Knowledge Representation

Fabian M. Suchanek

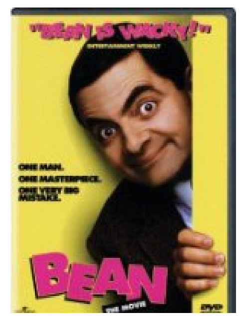
Overview

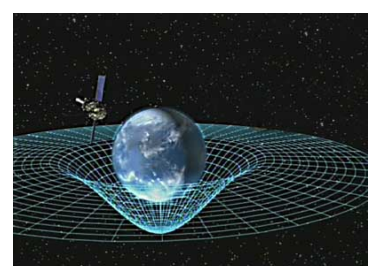
- Entities & Classes
- Relations
- Binary Relations
- Schema
- Knowledge Graphs
- N-ary Relations and Events
- Canonic Entities
- Reality

Entity

An entity (also: instance, resource)
is any particular object of the world
or imagination, be it abstract or concrete.



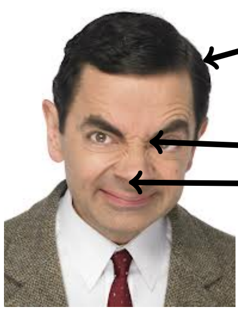



digressions>47

(The exact definition of “instance” is a philosophical conundrum. Don’t worry about it.)

Digression: Entities



Is this an entity?

Or this?

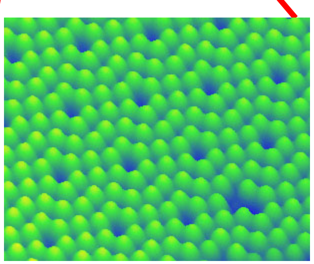
How many entities are there?

Digression: Entities

Is this an entity?

Or this?

How many entities are there?



Isn't everything just atoms?

Digression: Identity

Over time, all parts of a ship are replaced at some point of time. Then, is it still the same ship?

see: Theseus's ship on Wikipedia

Digression: Identity

Over time, all parts of a ship are
replaced at some point of time.
Then, is it still the same ship?

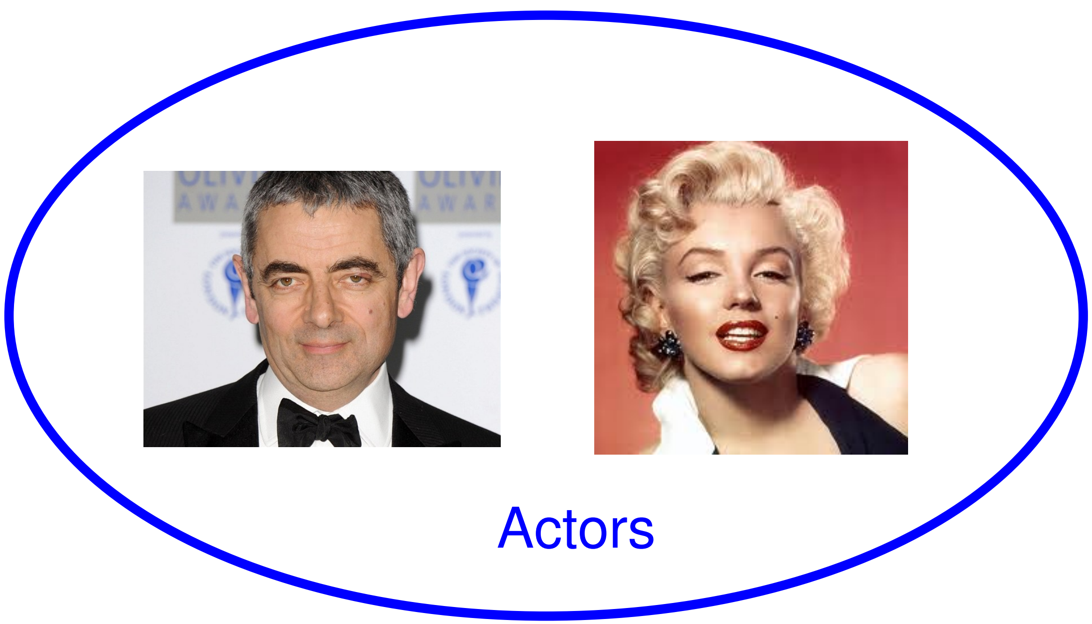
see: Theseus's ship on Wikipedia

Humans replace their cells every 7 years.

New York Times

Class

A class (also: concept) is a set of similar entities.



Cars

Cities

Rivers

Universities

Theories

...

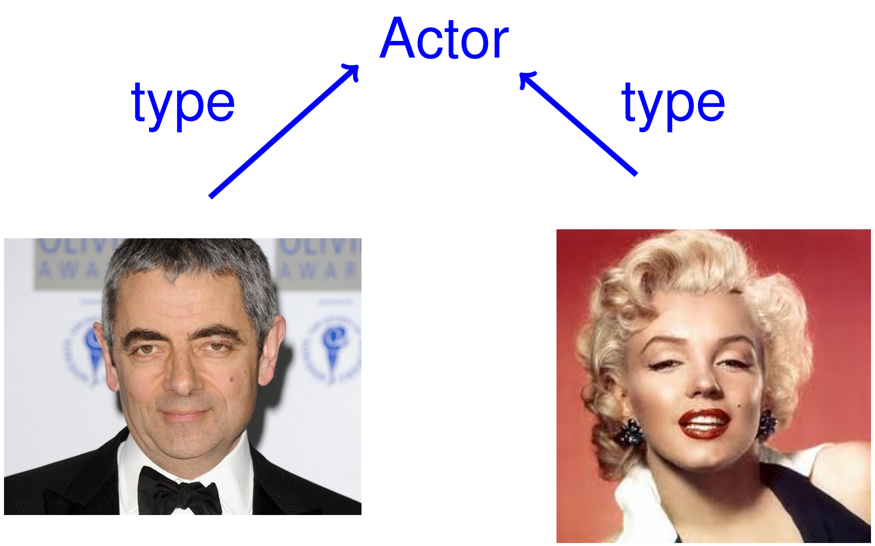
>instances

>instances,subclasse

(The exact definition of "class" is a philosophical conundrum, too)

Instance of a class

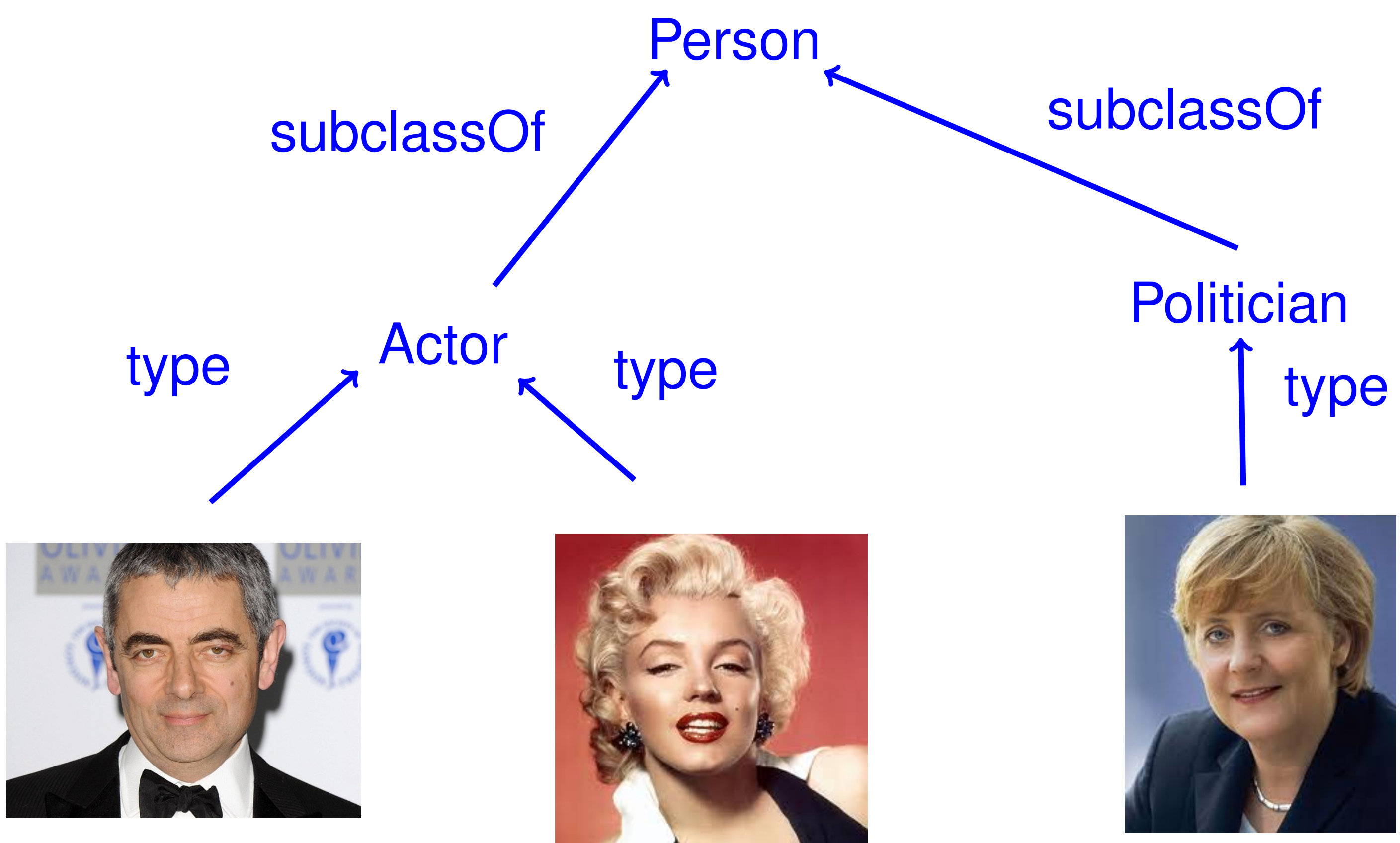
An entity is an instance of a class
(also: belongs to a class, has the type),
if the entity is an element of that class.



Def: Subclass, Taxonomy

A class is a subclass of another class, if all instances of the first class are also instances of the second class.

A taxonomy is a hierarchy of classes.



Cheat Sheet

If we can say...	then...
• "a/an X", "every X"	X is a class
• "Xs" (plural)	X is a class
• "This is X"	X is an entity
• "X is a Y"	X is an inst. of Y
• "Every X is a Y"	X is a subclass of Y

Cheat Sheet

If we can say...	then...
• “a/an X”, “every X”	X is a class
• “Xs” (plural)	X is a class
• “This is X”	X is an entity
• “X is a Y”	X is an inst. of Y
• “Every X is a Y”	X is a subclass of Y

Try it out: car, Mr. Bean, Coli bacteria, Ford, time

>task

Task: Subclasses

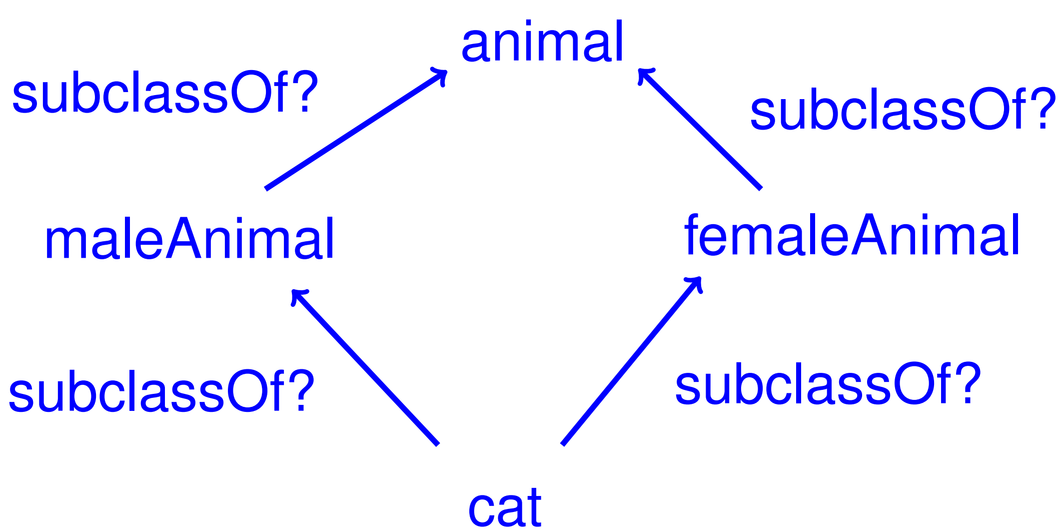
Draw a picture of instances and subclasses in the animal kingdom.

How do you deal with “male” and “female”?

Task: Subclasses

Draw a picture of instances and subclasses in the animal kingdom.

How do you deal with “male” and “female”?



Intuition: Relations

A relation is like a table.

Relation “born”:

Person	City	Year
Atkinson	Consett	1955
Monroe	Los Angeles	1926
...		

Def:Relation

A relation (also: predicate) over classes is a subset of their cartesian product. The classes are called the domains of the relation.
The number of classes is called the arity of the relation.

$$R \subseteq C_1 \times C_2 \times ... \times C_n$$

$$born \subseteq person \times city \times year$$

$$born = \{< Atkinson, Consett, 1955 >,$$

$$< Monroe, LosAngeles, 1926 >, ...\}$$

Digression: Semantics

A relation is any subset of the cartesian product. It does not have to correspond to a real-world relation. Its name is arbitrary.

$$born = \{< Atkinson, Consett, 1955 >, \\ < Monroe, LosAngeles, 1926 >, ...\}$$

Digression: Semantics

The semantics/denotation of a symbol is
the “meaning” of the symbol,
i.e., the real world entity or tuples.

$$\underbrace{D(Atkinson)}_{\text{Syntax}} = \underbrace{RowanAtkinson}_{\text{Semantics}}$$

$$D(born) = \{< x, y, z > | xwasborninyinyearz\}$$

Digression: Semantics

But these are again symbols.

$$D(born) = \{< x, y, z > |xwasborninyinyearz\}$$

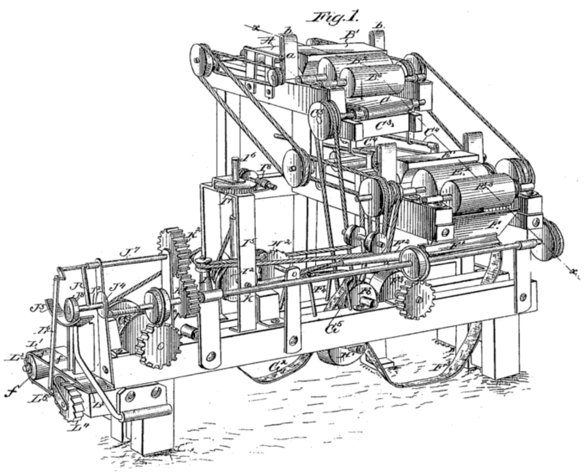
What is their semantics?

Digression: Semantics

$D(Atkinson was born in Consett)$

Digression: Semantics

$D(AtkinsonwasborninConsett)$



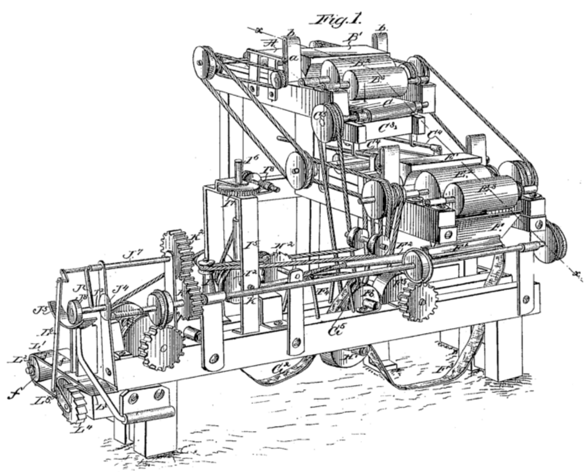
Linguistic parsing

+ Montague semantics

$= (\lambda xy < x, y > \in born) AtkinsonConsett$

Digression: Semantics

$D(AtkinsonwasborninConsett)$



Linguistic parsing

+ Montague semantics

$$= (\lambda xy < x, y > \in born) Atkinson Consett$$

$$= < Atkinson, Consett > \in born$$

Digression: Semantics

$$
\begin{tikzcd}
 & D(born) \arrow[dr] & \\
born \arrow[ur] & & \{Peopleborn\} \arrow[dl] \\
 & \lambda born \arrow[ul] &
\end{tikzcd}
$$

One man's semantics
is another man's syntax.
All we do is manipulating symbols.
There is no connection to the real world.

Def: Binary Relation, Triple

A binary relation is a relation of arity 2.

$$bornInCity \subseteq person \times city$$

For binary relations, the first class is called the domain and the second class is called the range.

An element of a binary relation is called a fact (or: triple), and we usually visualize it by an arrow:

$$bornInCity(Atkinson, Consett)$$

$\xrightarrow{\text{bornInCity}}$

functions>3

The first argument of a fact is the subject, the second the object.

Def: Inverse

The inverse of a binary relation r is a relation r', such that $r'(x, y)$ iff $r(y, x)$.

$livesInCity \subseteq person \times city$

$livesInCity(Atkinson, Consett)$

inverses of each other

$hasInhabitant \subseteq city \times person$

$hasInhabitant(Consett, Atkinson)$

Def: Function

A function (also: functional relation) is a binary relation that has at most 1 object for each subject.

$$r\ functional \equiv \forall x : |\{y : r(x,y)\}| \leq 1$$

hasNationality

hasSpouse (in medieval ages)

hasNumberOfTeeth

Def:Inverse Functional Rel.

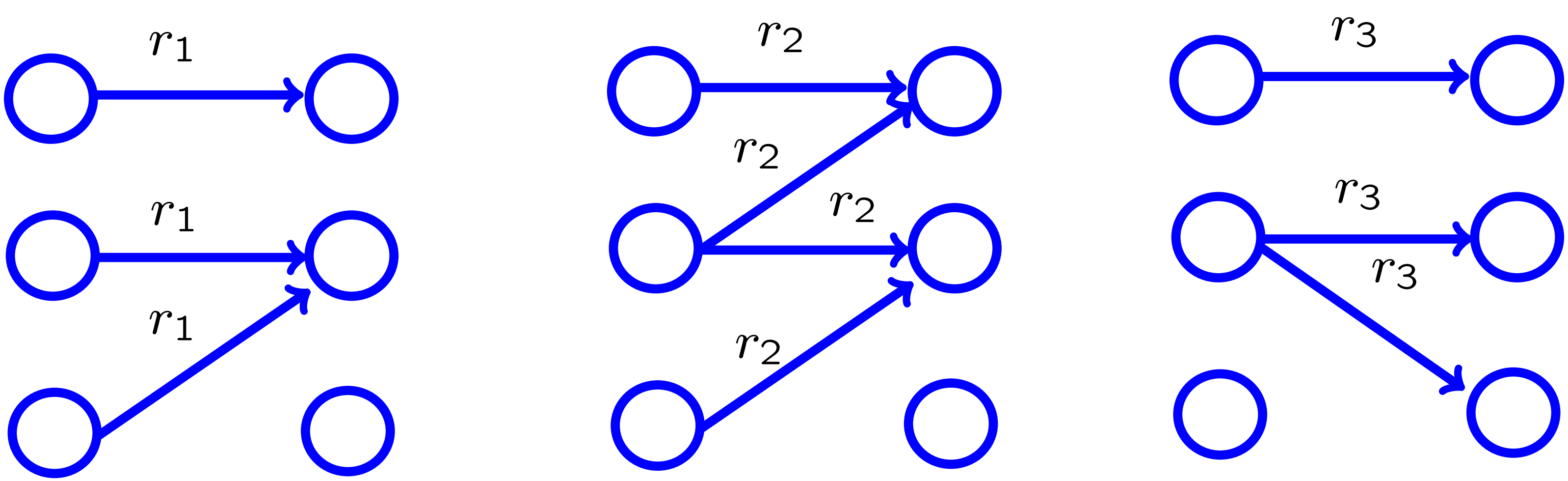
An inverse functional relation is a relation whose inverse is functional.

$$r inv.functional \equiv \forall y : |\{x : r(x,y)\}| \leq 1$$

hasCitizen
hasSpouse (in medieval ages)

Task: Functions

Which of the following relations are functional?



Digression: Equality

If two entities share the same object of an inverse functional relation, they are equal.

$$hasEmail(Bean, me@bean.com)$$

$$hasEmail(MrBean, me@bean.com)$$

$$\Rightarrow MrBean = Bean$$

$$born(Bean, 1955)$$

$$born(MrBean, 1955)$$

(Nothing follows)

Def: Name

A name (also: label) of an entity is a human-readable string attached to that entity.

Entity

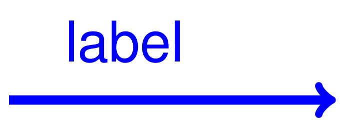
The entity is called the meaning of the name.

Name

“Mr. Bean”

Label

label is a binary relation that holds between an entity and its name.



“Mr. Bean”

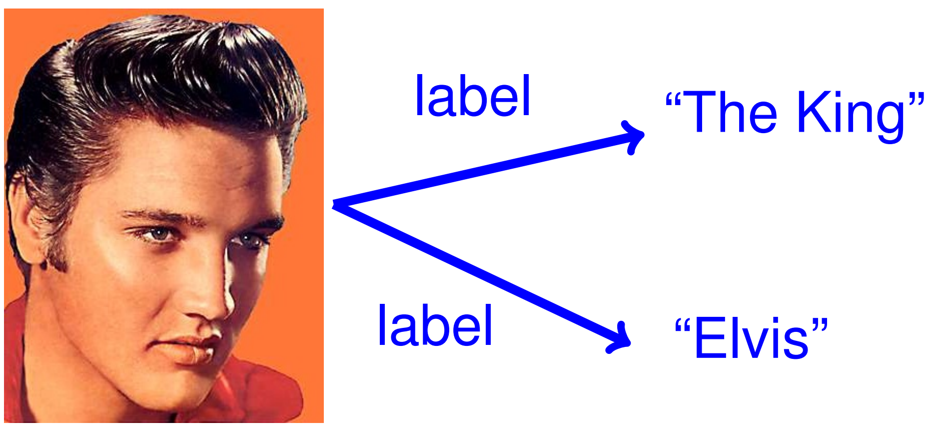
$$= label(MrBean, "Mr.Bean")$$

Def:Synonymy

If an entity has multiple names,

the names are called synonymous.

(The adjective for the names is “synonymous”, each name is a “synonym”, the phenomenon is called “synonymy”)

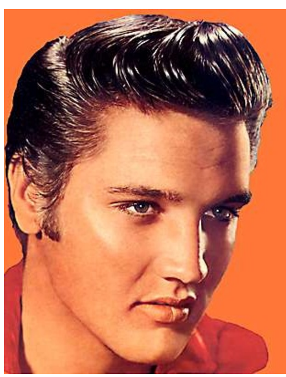


Def:Ambiguity

If a name is attached to multiple entities,

the name is called ambiguous.

(The adjective for the names is “ambiguous”, the phenomenon is called “ambiguity”)



label

“The King”

label

>classes

>Task

Task:Names

List some entities with their names,

some ambiguous names,

and some synonyms.

Classes as binary relations

One way to represent a class is by the binary relations $type, subclassOf$.

$$type \subseteq entity \times class$$

$$type(Atkinson, actor)$$

$$subclassOf \subseteq class \times class$$

$$subclassOf(actor, person)$$

Person

$\uparrow$ subclassOf

Actor

$\uparrow$ type

(Watch out: here, “entity” is the class of entities, “class” is the set of class names. See later in the lecture.)

Digression: Classes and Relations

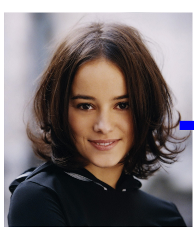
A fact can be modeled as a class or as a relation.

type

woman

gender

female



type

singer

job

singer

>domain

Domains as binary relations

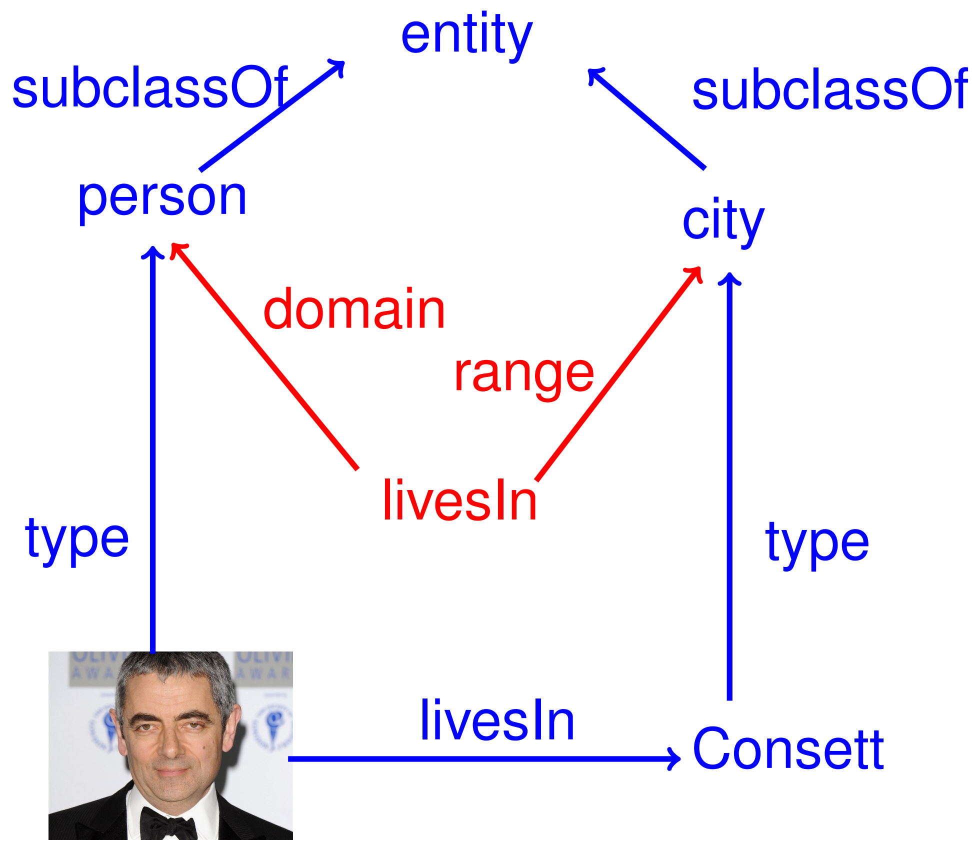
The domain and range of relations can be expressed by binary relations $domain$ and $range$.

$domain \subseteq relation \times class$

$domain(livesIn, person)$

$range \subseteq relation \times class$

$range(livesIn, city)$

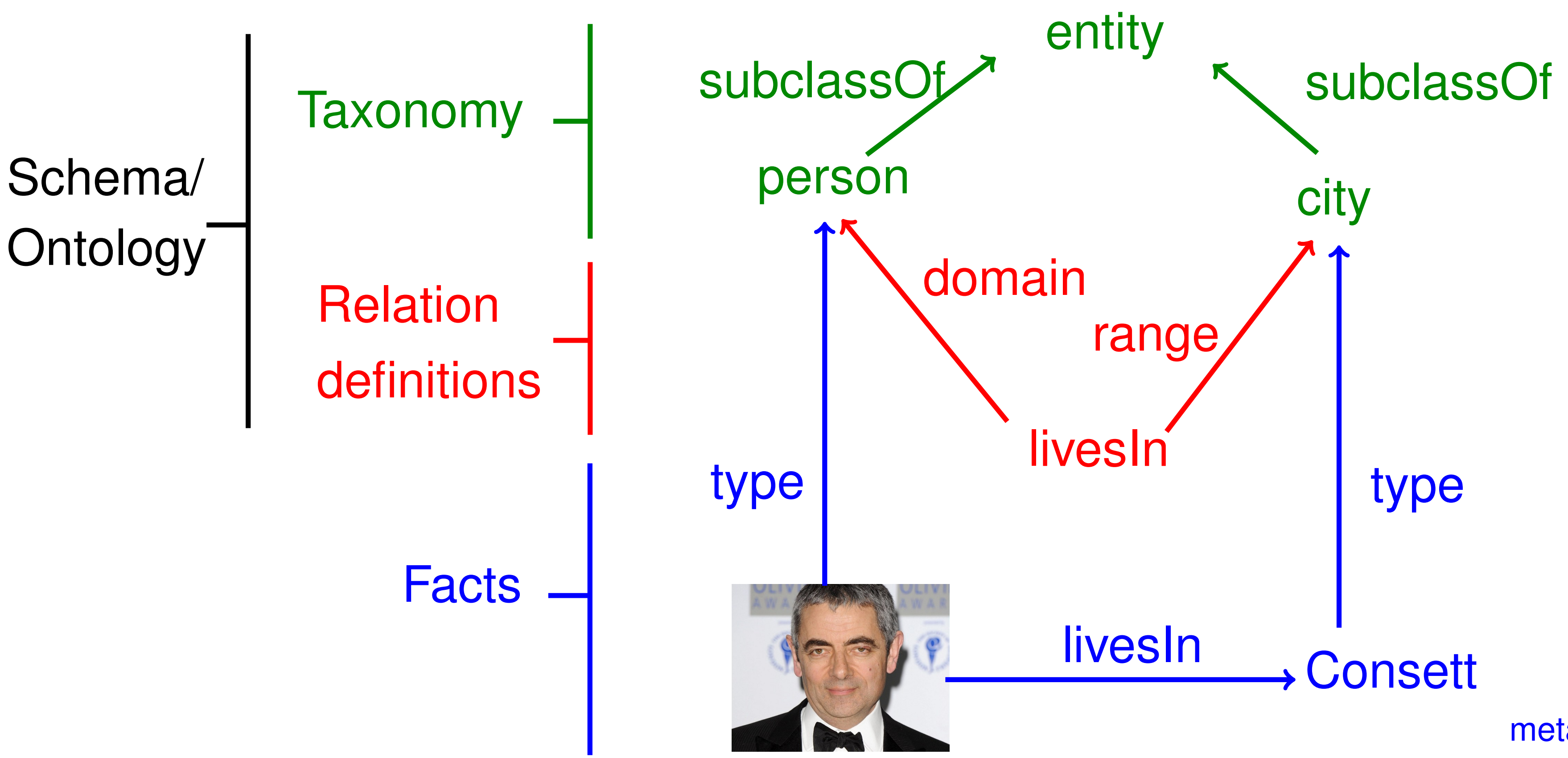


(Watch out: here, “relation” is the set of relation names. See later in this lecture.)

Def: Schema

A schema (or: ontology) consists of

- a taxonomy (= set of classes with their subclassOf-links)
- relation definitions (= a set of relations with domains and ranges)



meta>48
>meta&graph

Task: Schema

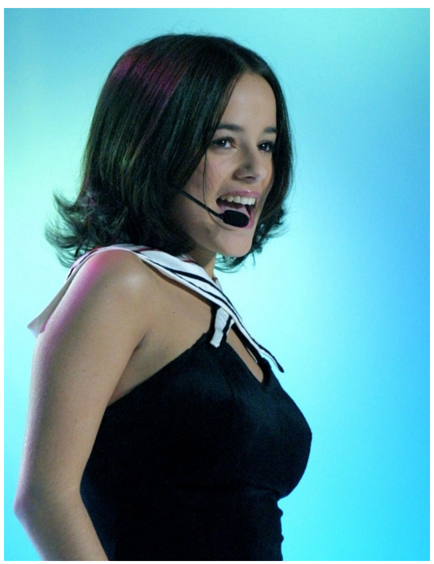
Define a schema for the domain of politics.

In that schema, express that Hollande is the president of France, Merkel is the chancellor of Germany, and that Merkel loves Hollande.

Digression: Classes as arguments

A relation tuple contains entities.

Can it also contain classes?



$$type(Alizee, singer)$$

Entity Class

?

Digression: Class Entity

A class entity is an entity that represents the class.

The class entity "Marianne"...

... represents the class
"French People"

$$frenchPeople = \{Hollande, Alizee, ...\}$$

Digression: The Class Class

"Class" is the class of all class entities.

$$class = \{car, city, river, singer, actor, ...\}$$

This class can appear in relations

$$type \subseteq entity \times class$$

$$type = \{< Alizee, singer >, < Atkinson, actor >, ...\}$$

Digression: Type

Type is a binary relation, which contains $<$x,y$>$ if x is an instance of the class represented by y.

$$type \subseteq entity \times class$$

$$type(Alizee, Singer)$$

$$type(Elvis, livingPeople)$$

Digression: SubClassOf

SubClassOf is a binary relation on class entities, which contains <x,y> if the class represented by x is a subclass of the class represented by y.

$$subClassOf \subseteq class \times class$$

$$subClassOf(Singers, Persons)$$

$$subClassOf(Cars, Vehicles)$$

Task>46

Digression: Task: Class entities

Draw a knowledge graph with the relations subClassOf and type.

Example: RDF

Digression: Domain and Range

A relation entity is an entity that represents a relation.

$relation$ is the class of relation entities.

$domain$ is a relation on binary relation entities and class entities, which contains <x,y> if the domain of the relation represented by x is the class represented by y.

$$domain \subseteq relation \times class$$

$$domain(bornInCity, Person)$$

$range$ is defined like $domain$ and identifies the range of a relation.

$$range \subseteq relation \times class$$

$$range(BornInCity, City)$$

Task>48

Digression: Task: Meta Relations

Draw a knowledge graph with the relations $domain$ and $range$.

Can $domain$ and $range$ appear as nodes?

Def:Knowledge Graph

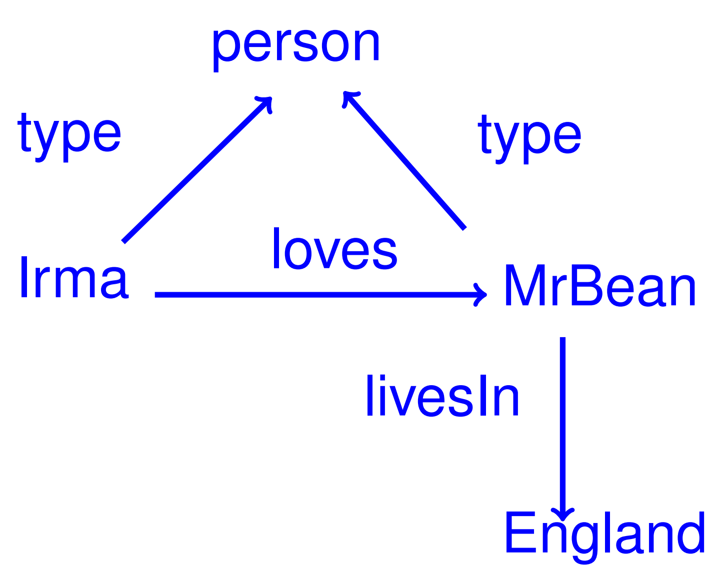
A knowledge graph (also: Entity-Relationship graph, Knowledge base, KB) is a directed labeled multi-graph that has an edge x->y with label r, iff r(x,y).

$loves(Irma, MrBean)$

$type(Irma, person)$

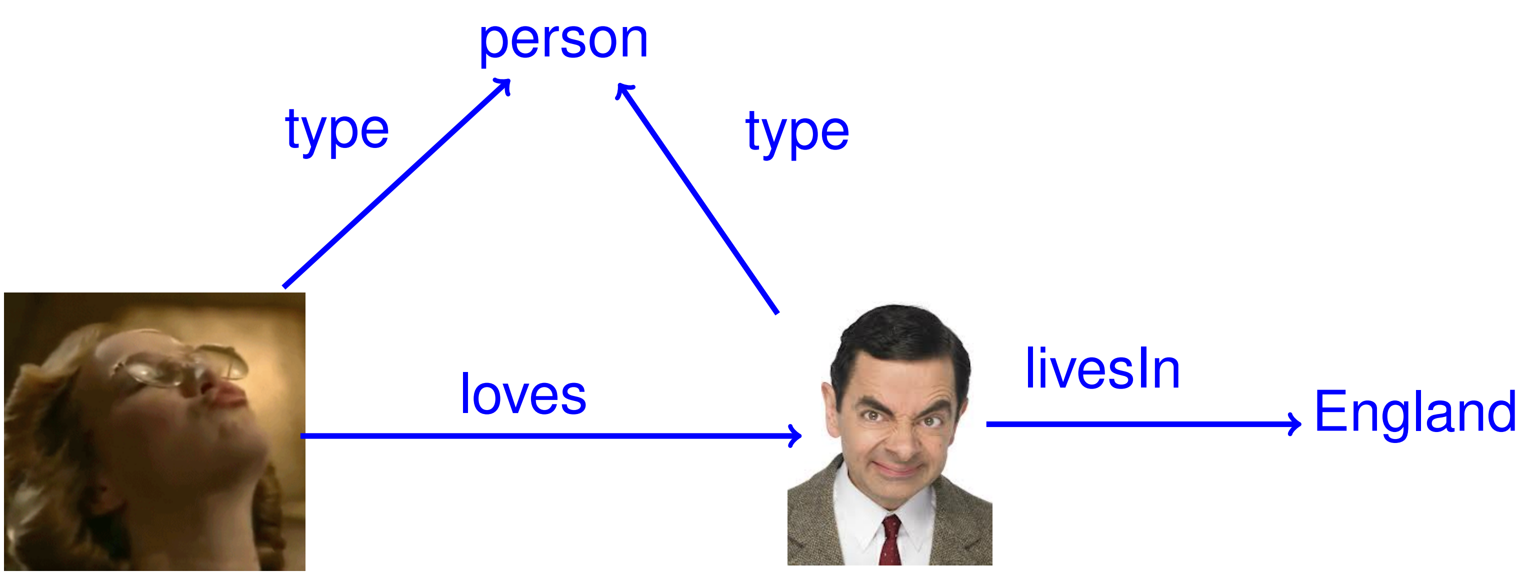
$type(MrBean, person)$

$livesIn(MrBean, England)$



Def:Knowledge Graph

A knowledge graph (also: Entity-Relationship graph, Knowledge base, KB) is a directed labeled multi-graph that has an edge x->y with label r, iff r(x,y).



>triple store

Def: Triple Store

A triple store is a table that contains
a KB of binary relations in the form
of 3 columns: subject, relation, object.

Subject	Relation	Object
Irma	loves	MrBean
Irma	type	person

(The middle column is often called “Predicate”)

Fact Representations

Binary relations can be represented as

- relation

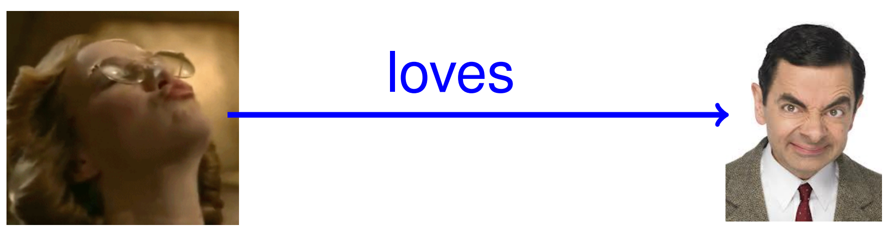
  $loves \subseteq person \times person$

  $loves(Irma, MrBean)$

- table

loves	Person	Person
	Irma	MrBean

- graph



- triple store

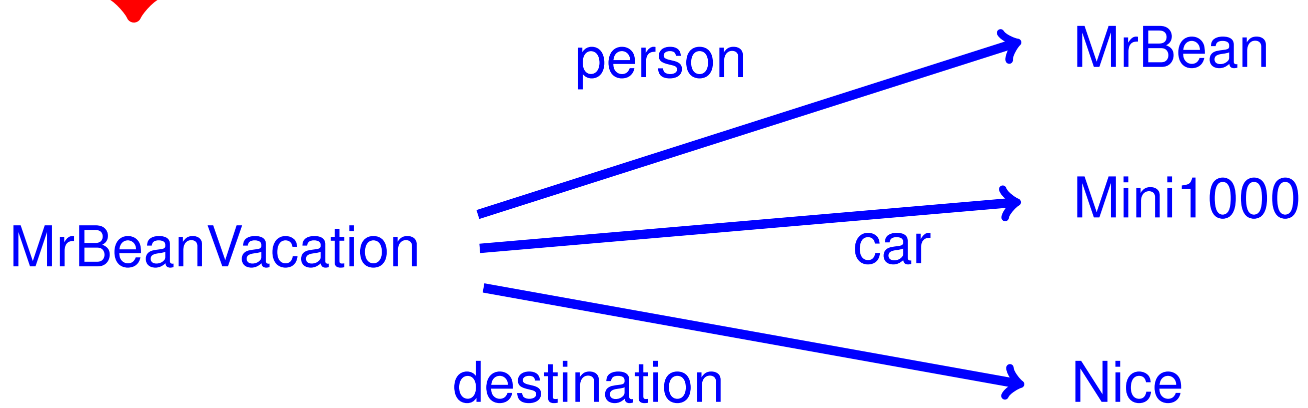
Subject	Relation	Object
Irma	loves	MrBean

events>

n-ary facts as binary facts

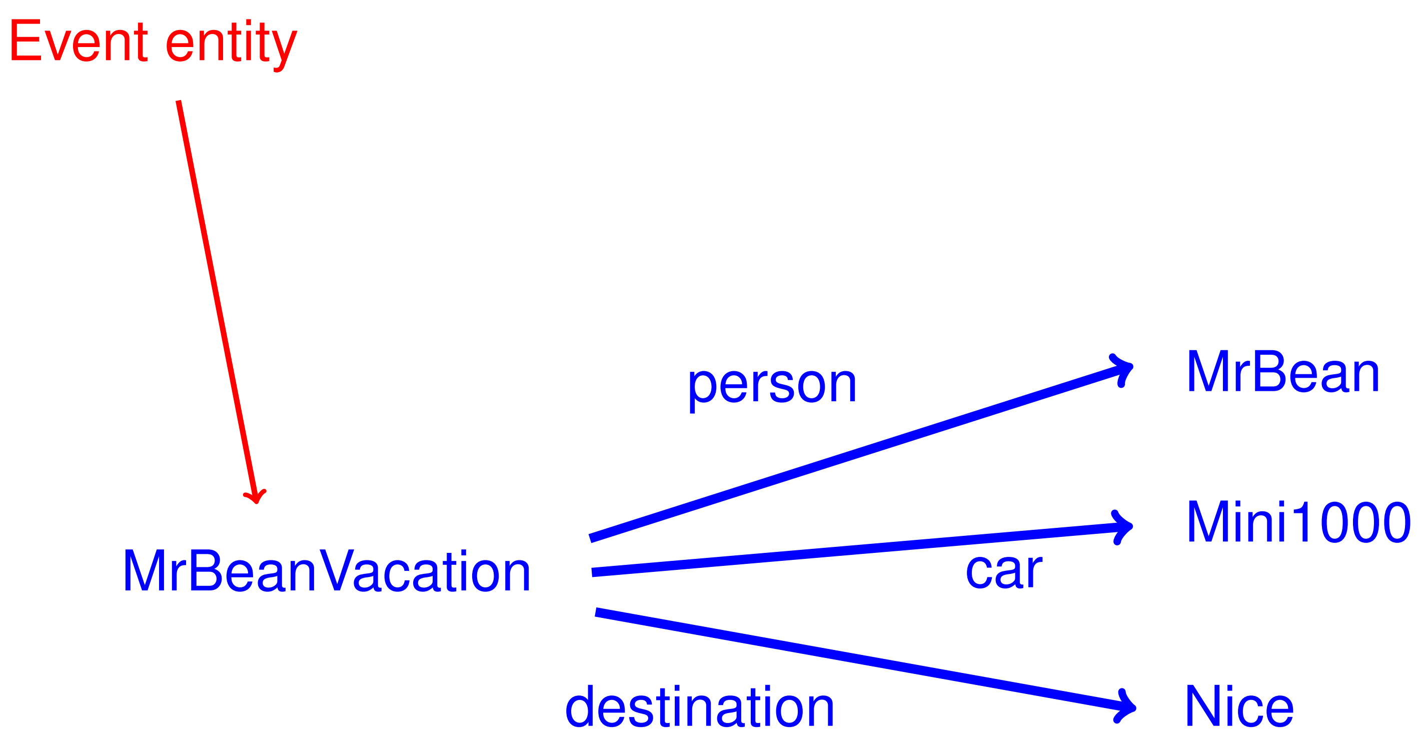
Every n-ary fact can be represented as binary facts.

drives	Person	Car	Destination
	MrBean	Mini1000	Nice



Def: Event Entity

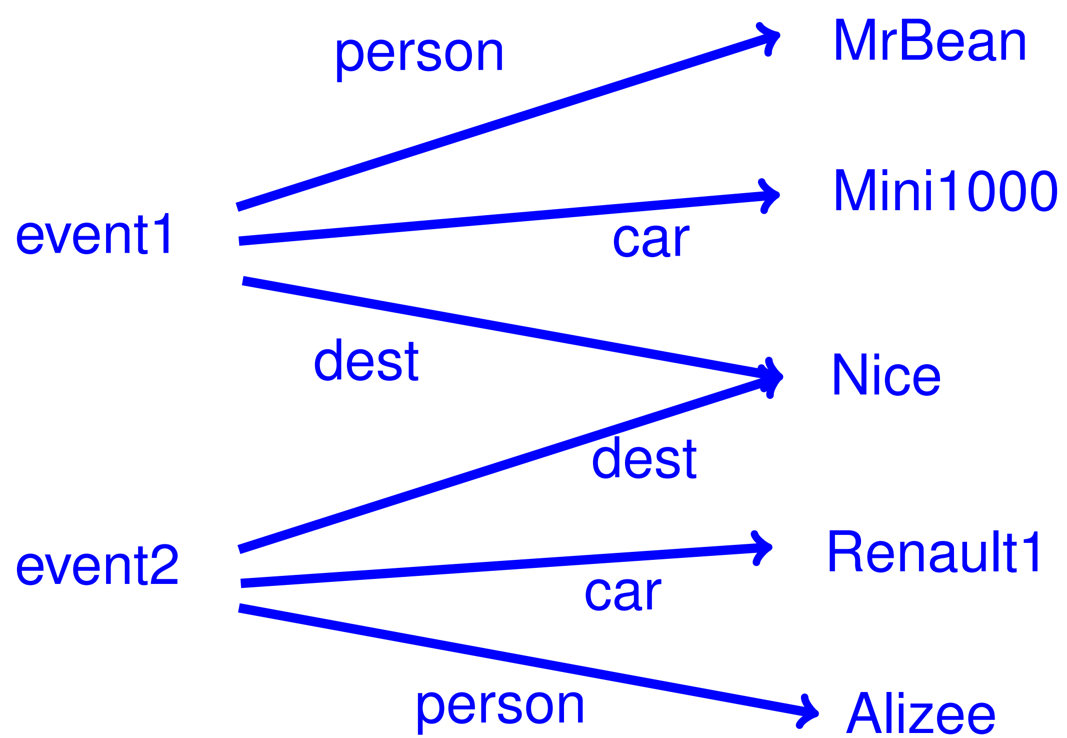
An event entity represents an n-ary fact.



Event Entities

drives	Person	Car	Destination
	MrBean	Mini1000	Nice
	Alizee	Renault1	Nice

Every row becomes a new event entity with n outgoing edges.



>task

Task: Event Entities

Draw a knowledge graph for the following facts.

Irma loves Mr. Bean since 1955.
Mr. Bean drives with Irma to the cinema.
Irma and Mr. Bean watch “Titanic”.
The movie is about the trip of the ship
“Titanic” from Europe to New York.

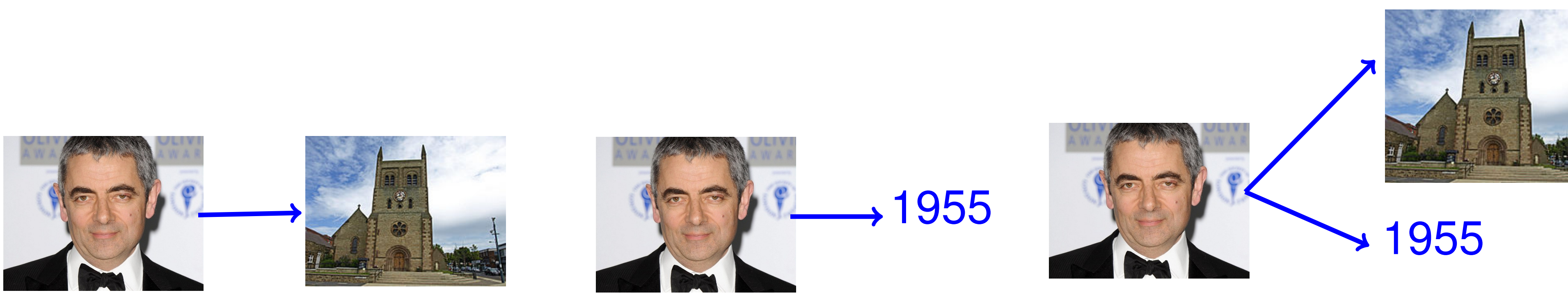
(There may be multiple solutions)

Binary relations are flexible

n-ary relations enforce the presence of all arguments:

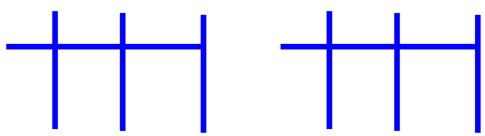
born	Person	City	Year
	Atkinson	Consett	1955

Binary relations don't:



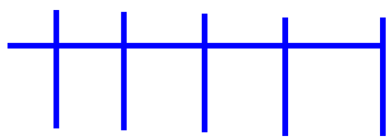
Binary vs n-ary

Binary and n-ary relations can represent the same facts.



binary

- more relations
- less arity
- more flexibility



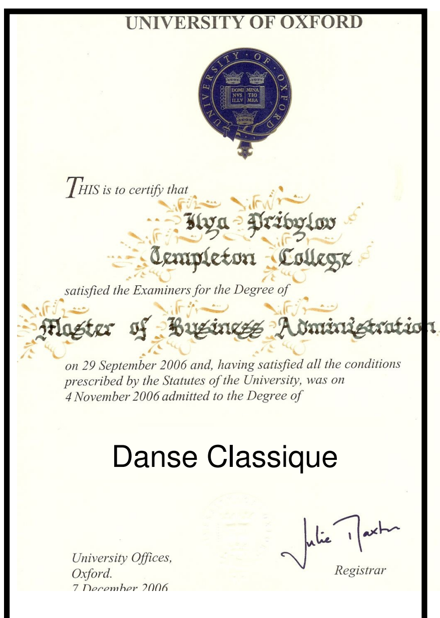
n-ary

- less relations
- more arity
- more control

Reify$>$63
Canonicity$>$73
Digression$>$77

Reified statements

A reified statement is an entity that represents a statement. This phenomenon is called reification.



represents

Alizee

completed

Dance School

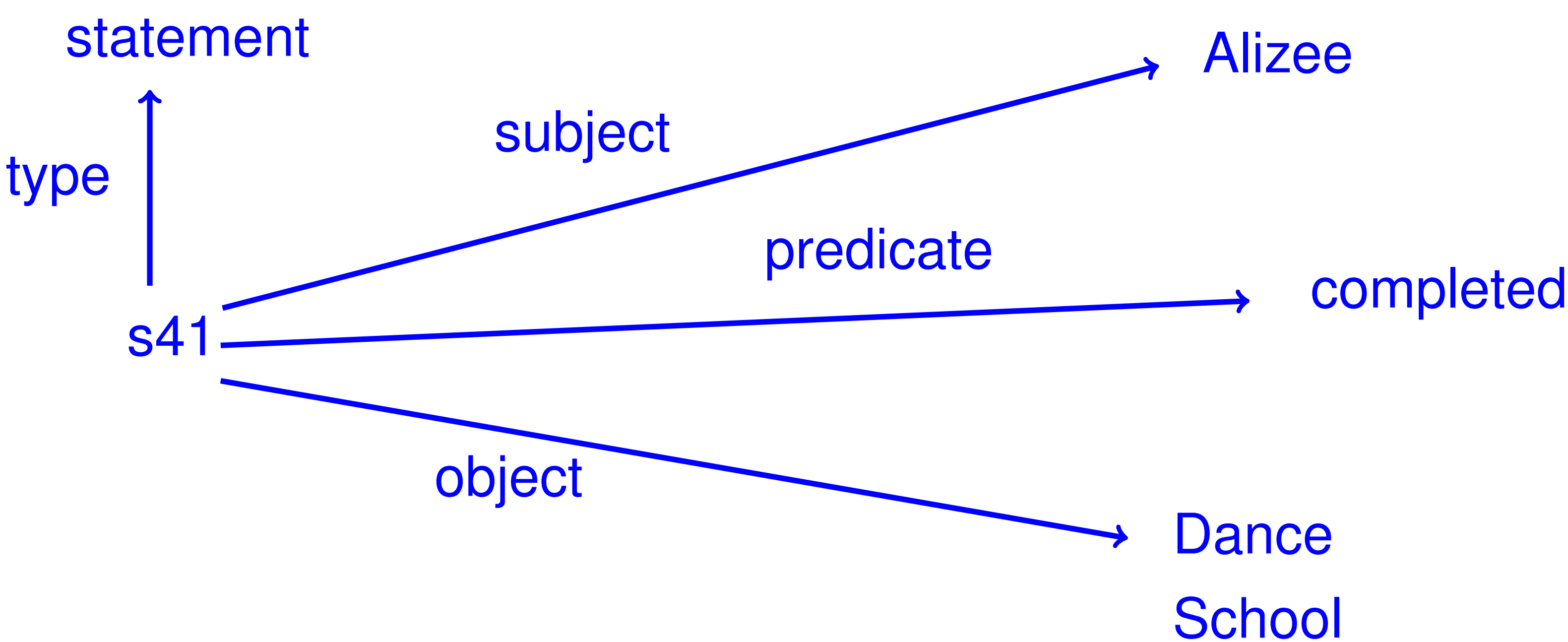
Reification Vocabulary

$statement$ = class of reified statements

$$subject \subseteq statement \times entity$$

$$predicate \subseteq statement \times relation$$

$$object \subseteq statement \times entity$$



Example: Reification

$hopes(Pierre, s42)$

$subject(s42, Alizee)$

$predicate(s42, type)$

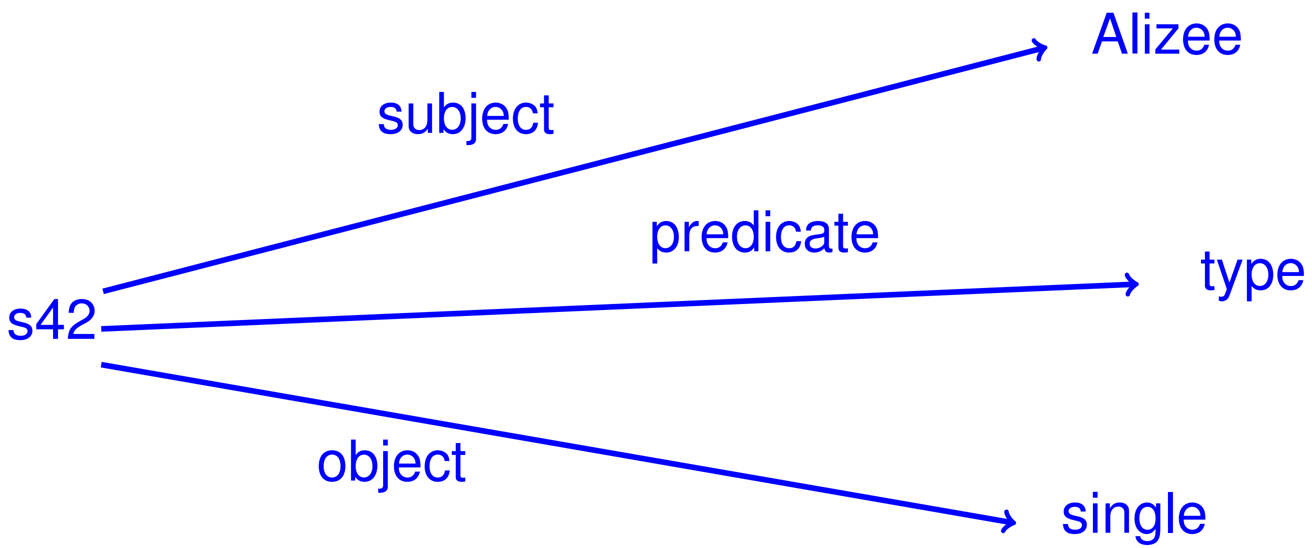
$object(s42, single)$

Simplified notation:

$$hopes(Pierre, type(Alizee, single))$$

Reification and Event Entities

The difference to event entities is that reified statements can be hypothetical (= not part of the KB).



Task>63

Canonicity>73

Digression>77

Task: Reification

Write down a knowledge base

with some reified facts.

Can you reify facts that have reified arguments?

Def: Canonic Entities

An entity is canonic in a KB, if there is no other entity in the KB that represents the same real-world object.

Alizee	produced	Gourmandises
A. Jacotey	produced	Psychédélices
...	...	...

not canonical

(Here, we distinguish exceptionally between an entity in the KB, and the real world object. This distinction is correct, but rarely necessary otherwise, which is why this lecture does not make the difference)

Def: Canonic Relations

An relation is canonic in a KB, if there is no other relation in the KB that represents the same real-world relation.

Alizee	produced	Gourmandises
Alizee	hasProduced	Psychédélices
...	...	...

not canonical

(Here, we distinguish exceptionally between a relation in the KB, and the real world relation. This distinction is correct, but rarely necessary otherwise, which is why this lecture does not make the difference)

Use of Canonicity

Canonicity is essential for

- counting
- answering queries
- constraint satisfaction

Alizee	produced	Gourmandises
Alizee	hasProduced	Psychédélices
...	...	...

not canonical

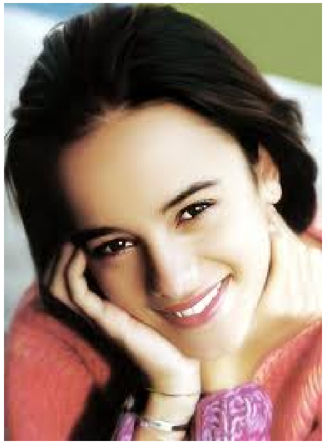
Canonicity and Names

A canonic entity can have multiple names.

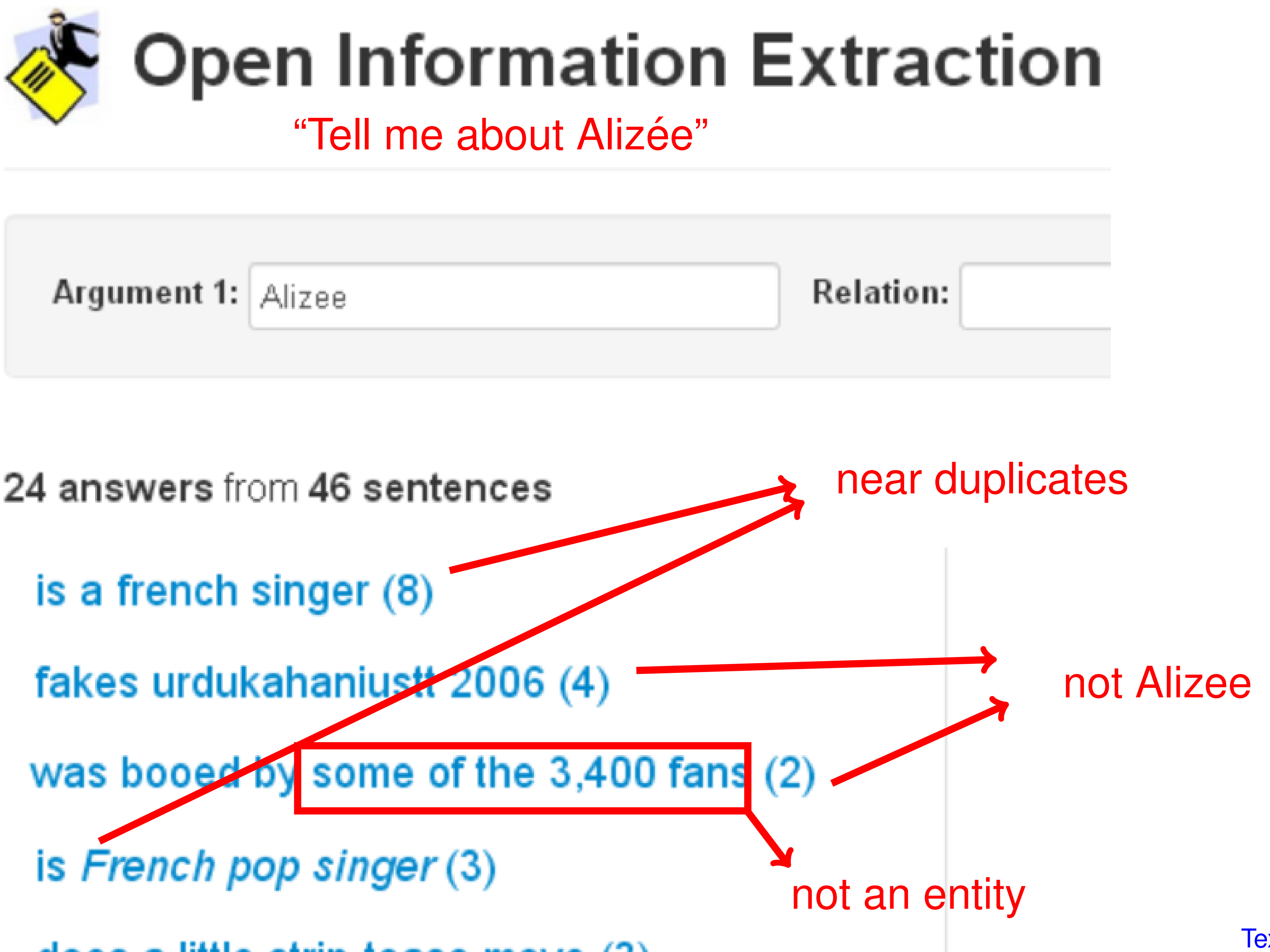
Alizee	produced	Gourmandises
Alizee	produced	Psychédélices
Alizee	label	”Alizee”
Alizee	label	”A. Jacotey”
produced	label	”produced”
produced	label	”has produced”
...	...	...

Canonicity is not easy

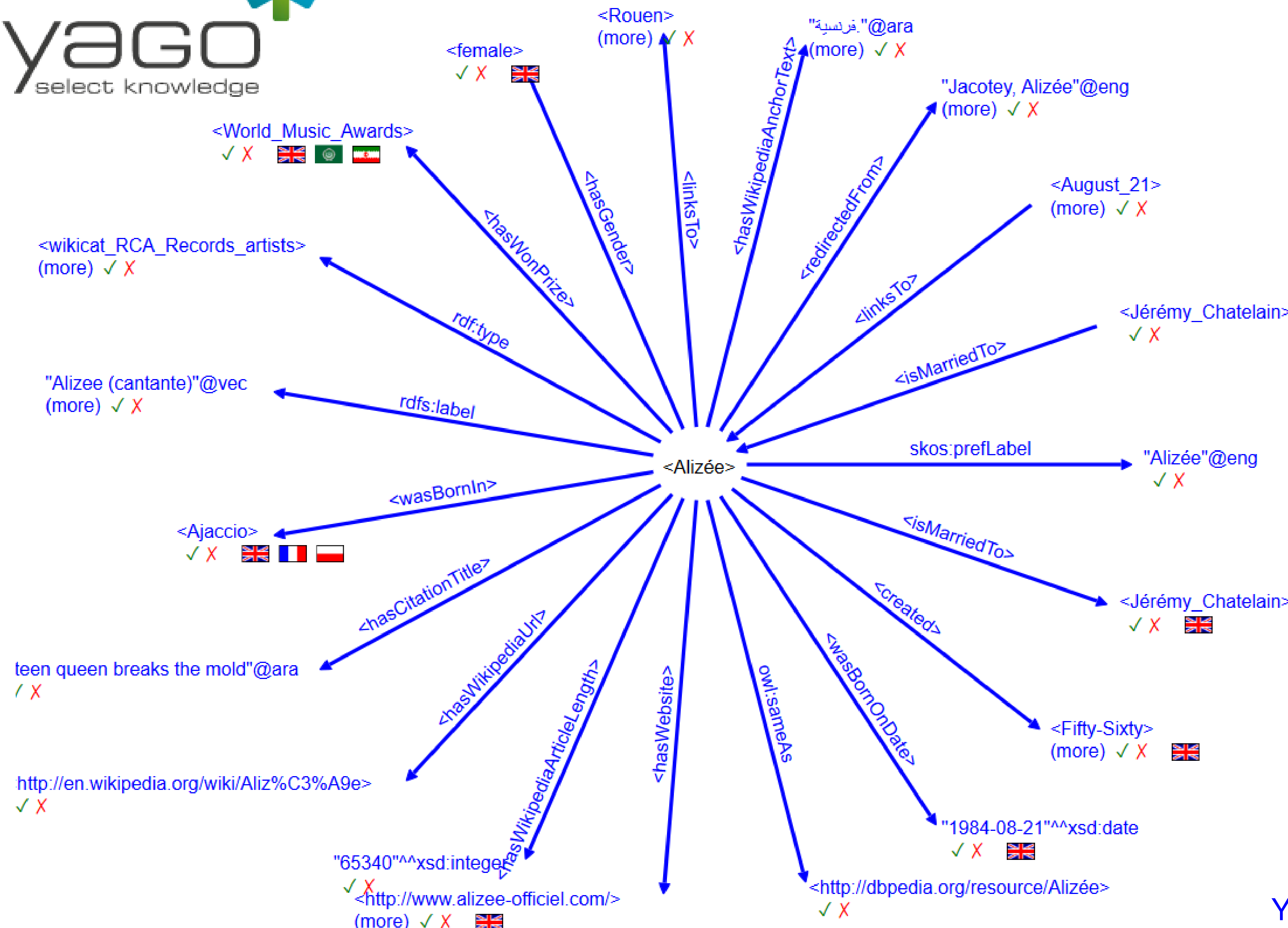
Jacotey is considered one of the “100 Se
women of the world”. The singer said in
that Alizée is married, but she lives sepa


Example: Non-Canonicity



Example: Canonicity



Example: Non-Canonicity

Open Information Extraction

"Who built the pyramids?"

Argument 1: **Relation:** built

192 answers from **865 sentences**

all | person (29) | deceased person (16) | location (13) | monarch (12)

Egyptians (278) → correct

Pharaoh (41) → not bad

correct, not bad → useful

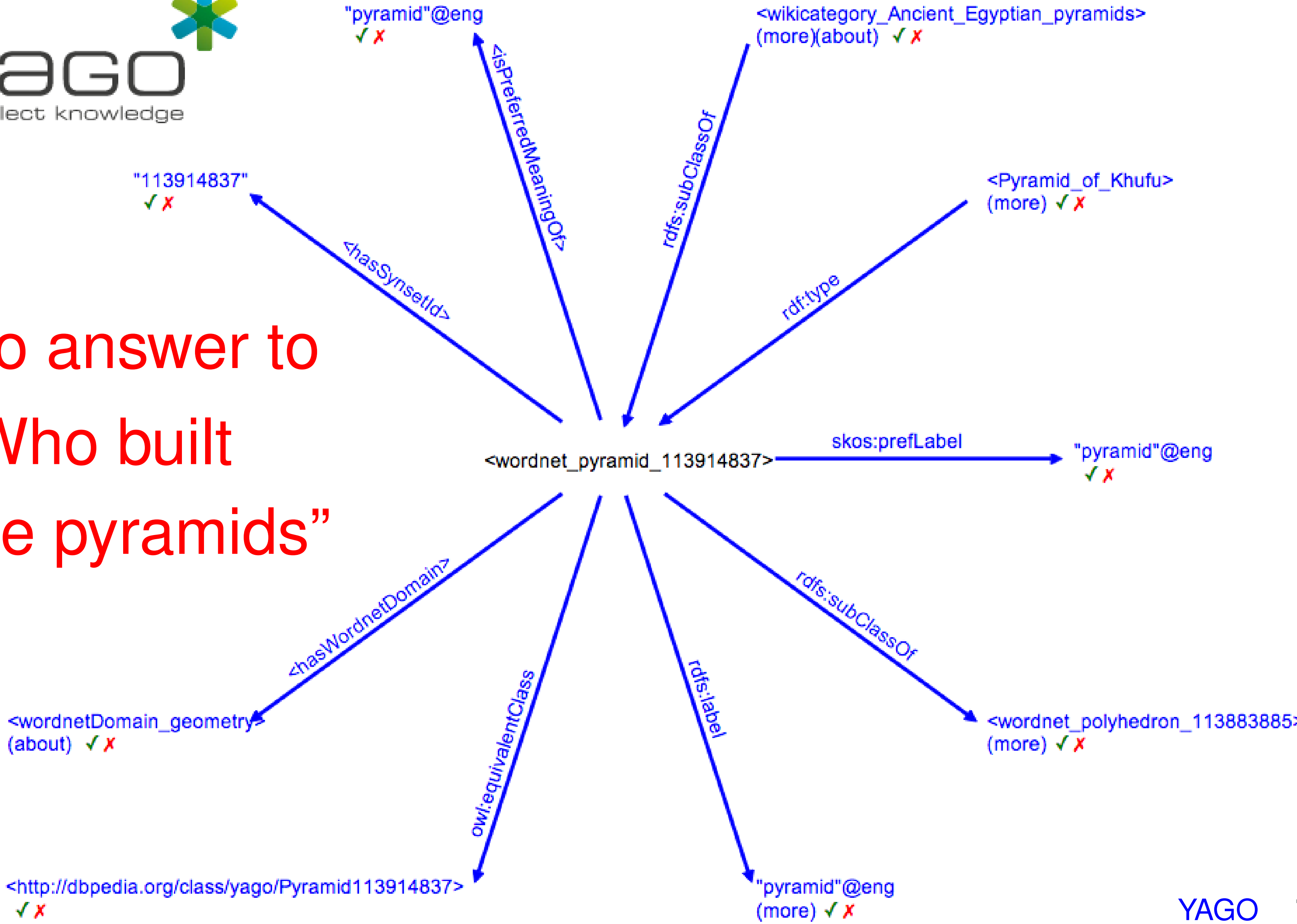
Aliens (35) → less likely

the Ancient Egyptians (31) → duplicate

people (25)

TextRunner

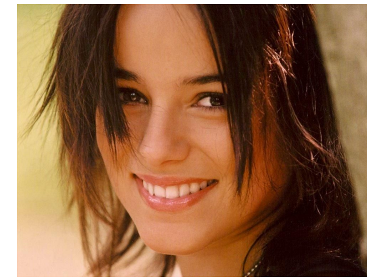
Example: Canonicity



No answer to "Who built the pyramids"

Canonicity as Trade-Off

non-canonic



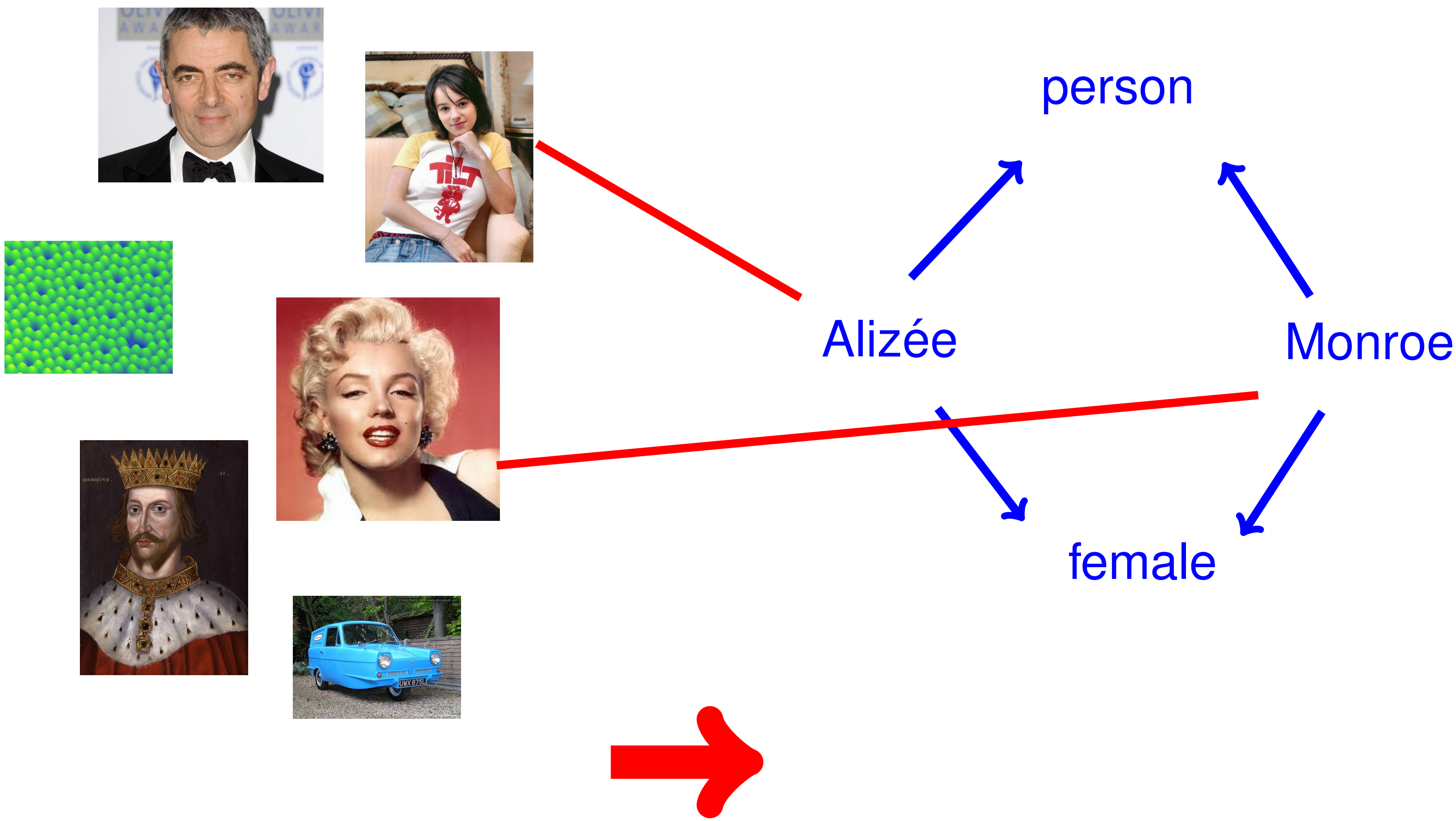
canonic

- easier to extract
- less easy to use
- more noise
- more data

- difficult to extract
- easy to use
- less noise
- less data

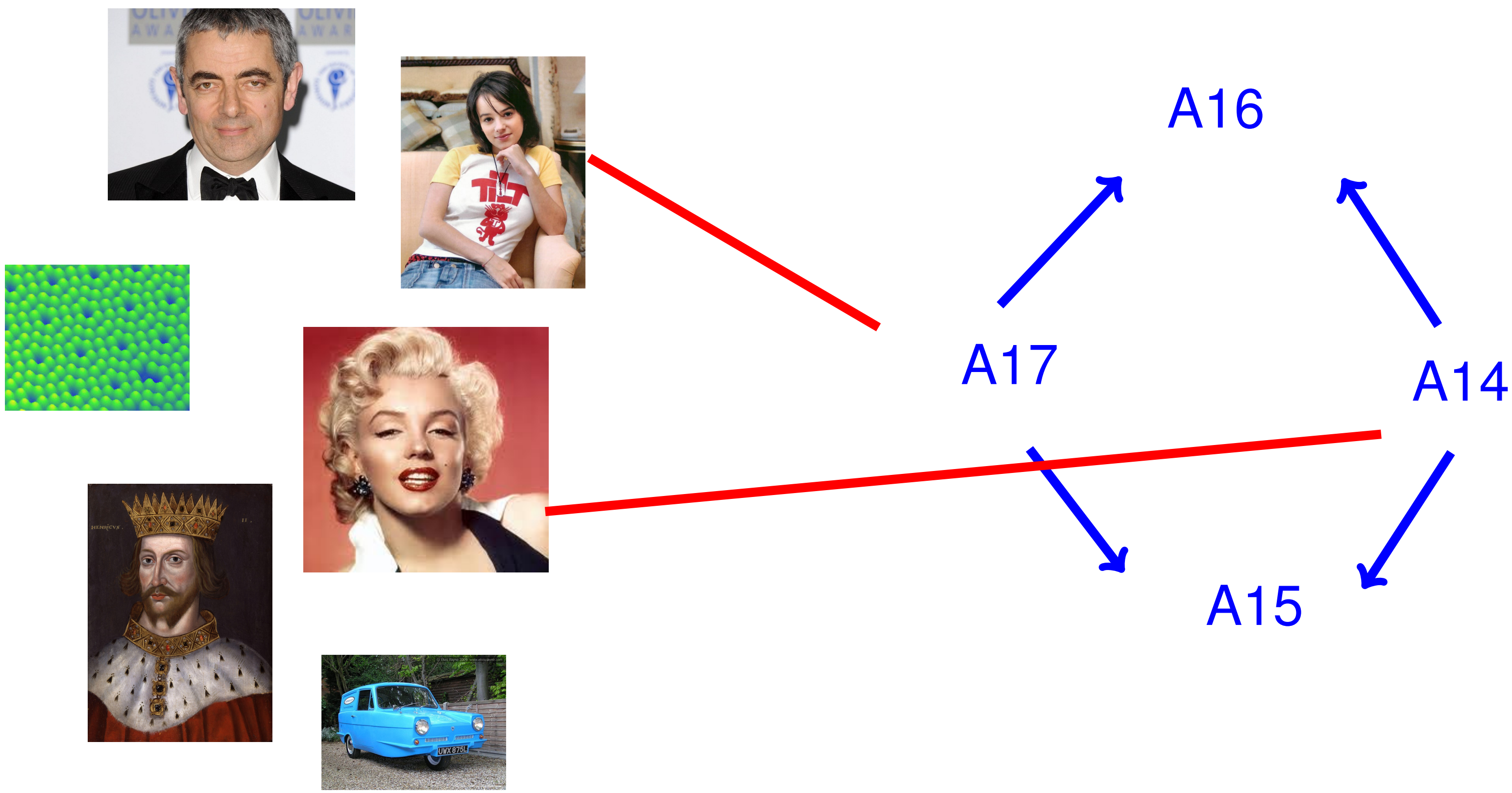
Digression: Reality

We model reality by a representation.



Digression: Reality

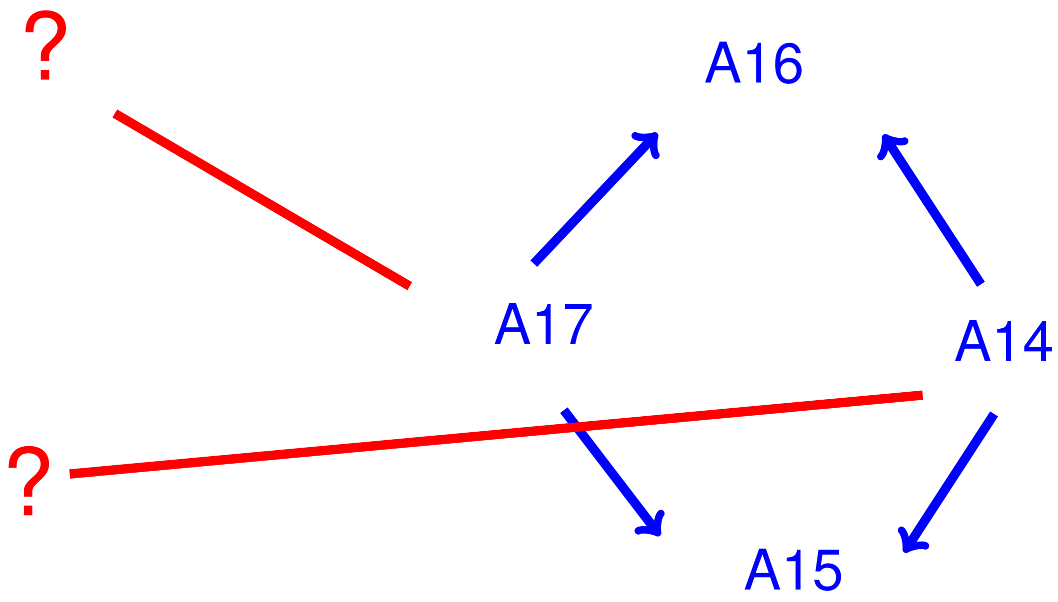
Our identifiers are arbitrary names.



Digression: Reality

Can we reconstruct reality

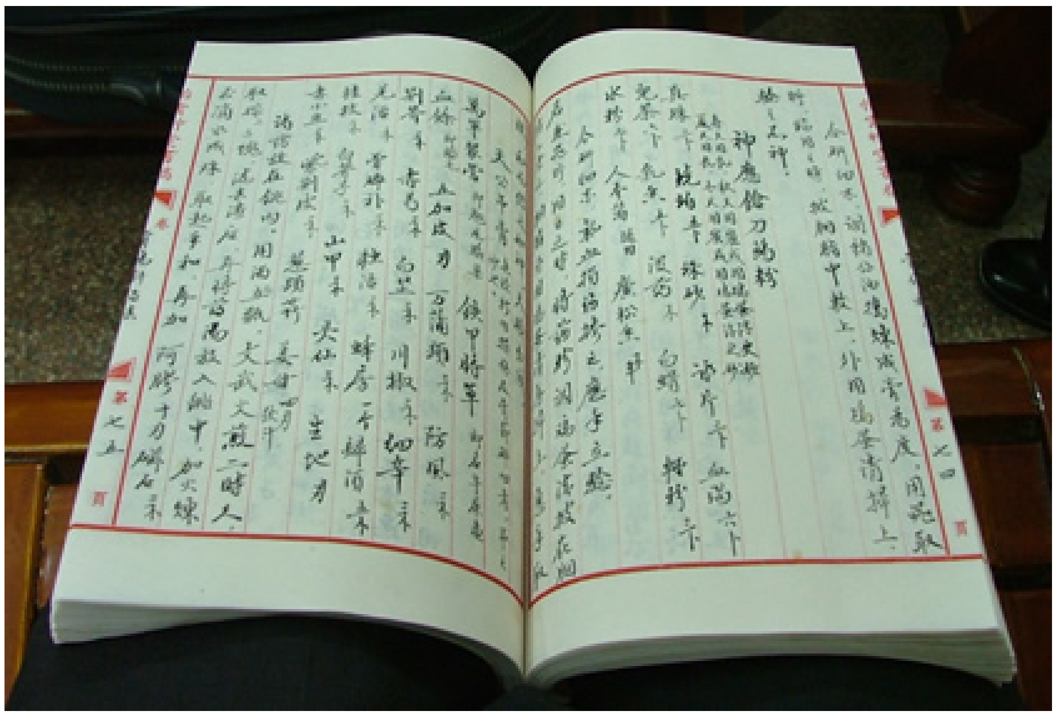
from our model?



Digression: Reality

Most likely no: A Chinese dictionary

is a model of the world...



...yet, by reading it, you cannot learn Chinese.

->corpus

->character-encodings

->named-entity-recognition

Sponsored Link: Dancing Class

Join the free dancing class at Télécom ParisTech

Mondays 19:30-21:30, in English

https://suchanek.name/dancing

References

The Knowledge Representation model
we saw today is RDF without URIs:

RDF Primer

->corpus

->character-encodings

->named-entity-recognition