  domain-specific

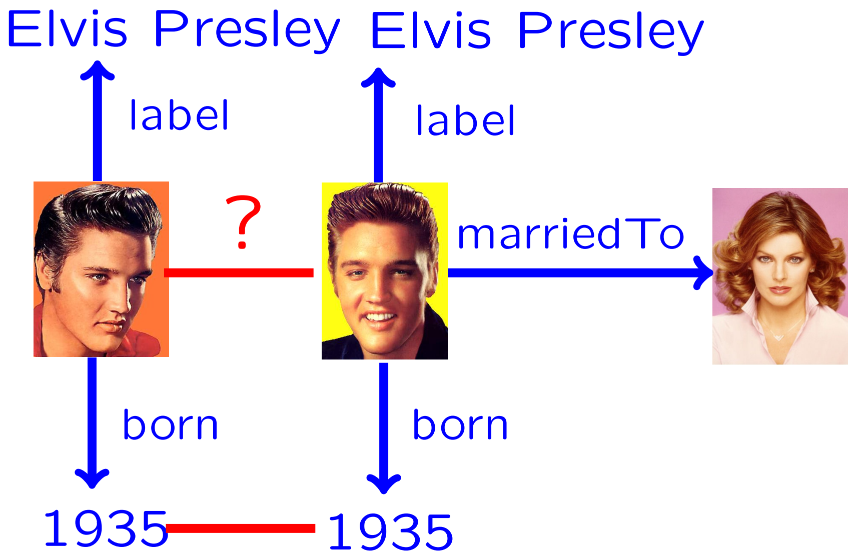


We chose a particularly simple equality:

$$Pr(x \equiv y) := (x = y)?1 : 0$$

Equality of Instances

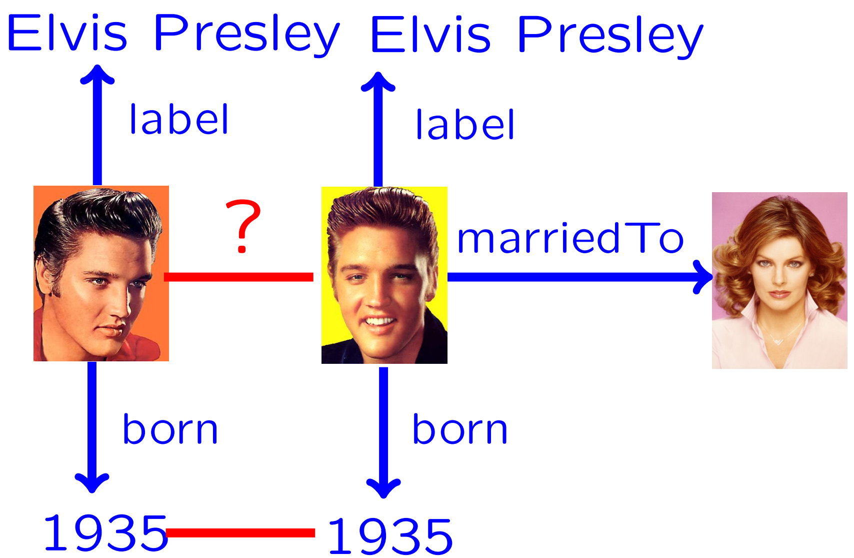
For instances, relations give a hint:



Equality of Instances

For instances, relations give a hint:

Not many people are
called Elvis =>
highly indicative

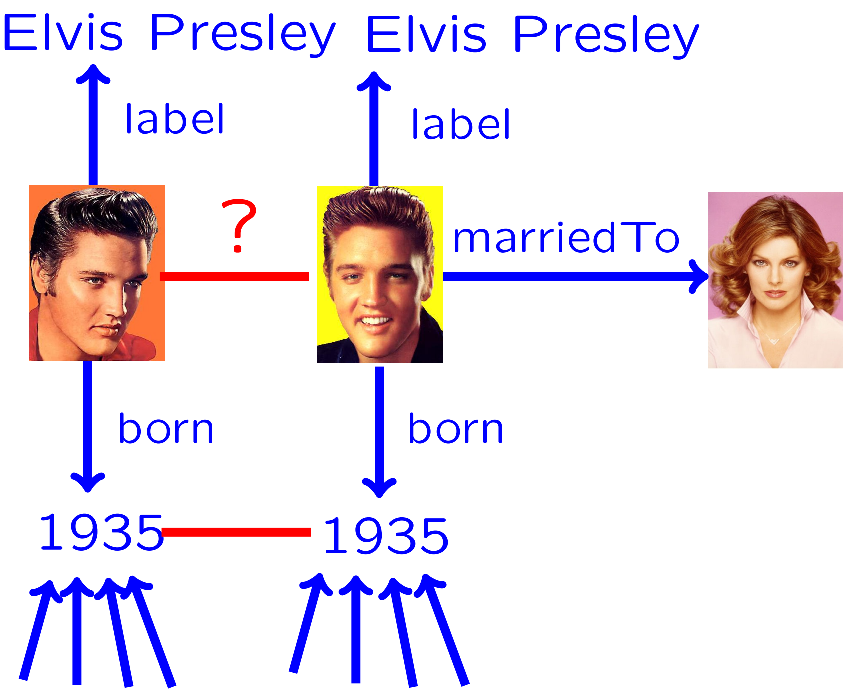


Equality of Instances

For instances, relations give a hint:

Not many people are
called Elvis =>
highly indicative

Many people are born
in 1935 =>
less indicative



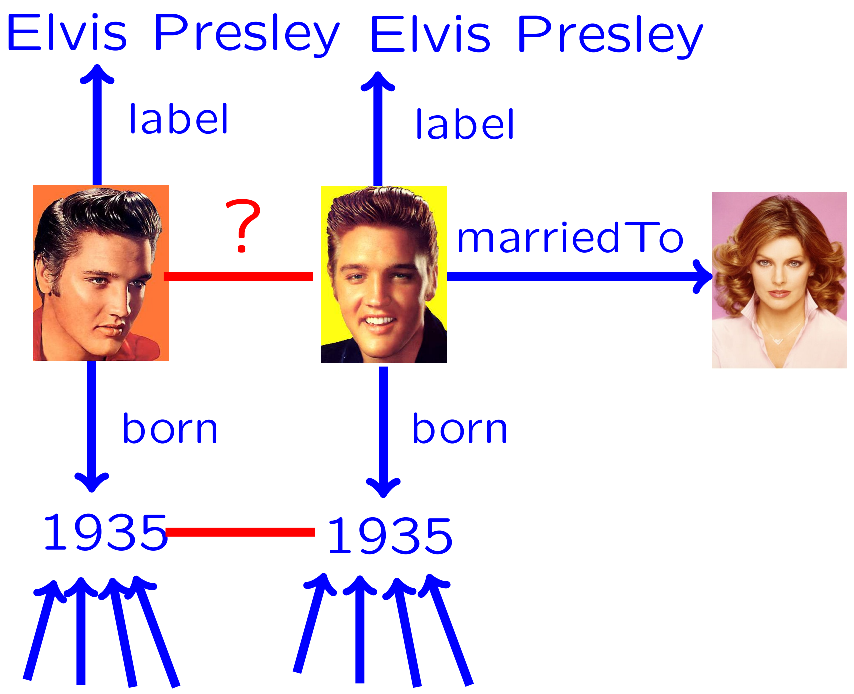
Local Inverse Functionality

The local inverse functionality of a relation with an argument is one over the number of incoming links:

$$ifun(r, y) := \frac{1}{|\{x : r(x,y)\}|}$$

Not many people are called Elvis => highly indicative

Many people are born in 1935 => less indicative



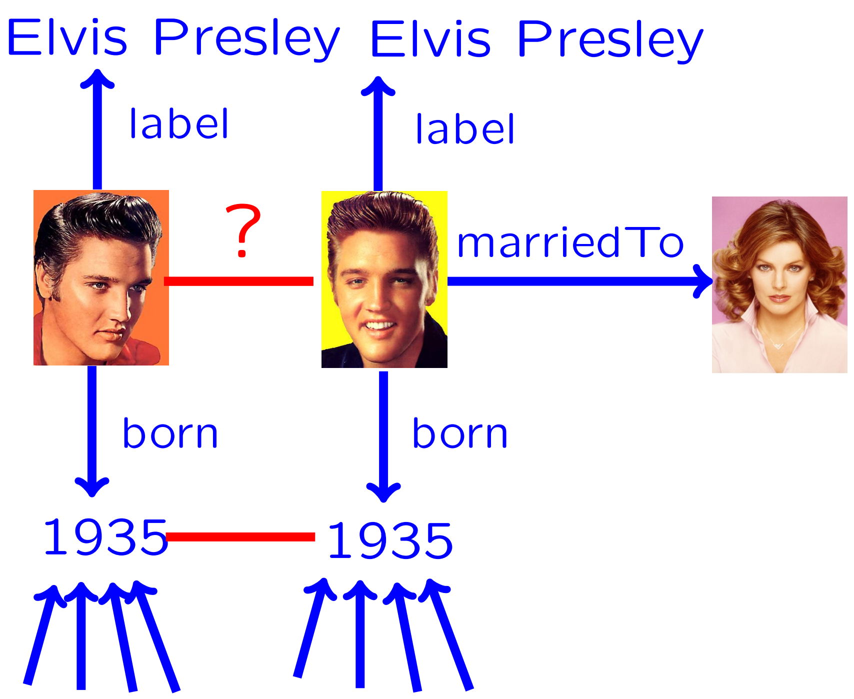
Local Inverse Functionality

The local inverse functionality of a relation with an argument is one over the number of incoming links:

$$ifun(r, y) := \frac{1}{|\{x : r(x,y)\}|}$$

Only one person
is called Elvis

$$ifun(label, Elvis) = 1$$



Local Inverse Functionality

The local inverse functionality of a relation with an argument is one over the number of incoming links:

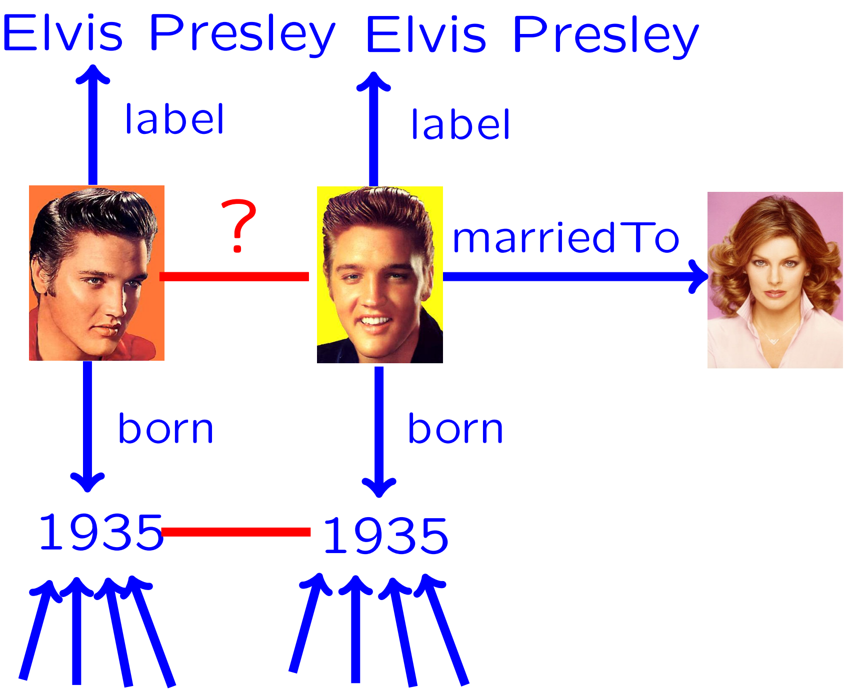
$$ifun(r,y) := \frac{1}{|\{x : r(x,y)\}|}$$

Only one person
is called Elvis

$$ifun(label, Elvis) = 1$$

10 people are born
in 1935

$$ifun(born, 1935) = 0.1$$



Probability of Inverse Functionalit

We define the probability of a relation being inverse functional as the harmonic mean of the local inverse functionalities:

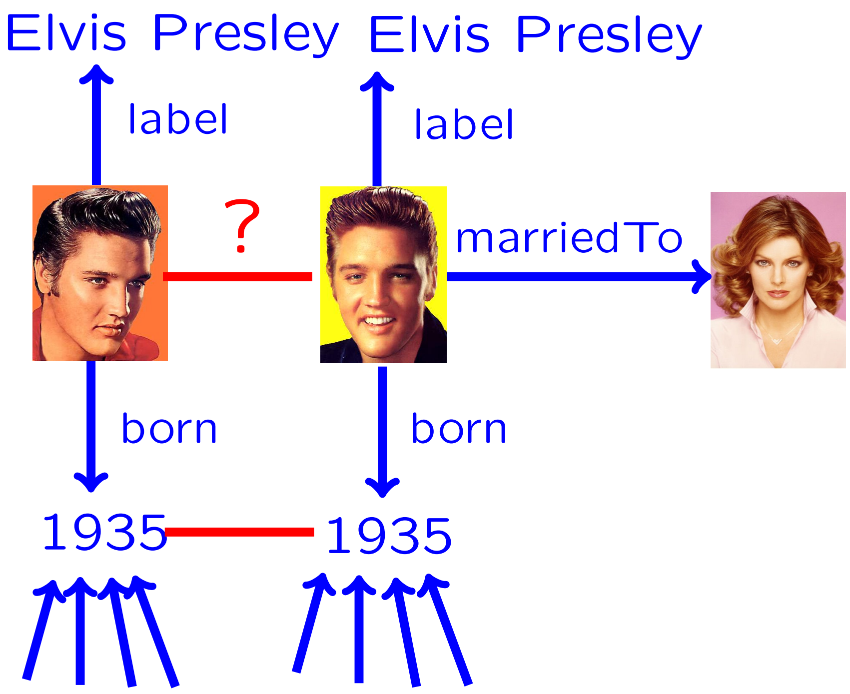
$$Pr(ifun(r)) := HM_y ifun(r, y)$$

Few people share the same name

$$Pr(ifun(label)) = 0.8$$

Many people share their year of birth

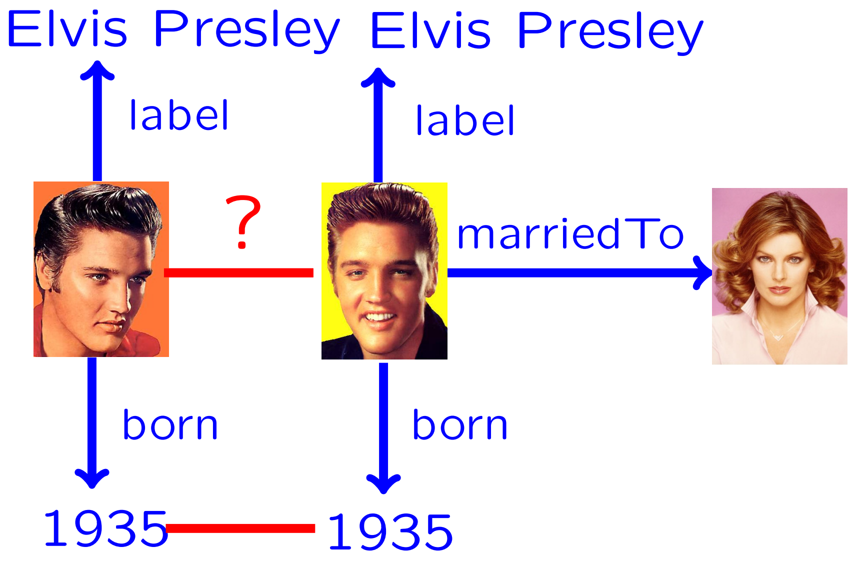
$$Pr(ifun(born)) = 0.01$$



Equality of Instances

x and x' shall be matched if

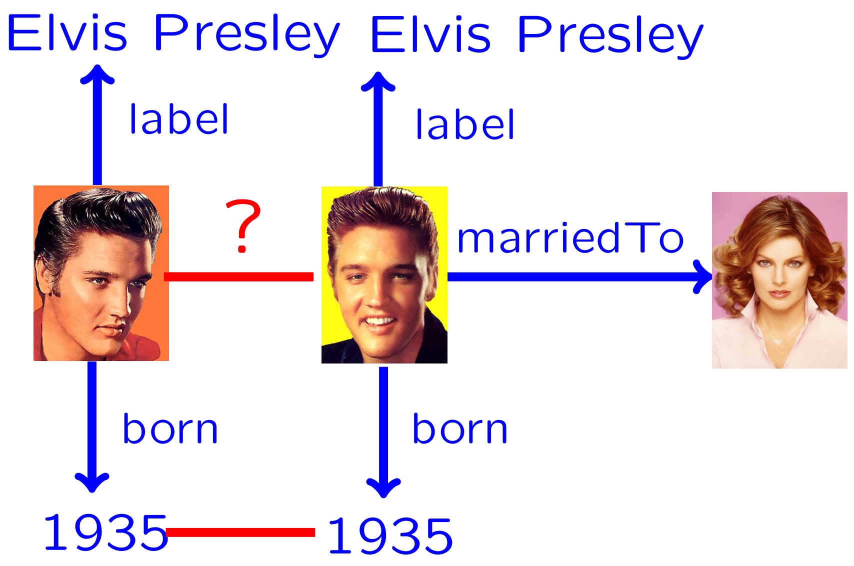
$$\exists r, y, y' with r(x,y), r(x',y') : y \equiv y' \wedge ifun(r)$$



Equality of Instances

$Pr(x \equiv x')$

$$= Pr(\ \exists r, y, y' with r(x, y), r(x', y') : y \equiv y' \wedge ifun(r) \qquad\qquad)$$

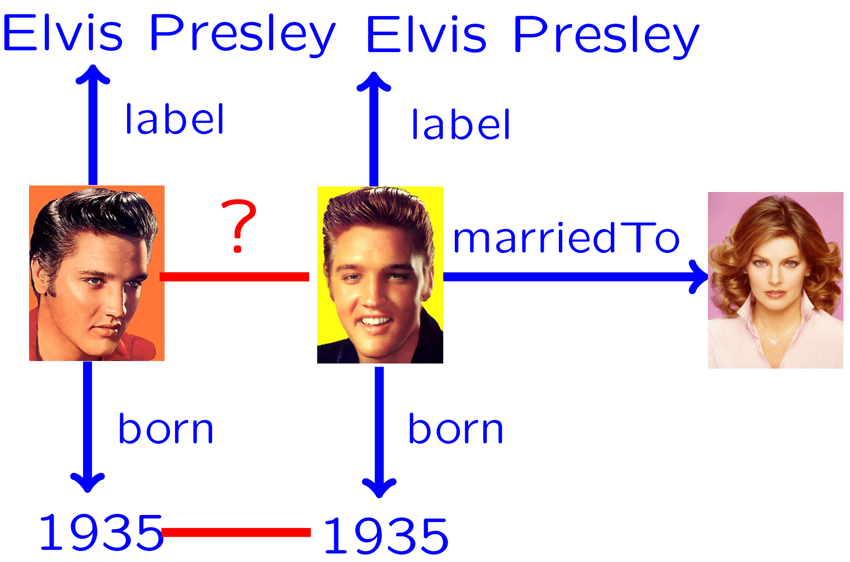


Equality of Instances

$$Pr(x \equiv x')$$

$$= Pr(\ \exists r, y, y' with r(x,y), r(x',y') : y \equiv y' \wedge ifun(r) \qquad\qquad)$$

$$= 1 - \prod_{r(x,y), r(x',y')} (1 - Pr(y \equiv y') Pr(ifun(r)))$$



Equality of Instances

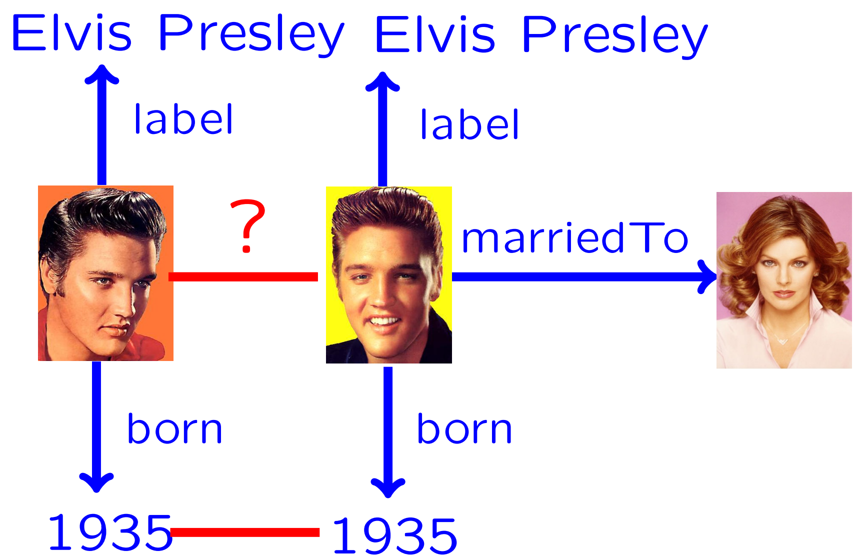
$$Pr(x \equiv x')$$

$$= Pr(\ \exists r, y, y' with r(x,y), r(x',y') : y \equiv y' \wedge ifun(r) \qquad\qquad)$$

$$= 1 - \prod_{r(x,y), r(x',y')} (1 - Pr(y \equiv y') Pr(ifun(r)))$$

This evaluates to 1 iff

- There is at least one highly inverse functional r
- There is one shared argument y=y'



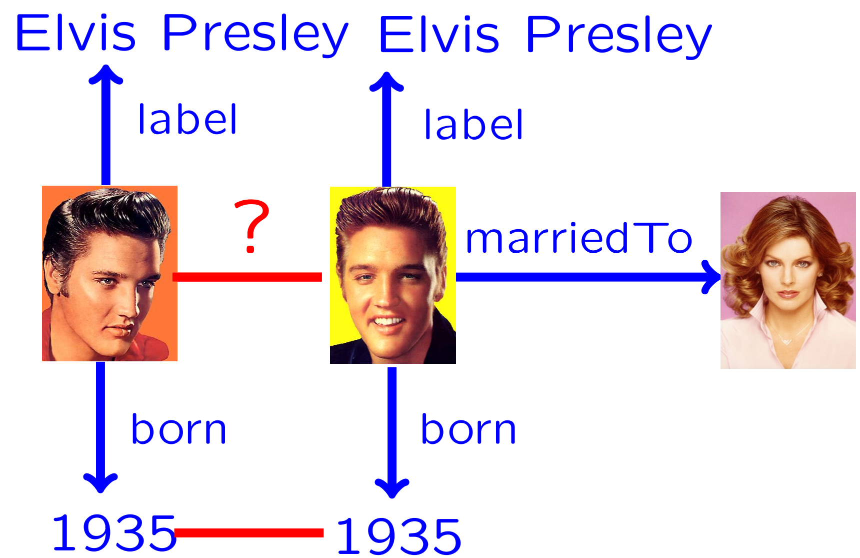
Equality of Instances

$Pr(x \equiv x')$

$= Pr(\ \exists r, y, y' with r(x,y), r(x',y') : y \equiv y' \wedge ifun(r) \qquad\qquad)$

$= 1 - \prod_{r(x,y),r(x',y')}(1 - \underline{Pr(y \equiv y')}Pr(ifun(r)))$

- Literals: precomputed
- Others: recursive

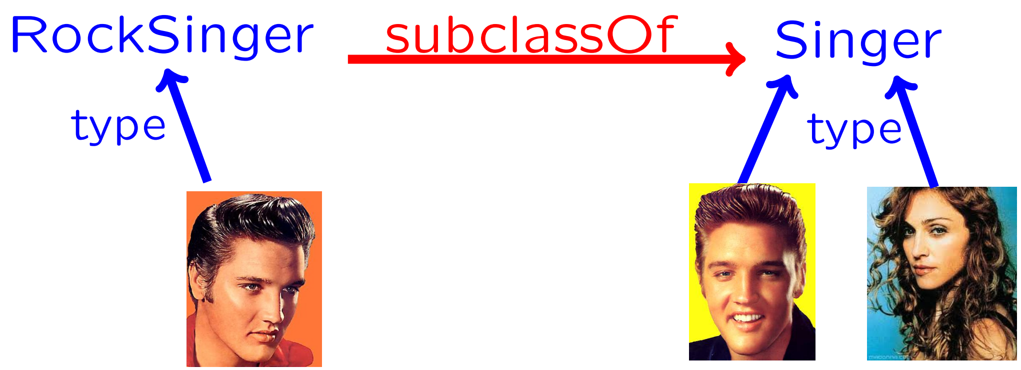


Equality of Classes

If all instances of one class are instances of the other then the former subsumes the latter

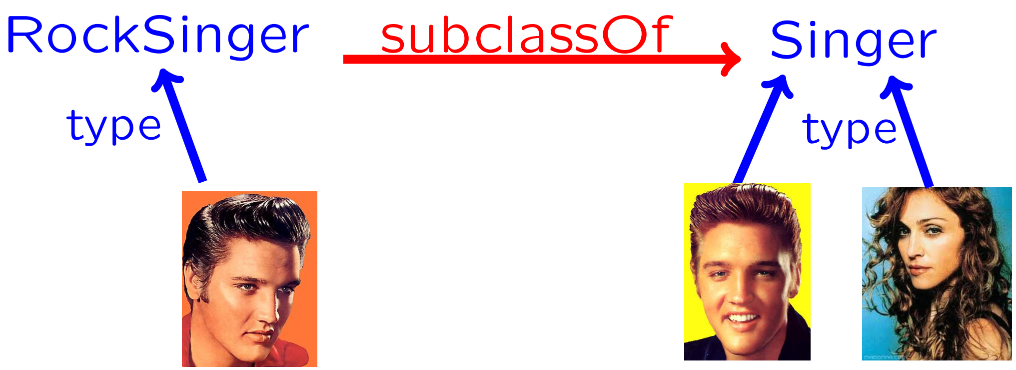
Equality of Classes

If all instances of one class are instances of the other then the former subsumes the latter



Equality of Classes

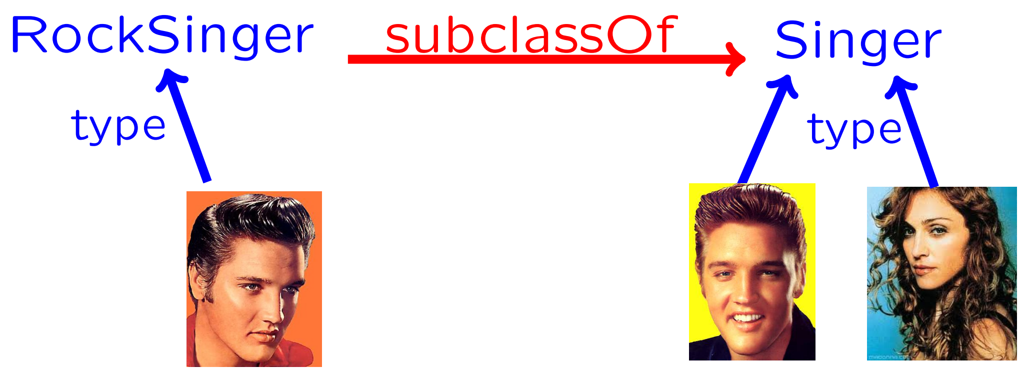
If all instances of one class are instances of the other then the former subsumes the latter



$$Pr(C \subseteq D) = \frac{|C \cap D|}{|C|}$$

Equality of Classes

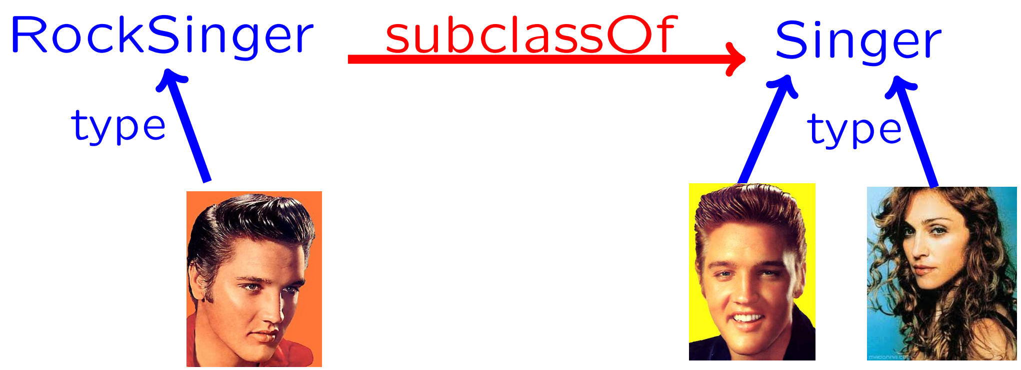
If all instances of one class are instances of the other then the former subsumes the latter



$$Pr(C \subseteq D) = \frac{|C \cap D|}{|C|} = \frac{\sum_{x \in C} Pr(\exists y \in D : x \equiv y)}{|C|}$$

Equality of Classes

If all instances of one class are instances of the other then the former subsumes the latter

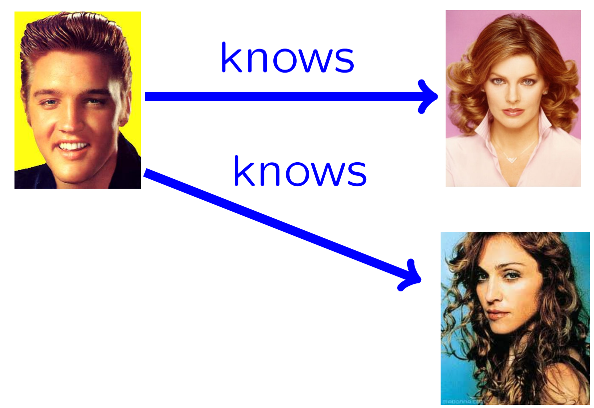


$$Pr(C \subseteq D) = \frac{|C \cap D|}{|C|} \qquad = \frac{\sum_{x \in C} Pr(\exists y \in D : x \equiv y)}{|C|}$$

$$Pr(C \subseteq D) = \frac{\sum_{x \in C}(1 - \prod_{y \in D}(1 - Pr(x \equiv y)))}{|C|}$$

Equality of Relations

If every pair of one relation is a pair of another relation, then the first is a sub-property of the second.



Equality of Relations

If every pair of one relation is a pair of another relation, then the first is a sub-property of the second.

Equality of Relations

If every pair of one relation is a pair of another relation, then the first is a sub-property of the second.

Can be solved analogously to the subsumption of classes.

Algorithm

1. Fix equalities for literals

Literals: $Pr(x \equiv y) = fixedforliterals$

Algorithm

1. Fix equalities for literals
2. Set equalities for relations to a small initial value

Relations: $Pr(p_1 \subseteq p_2) =$ 0.1

Literals: $Pr(x \equiv y) = fixedforliterals$

Algorithm

1. Fix equalities for literals
2. Set equalities for relations to a small initial value
3. Iterate the estimations for relations and instances to convergence (*)

Iterate

Relations: $Pr(p_1 \subseteq p_2) = \ 13\phi ..$

Instances: $Pr(x \equiv y) = \prod_{42}^{1} \alpha^{\beta} ...$

Literals: $Pr(x \equiv y) = fixedforliterals$

(*) We have no proof for convergence, but it seems to happen

Algorithm

1. Fix equalities for literals
2. Set equalities for relations to a small initial value
3. Iterate the estimations for relations and instances to convergence (*)
4. Compute the estimations for classes

Iterate

Relations: $Pr(p_1 \subseteq p_2) = \; 13\phi\cdot..$

Instances: $Pr(x \equiv y) = \prod_{42}^{1} \alpha^{\beta}...$

Literals: $Pr(x \equiv y) = fixedforliterals$

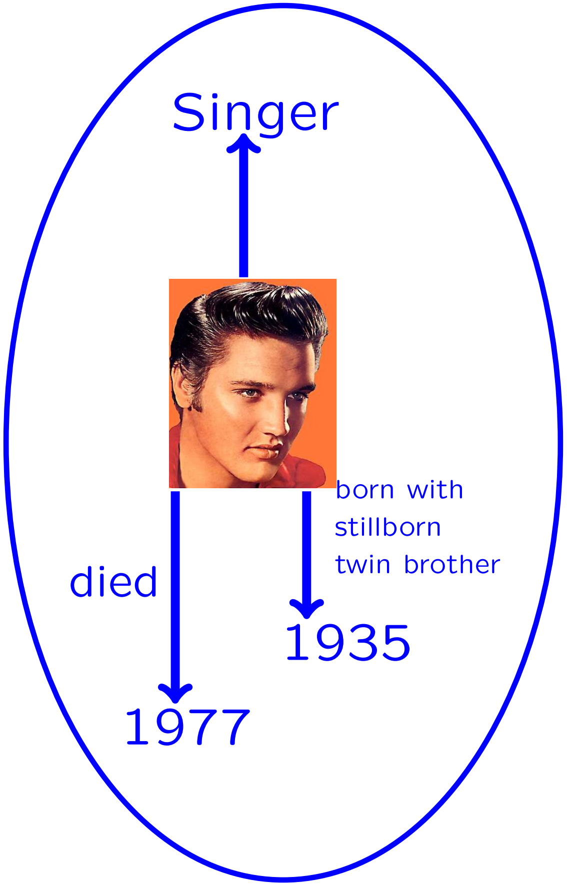
final step

Classes: $Pr(c_1 \subseteq c_2) = \pi m c^2...$

(*) We have no proof for convergence, but it seems to happen

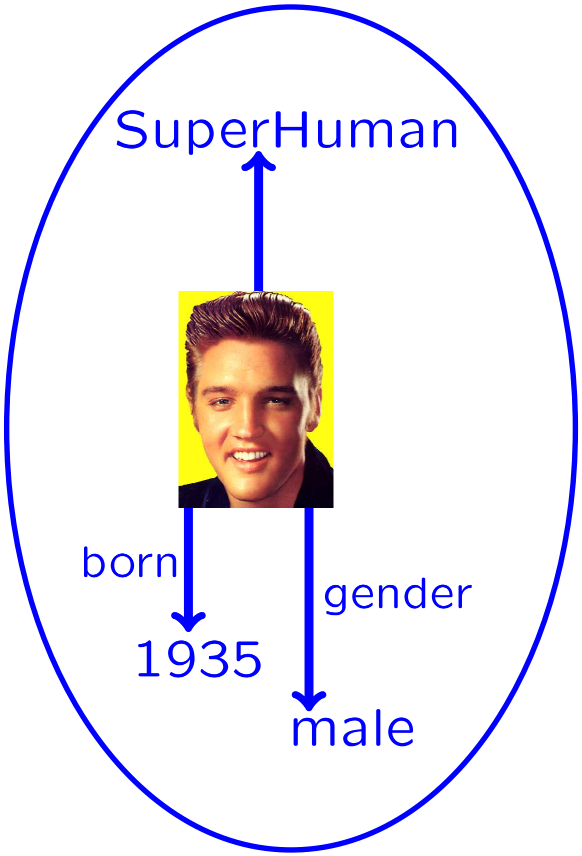
Experiment: YAGO and DBpedia

DBpedia



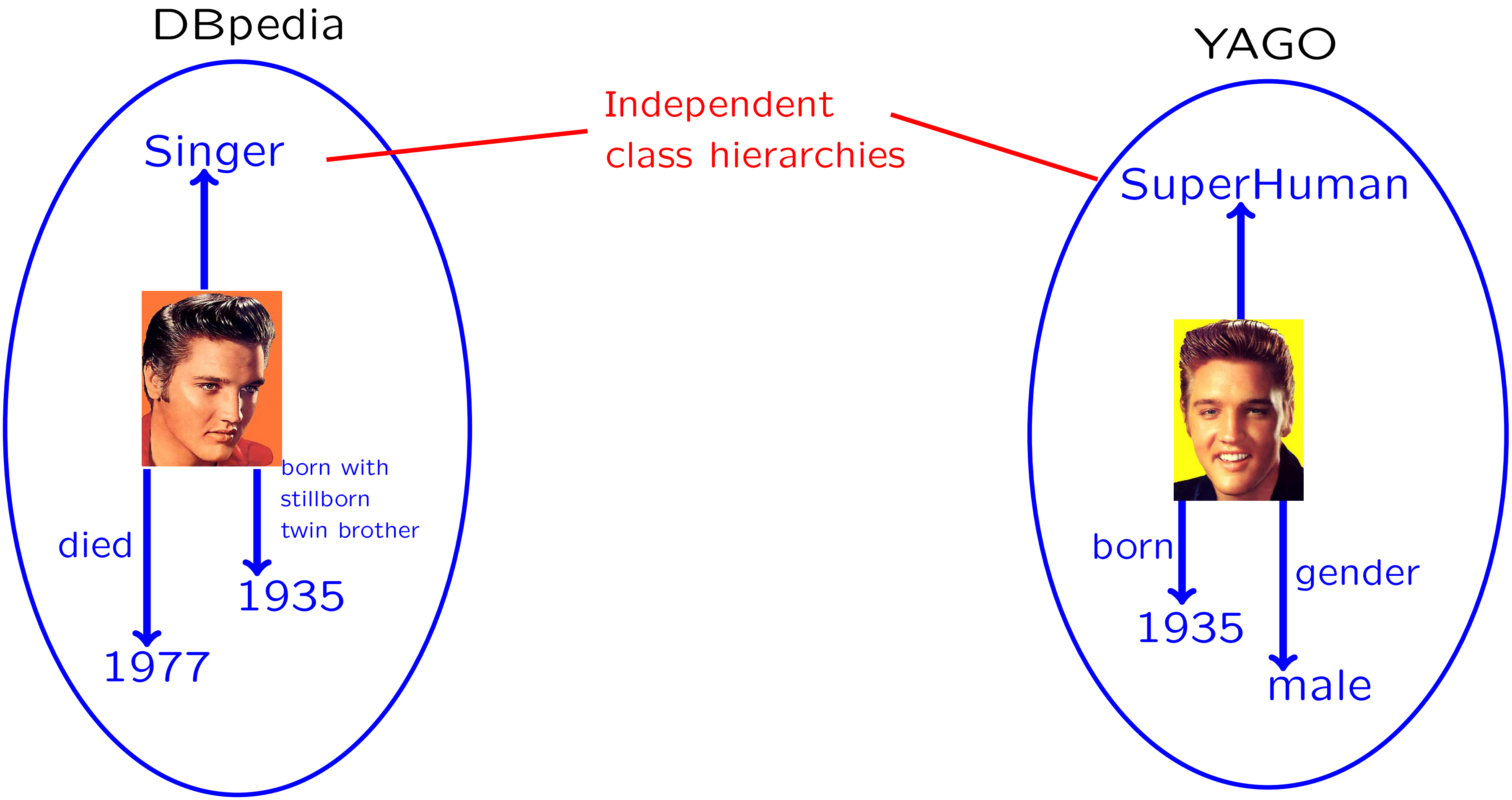
1.7 million instances
19 million facts

YAGO



2.6 million instances
18 million facts

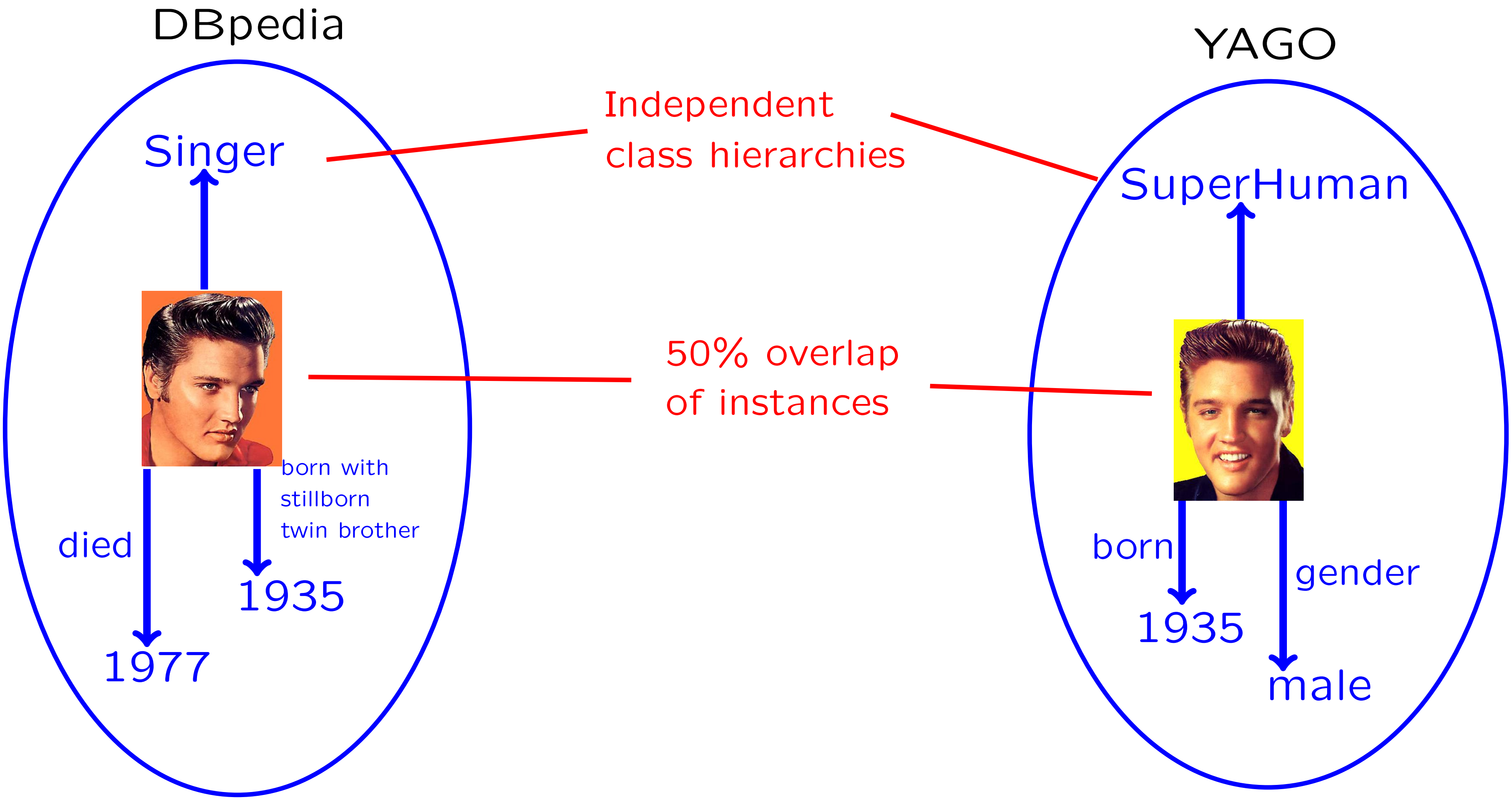
Experiment: YAGO and DBpedia



1.7 million instances
19 million facts

2.6 million instances
18 million facts

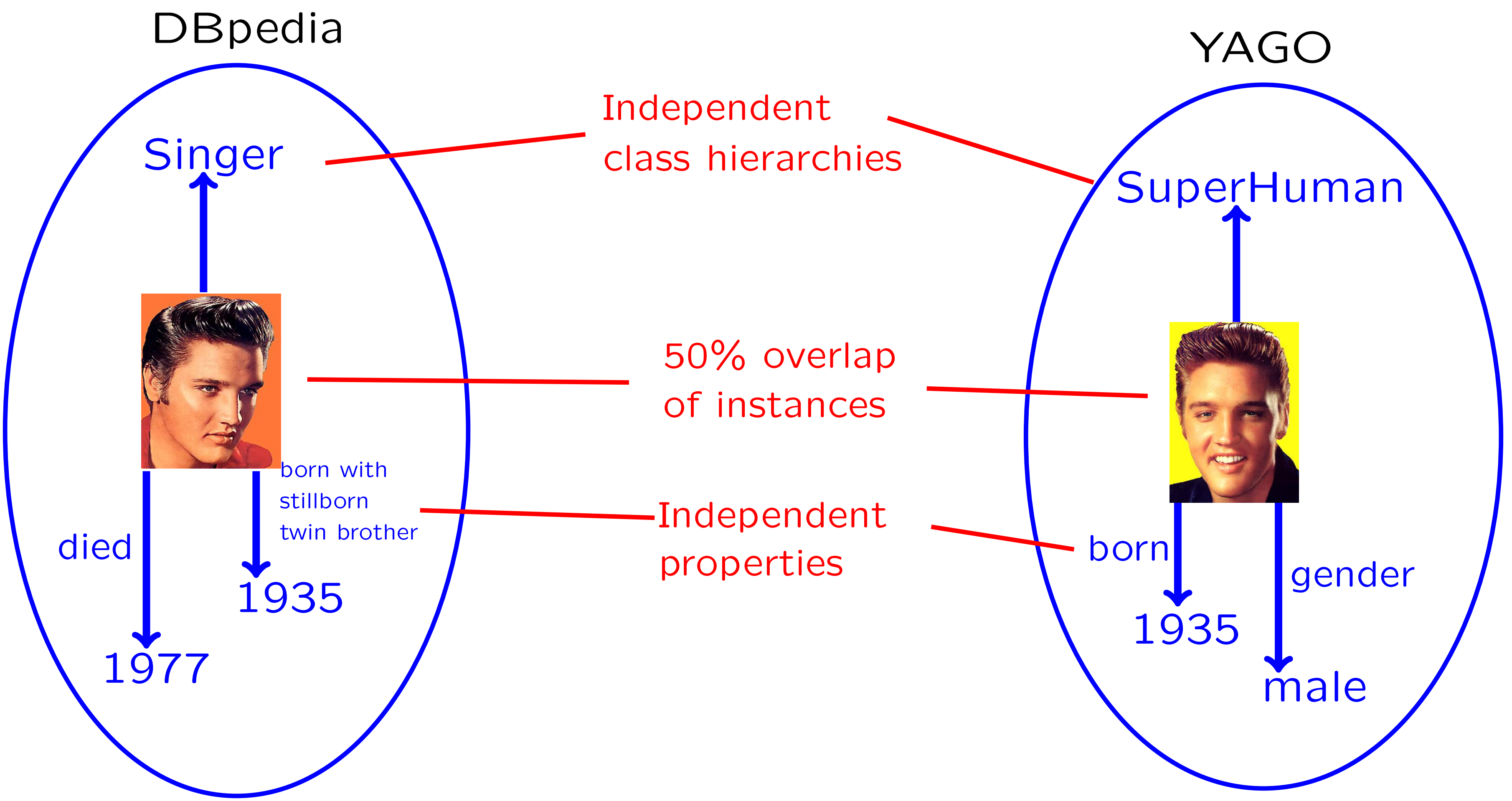
Experiment: YAGO and DBpedia



1.7 million instances
19 million facts

2.6 million instances
18 million facts

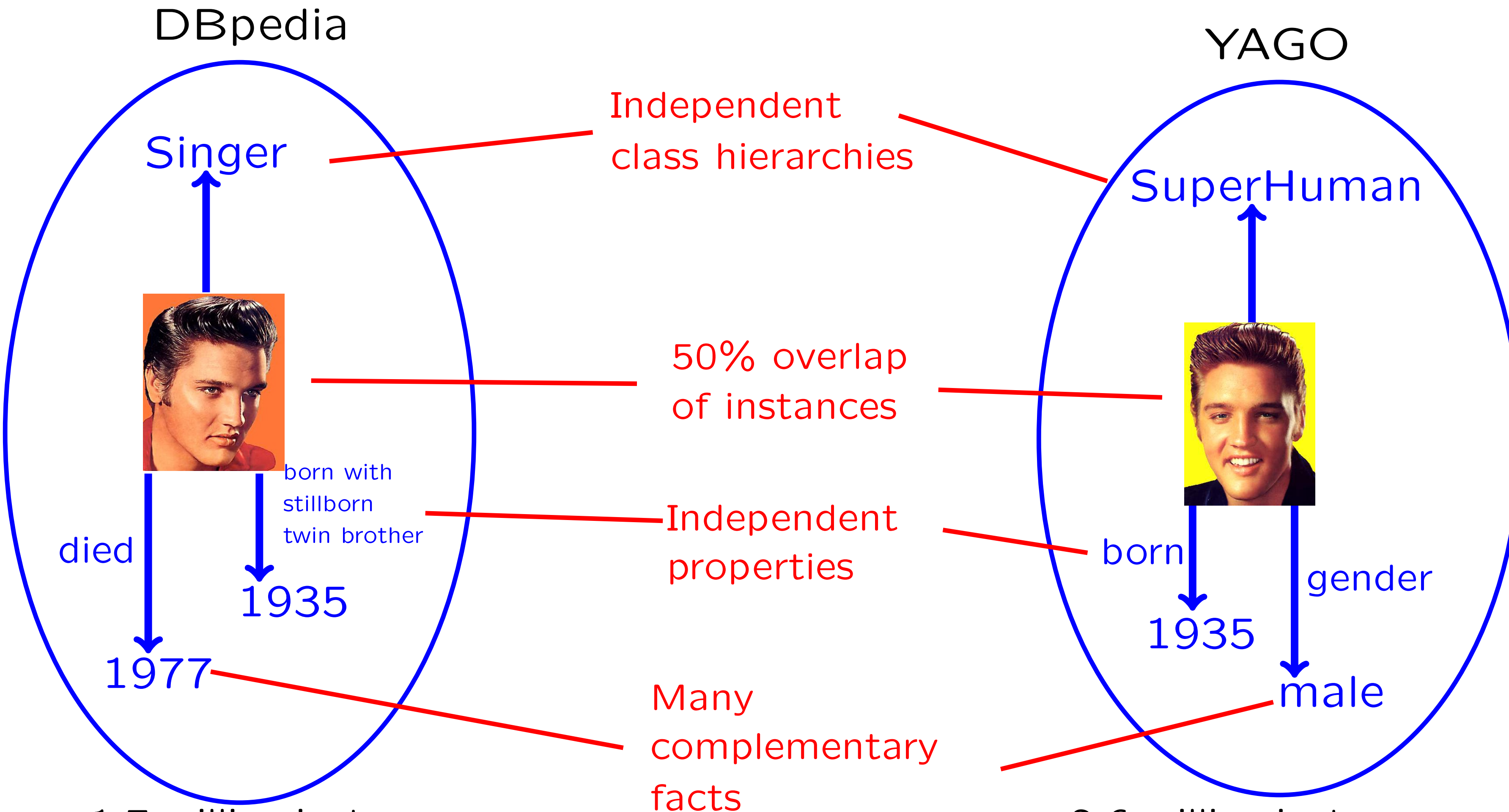
Experiment: YAGO and DBpedia



1.7 million instances
19 million facts

2.6 million instances
18 million facts

Experiment: YAGO and DBpedia



1.7 million instances
19 million facts

2.6 million instances
18 million facts

Experiment: YAGO and DBpedia

Matching the instances:

Iteration	Change	Time	Precision	Recall
1	–	4:04h	86%	69%
2	12%	5:06h	89%	73%
3	1%	5:00h	90%	73%
4	0.3%	5:30h	90%	73%

F-Measure increases

Experiment: YAGO and DBpedia

Matching the instances:

Iteration	Change	Time	Precision	Recall
1	–	4:04h	86%	69%
2	12%	5:06h	89%	73%
3	1%	5:00h	90%	73%
4	0.3%	5:30h	90%	73%

Matching the classes:

Precision: 74%
Aligns roughly half of DBpedia's classes

Experiment: YAGO and DBpedia

Matching the relations:

yago:actedIn = dbpedia:starring-inv
yago:hasChild = dbpedia:child
yago:hasChild = dbpedia:parent-inv
yago:created > dbpedia:writer
yago:diedIn > dbpedia:placeOfBurial
... and many more. Precision: 96%

Experiment: YAGO and DBpedia

Matching the relations:

yago:actedIn = dbpedia:starring-inv
yago:hasChild = dbpedia:child
yago:hasChild = dbpedia:parent-inv
yago:created > dbpedia:writer
yago:diedIn > dbpedia:placeOfBurial
... and many more. Precision: 96%

+ experiments on IMDB
+ experiments on OAEI standard datasets

Experiment: YAGO and DBpedia

Matching the relations:

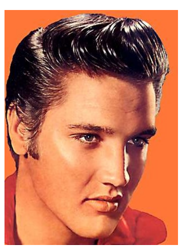
yago:actedIn = dbpedia:starring-inv
yago:hasChild = dbpedia:child
yago:hasChild = dbpedia:parent-inv
yago:created > dbpedia:writer
yago:diedIn > dbpedia:placeOfBurial
... and many more. Precision: 96%

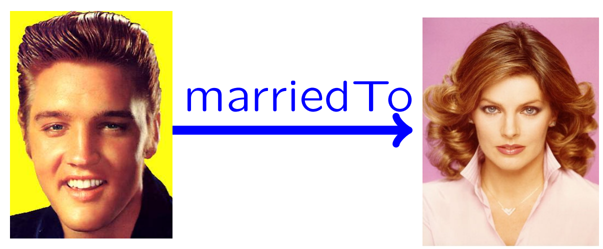
+ experiments on IMDB
+ experiments on OAEI standard datasets

All experiments run with the same settings.
No parameter tuning.

Conclusion

- PARIS matches relations, instances and schema holistically
- PARIS has no parameters to tune
- PARIS shows high precision and recall

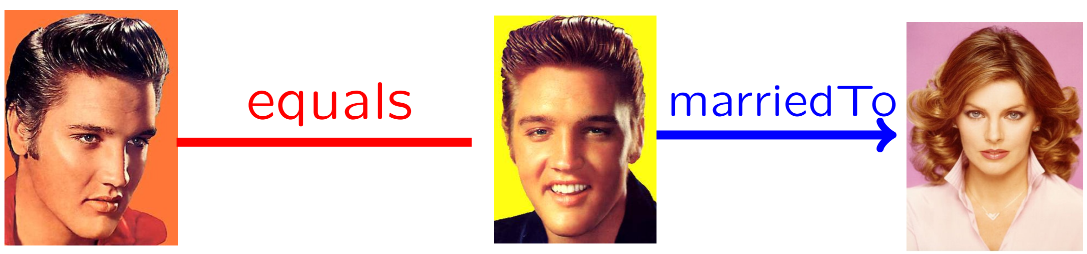




Conclusion

- PARIS matches relations, instances and schema holistically
- PARIS has no parameters to tune
- PARIS shows high precision and recall

- PARIS allows the YAGO Elvis to marry the DBpedia Priscilla



Happy End

Slides done with PowerLine, the free graphical SVG slide editor with Latex support

Backup Slides

with more details

Equality of Instances

x and x' shall be matched if

$\exists r, y, y' with r(x, y), r(x', y') : y \equiv y' \wedge ifun(r) large$

Equality of Instances

x and x' shall be matched if

$$\exists r, y, y' with r(x, y), r(x', y') : y \equiv y' \land ifun(r)$$

$$\neg\forall r, y, y' with r(x, y), r(x', y') : \neg(y \equiv y' \land ifun(r))$$

Equality of Instances

x and x' shall be matched if

$$\exists r, y, y' with r(x, y), r(x', y') : y \equiv y' \wedge ifun(r)$$

$$\neg \forall r, y, y' with r(x, y), r(x', y') : \neg (y \equiv y' \wedge ifun(r))$$

$$\neg \bigwedge_{r,y,y' with r(x,y), r(x',y')} \neg (y \equiv y' \wedge ifun(r))$$

Equality of Instances

x and x' shall be matched if

$$\exists r, y, y' with r(x, y), r(x', y') : y \equiv y' \wedge ifun(r)$$

$$\neg\forall r, y, y' with r(x, y), r(x', y') : \neg(y \equiv y' \wedge ifun(r))$$

$$\neg \bigwedge_{r,y,y' with r(x,y),r(x',y')} \neg(y \equiv y' \wedge ifun(r))$$

Let's compute the probability of this happening,

$$Pr(\neg \bigwedge_{r,y,y' with r(x,y),r(x',y')} \neg(y \equiv y' \wedge ifun(r)))$$

Equality of Instances

x and x' shall be matched if

$$\exists r, y, y' with r(x,y), r(x',y') : y \equiv y' \wedge ifun(r)$$

$$\neg\forall r, y, y' with r(x,y), r(x',y') : \neg(y \equiv y' \wedge ifun(r))$$

$$\neg \bigwedge_{r,y,y' with r(x,y), r(x',y')} \neg(y \equiv y' \wedge ifun(r))$$

Let's compute the probability of this happening,

$$Pr(\neg \bigwedge_{r,y,y' with r(x,y), r(x',y')} \neg(y \equiv y' \wedge ifun(r)))$$

(under independence assumptions)

$$= 1 - \prod_{r(x,y), r(x',y')} (1 - Pr(y \equiv y') Pr(ifun(r)))$$

Equality of Instances

x and x' shall be matched if

$$\exists r, y, y' with r(x,y), r(x',y') : y \equiv y' \wedge ifun(r)$$

$$\neg\forall r, y, y' with r(x,y), r(x',y') : \neg(y \equiv y' \wedge ifun(r))$$

$$\neg \bigwedge_{r,y,y' with r(x,y),r(x',y')} \neg(y \equiv y' \wedge ifun(r))$$

Let's compute the probability of this happening,

$$Pr(\neg \bigwedge_{r,y,y' with r(x,y),r(x',y')} \neg(y \equiv y' \wedge ifun(r)))$$

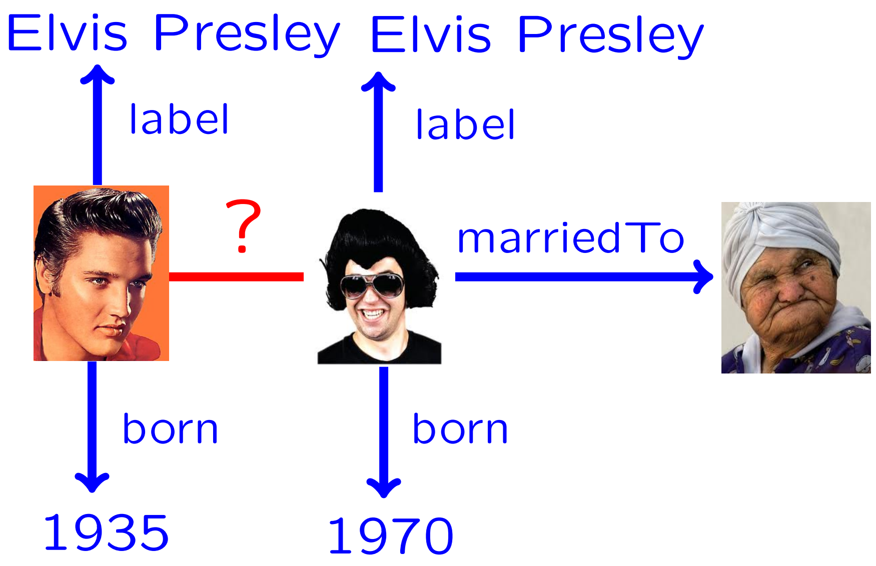
(under independence assumptions)

$$= 1 - \prod_{r(x,y),r(x',y')}(1 - Pr(y \equiv y')Pr(ifun(r)))$$

$$= Pr(x \equiv x')$$

Unequality of Instances

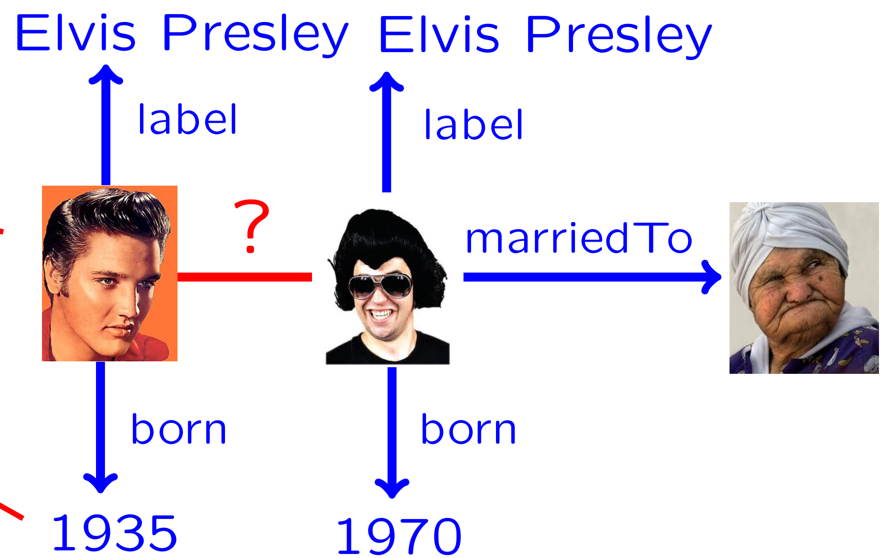
How do we guard against inequalities?



Unequality of Instances

How do we guard against inequalities?

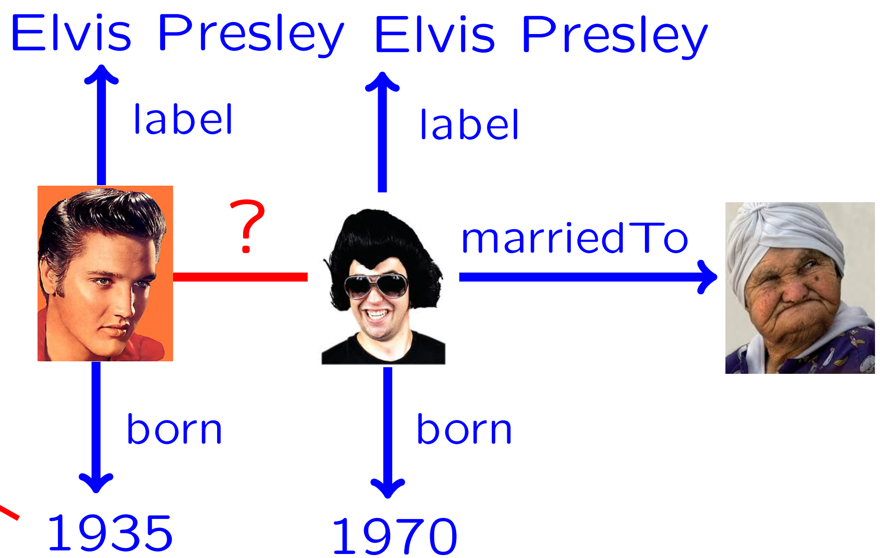
These cannot be equal,
because they have
a different value
for a functional relation.



Unequality of Instances

How do we guard against inequalities?

These cannot be equal,
because they have
a different value
for a functional relation.



The functionality is defined analogously to the inverse functionality.

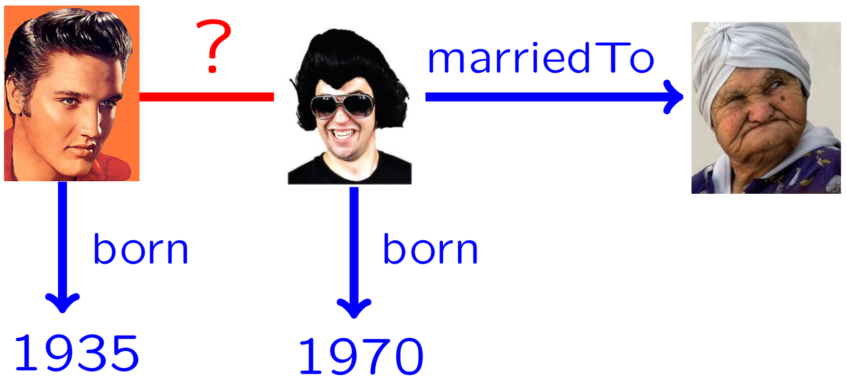
$$fun(bornIn) = 1$$

Unequality of Instances

Every value in one ontology should have a pendant in the other, weighted by the functionality:

$$Pr(x \equiv x') = \ldots \prod_{r(x,y)} (1 - fun(r) \prod_{r(x',y')} Pr(y \not\equiv y'))$$

This factor makes sure that
for every y, there is one
equivalent y'

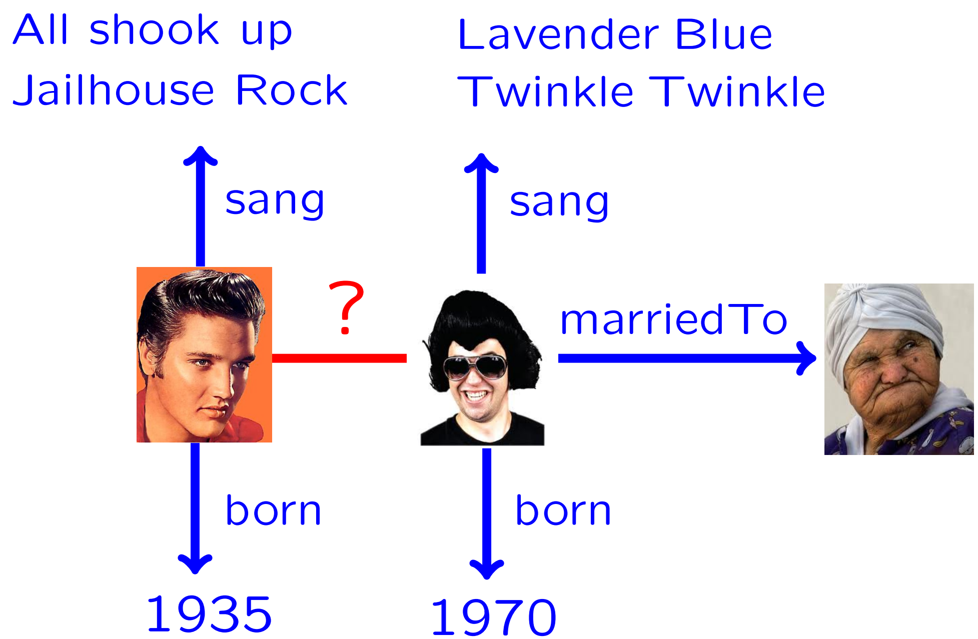


Unequality of Instances

Every value in one ontology should have a pendant in the other, weighted by the functionality:

$$Pr(x \equiv x') = ... \prod_{r(x,y)} (1 - fun(r) \prod_{r(x',y')} Pr(y \not\equiv y'))$$

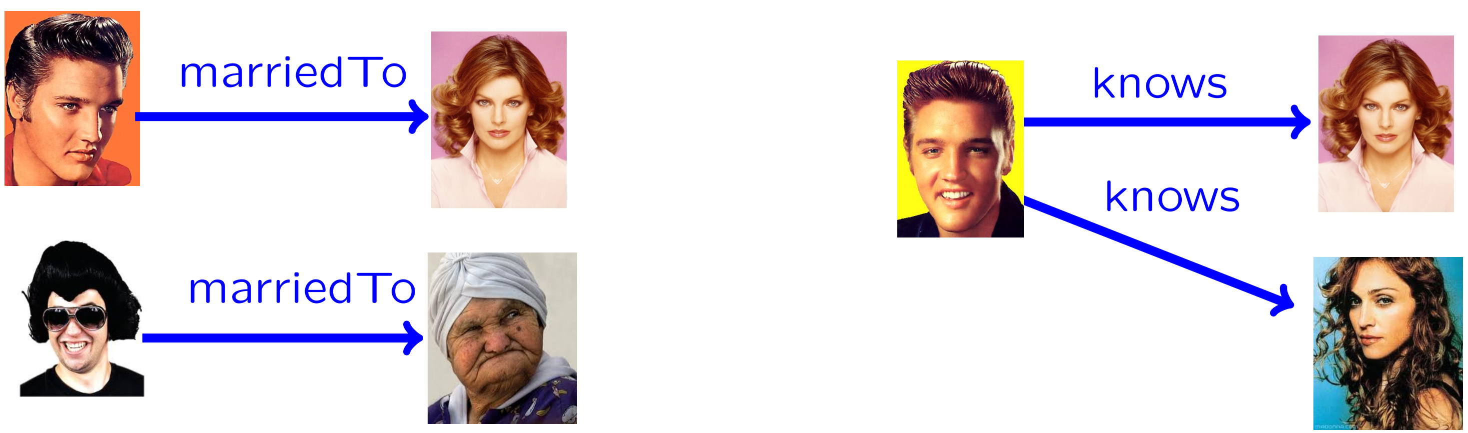
This factor makes sure that for every y, there is one equivalent y'



Equality of Relations

If every pair of one relation is a pair of another relation, then the first is a sub-property of the second.

marriedTo $\xrightarrow{\text{subpropertyOf}}$ knows

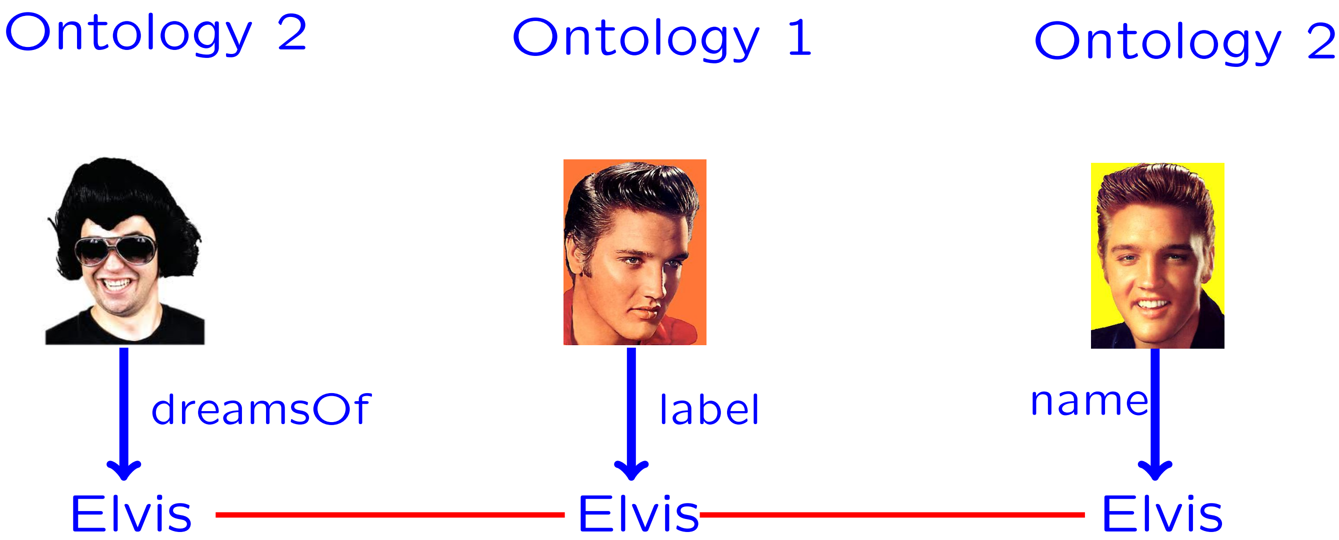


(Those in the intersection)

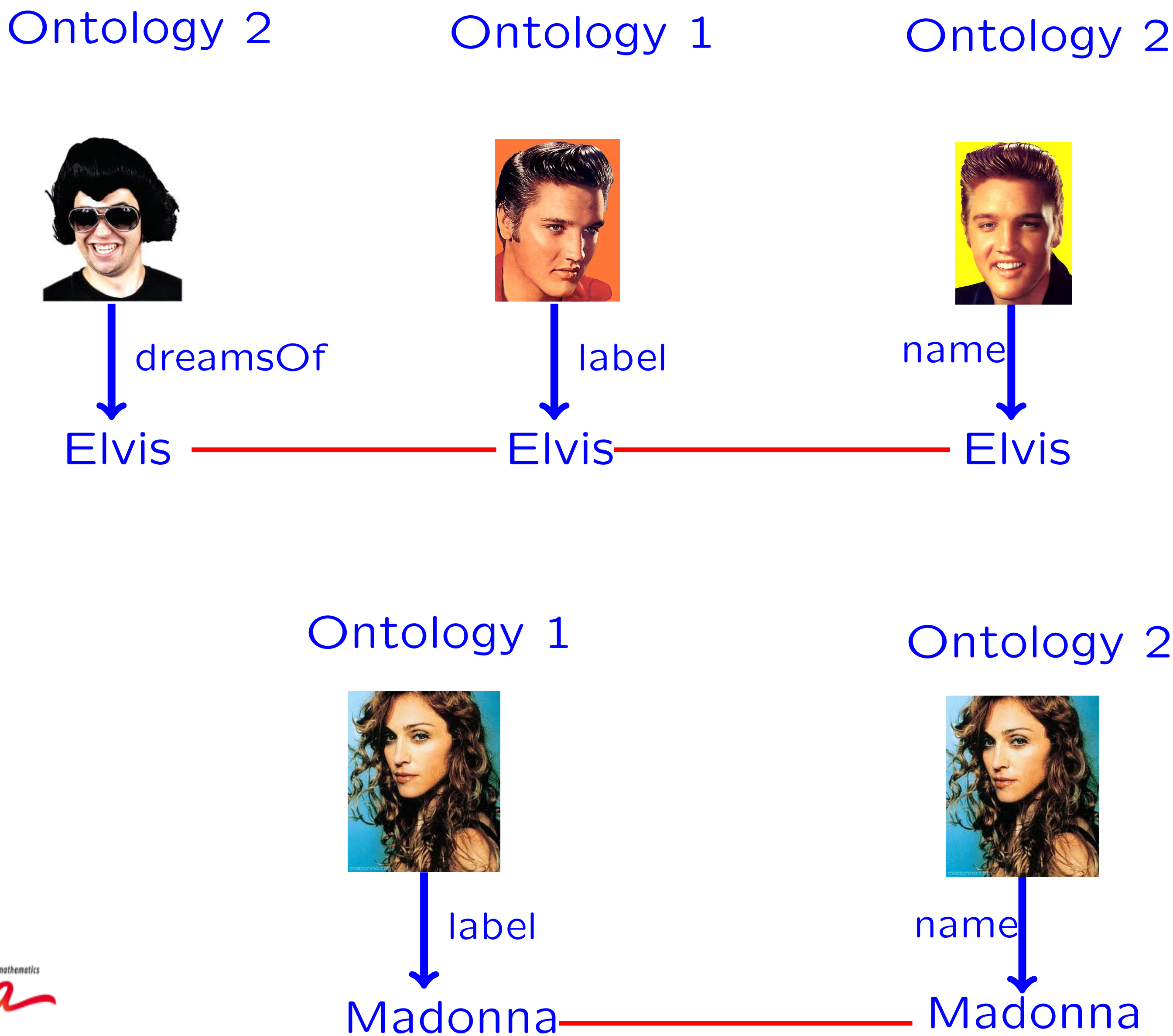
$$Pr(r \subseteq r') = \frac{\sum_{r(x,y)}(1-\prod_{r'(x',y')}(1-Pr(x\equiv x')\cdot Pr(y\equiv y')))}{\sum_{r(x,y)}(1-\prod_{x',y'}(1-Pr(x\equiv x')\cdot Pr(y\equiv y')))}$$

(Those that have a pendant)

Example

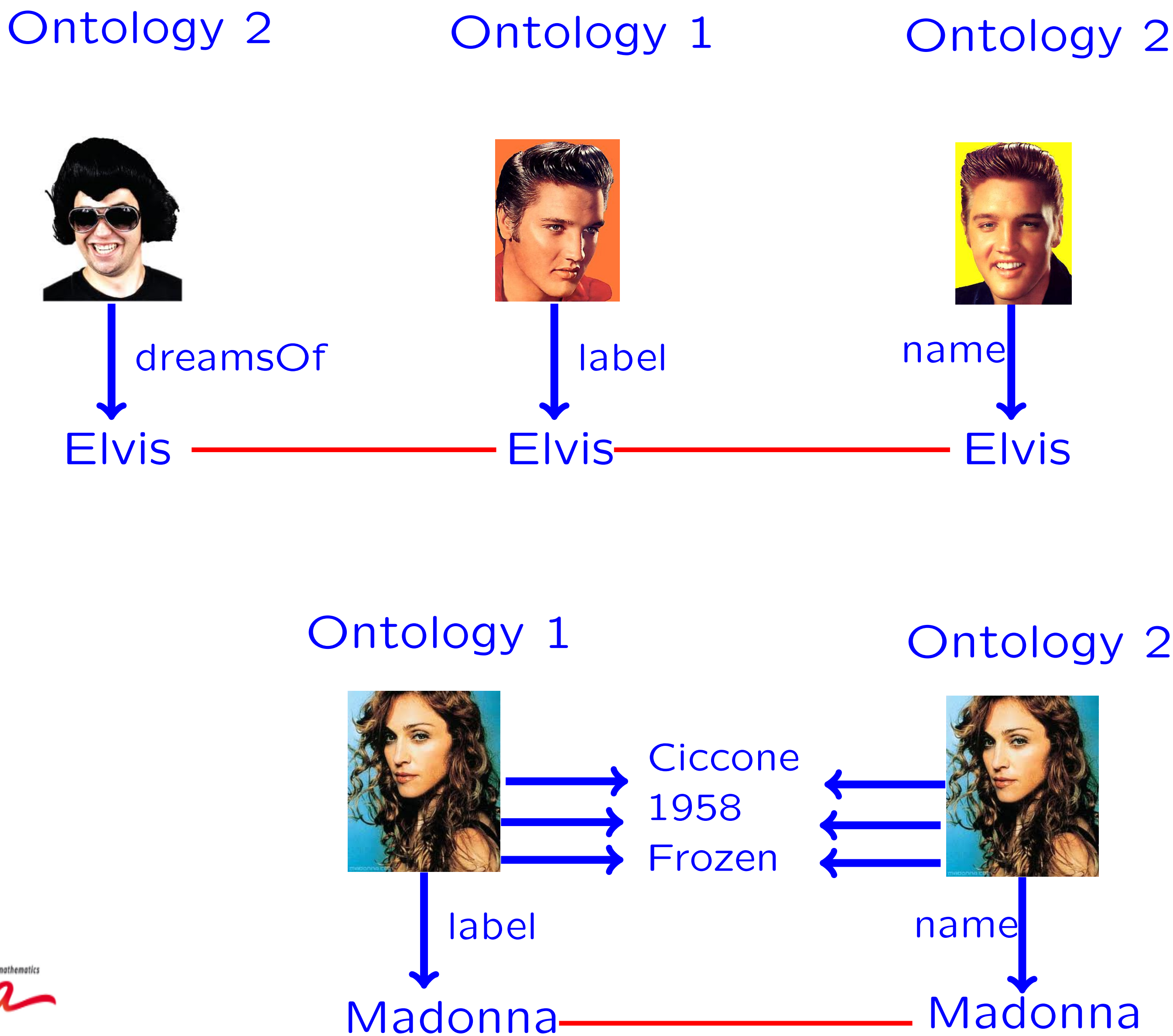

Ontology 2
Ontology 1
Ontology 2
dreamsOf
label
name
Elvis
Elvis
Elvis

Example

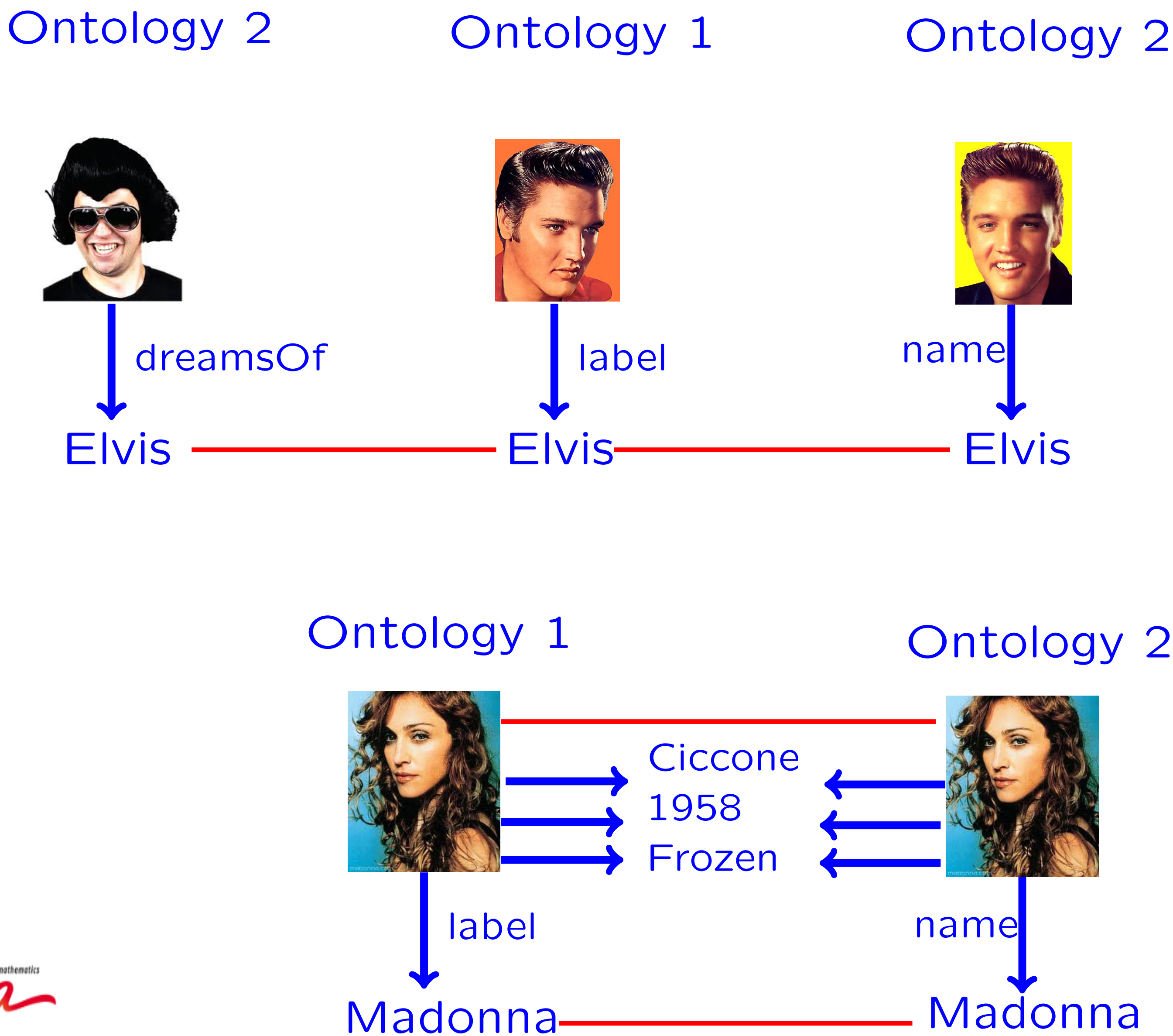


Ontology 2
Ontology 1
Ontology 2
dreamsOf
label
name
Elvis
Elvis
Elvis
Ontology 1
Ontology 2
label
name
Madonna
Madonna

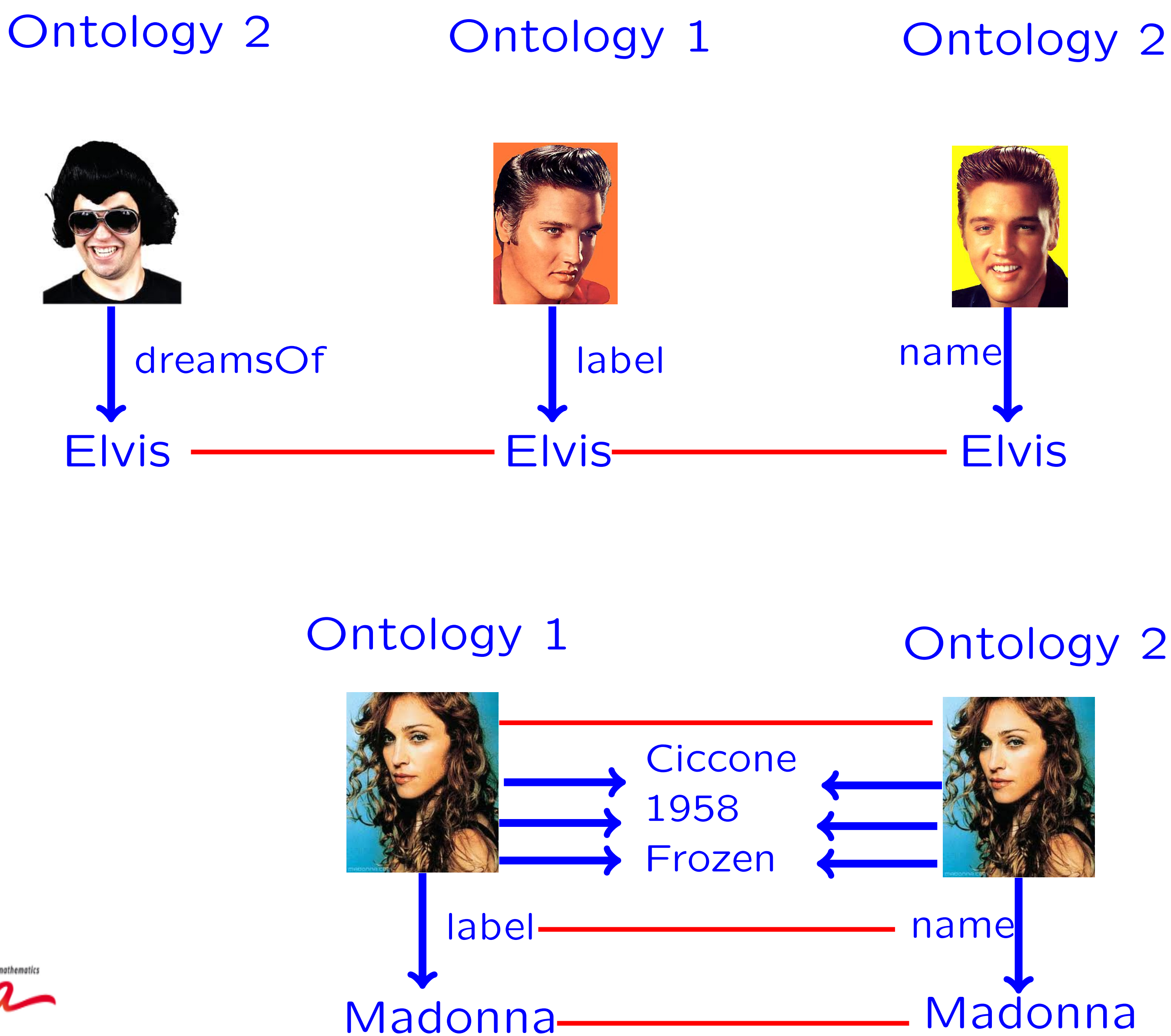
Example


Ontology 2
Ontology 1
Ontology 2
dreamsOf
label
name
Elvis
Elvis
Elvis
Ontology 1
Ontology 2
Ciccone
1958
Frozen
label
name
Madonna
Madonna

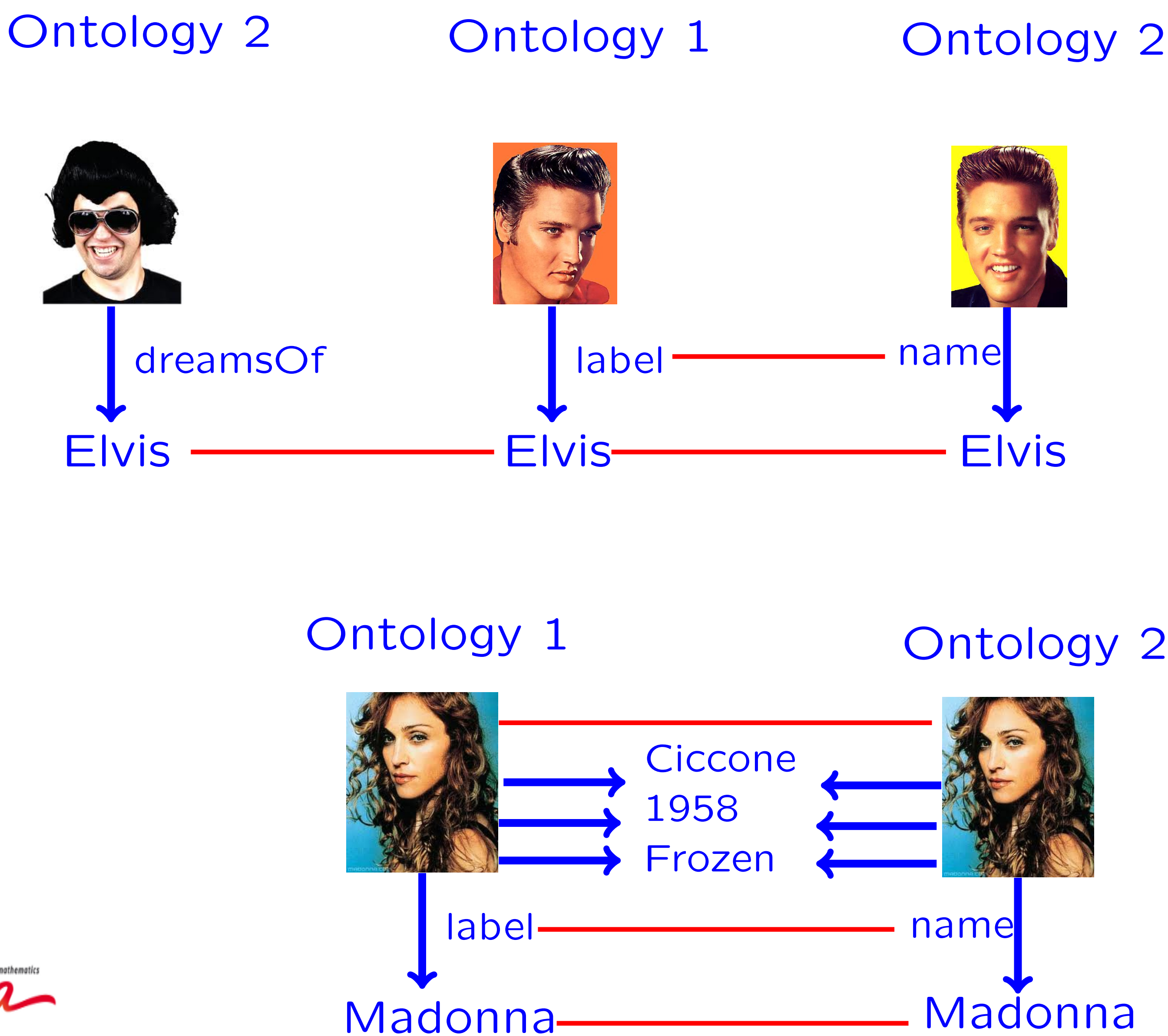
Example


Ontology 2
Ontology 1
Ontology 2
dreamsOf
label
name
Elvis
Elvis
Elvis
Ontology 1
Ontology 2
Ciccone
1958
Frozen
label
name
Madonna
Madonna

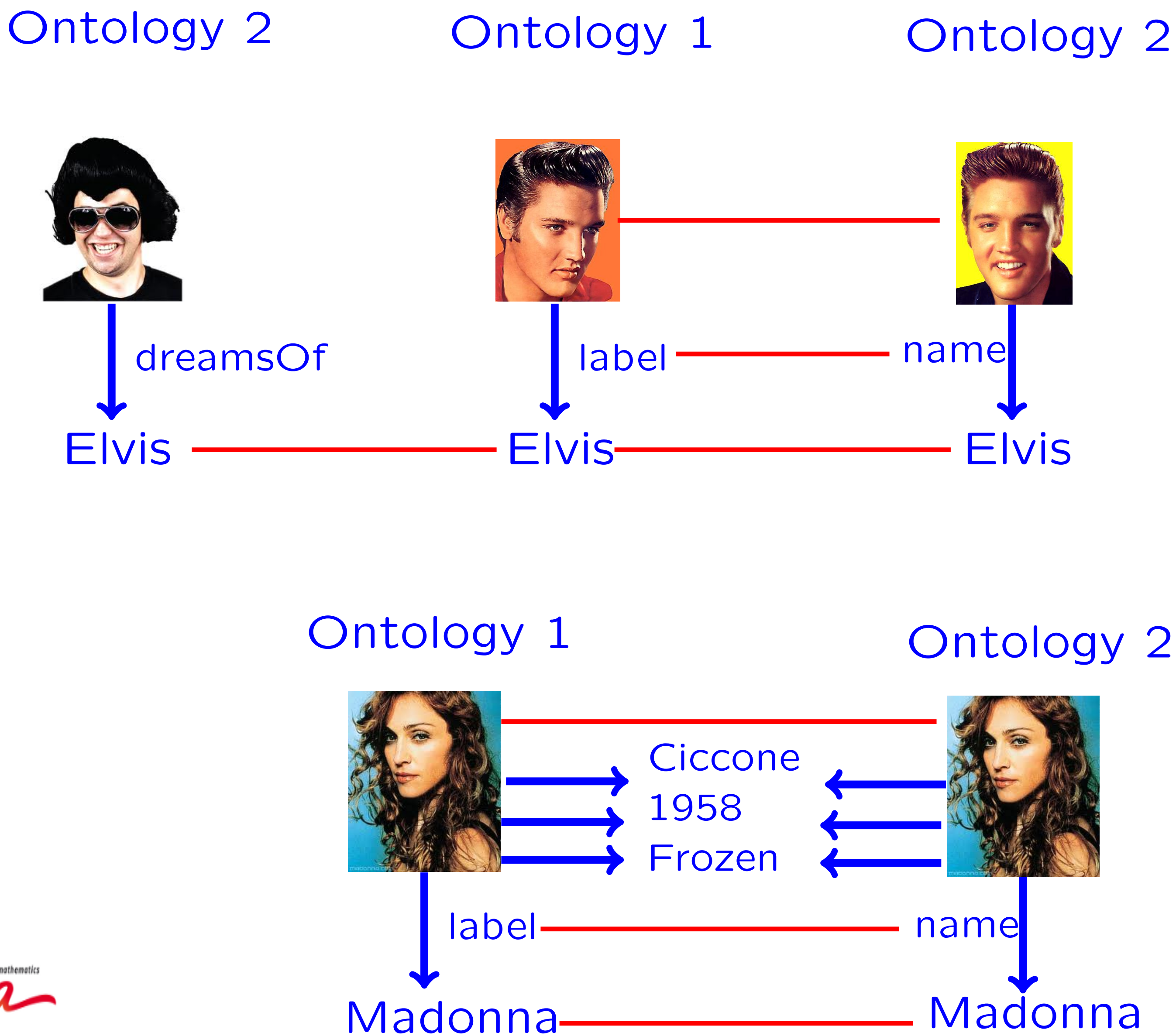
Example


Ontology 2
Ontology 1
Ontology 2
dreamsOf
label
name
Elvis
Elvis
Elvis
Ontology 1
Ontology 2
Ciccone
1958
Frozen
label
name
Madonna
Madonna

Example


Ontology 2
Ontology 1
Ontology 2
dreamsOf
label
name
Elvis
Elvis
Elvis
Ontology 1
Ontology 2
Ciccone
1958
Frozen
label
name
Madonna
Madonna

Example



Ontology 2
Ontology 1
Ontology 2
dreamsOf
label
name
Elvis
Elvis
Elvis
Ontology 1
Ontology 2
Ciccone
1958
Frozen
label
name
Madonna
Madonna