Knowledge Representation

Nada Mimouni

based on slides by: Fabian M. Suchanek

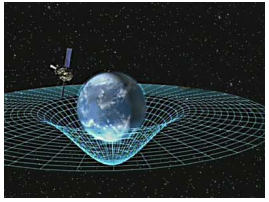
Overview

- Entities & Classes
- Relations
- Binary Relations
- Schema
- Knowledge Graphs
- Reification
- Canonic Entities
- Reality

Entity

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

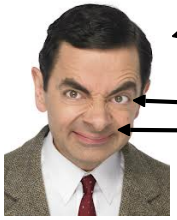



>digression

(The exact definition of "entity" is a philosophical conundrum. Don't worry about it.)

Digression: Entities

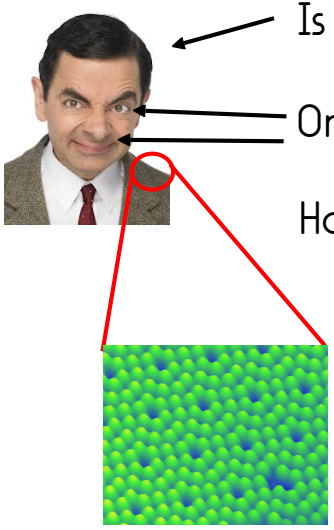


Is this an entity?

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.

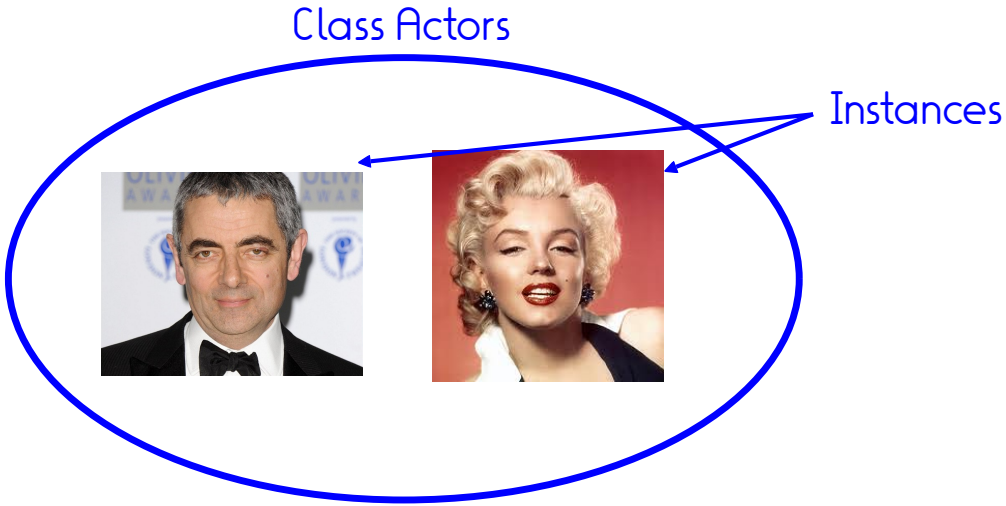
New York Times

Class

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

Entities that are not classes (and not literals, relations, ids) are called instances (or common entities).

Classes:
- Actors
- Cars
- Cities
- Rivers
- Universities
- Theories
- ...

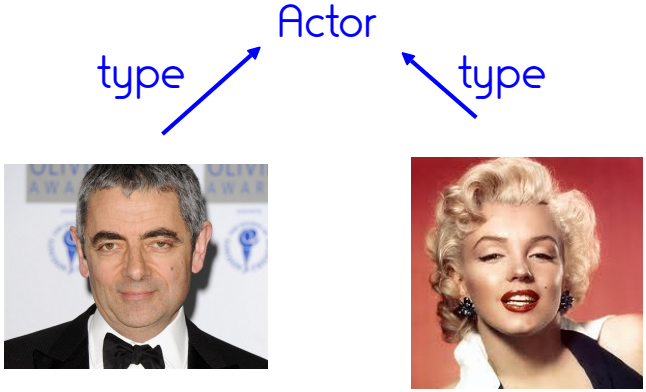


>instances
>instances,subclasses

(The exact definition of "class" is a philosophical conundrum, too. See later in this lecture.)

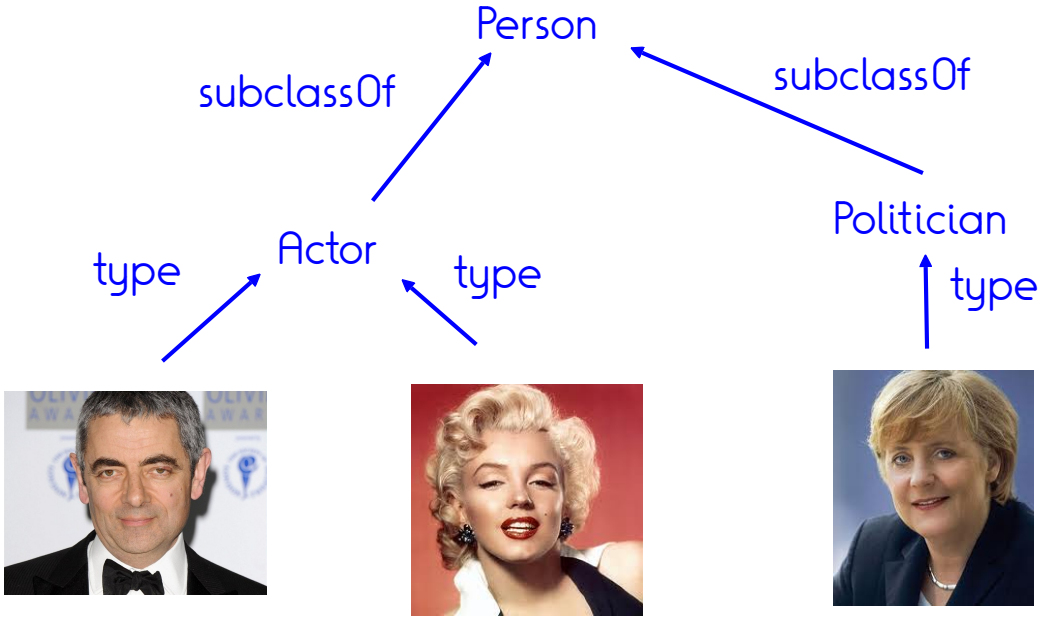
Def: Instance of a class

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



>examples

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 instance of some class
• "X is a Y"	X is an instance of Y
• "Every X is a Y"	X is a subclass of Y

>examples

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 instance of some class
• "X is a Y"	X is an instance of Y
• "Every X is a Y"	X is a subclass of Y

Try it out: city, Elvis, Coli bacteria, Ford, time

>examples

Examples

iPhone -> smartphone
finger -> hand
apple -> orange
flower -> plant
Paris -> city
fruit -> food
France -> Europe
Paris -> France
city -> country

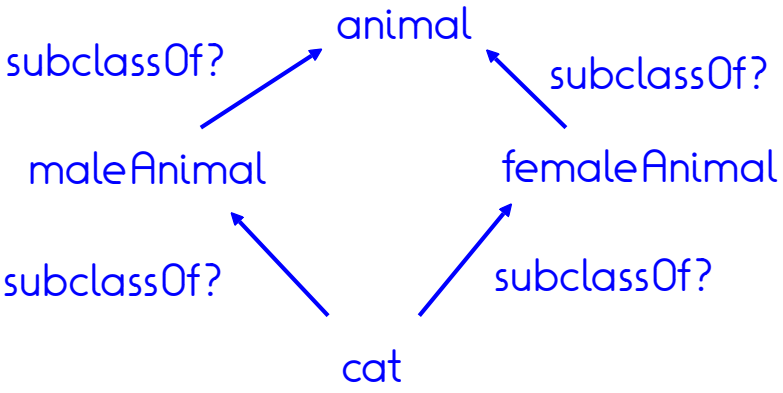
Examples

iPhone -> smartphone	subclass
finger -> hand	partof
apple -> orange	sibling
flower -> plant	subclass
Paris -> city	type
fruit -> food	subclass
France -> Europe	locatedIn
Paris -> France	locatedIn
city -> country	domain and range of "locatedIn"

Examples

Consider a taxonomy of the animal kingdom.

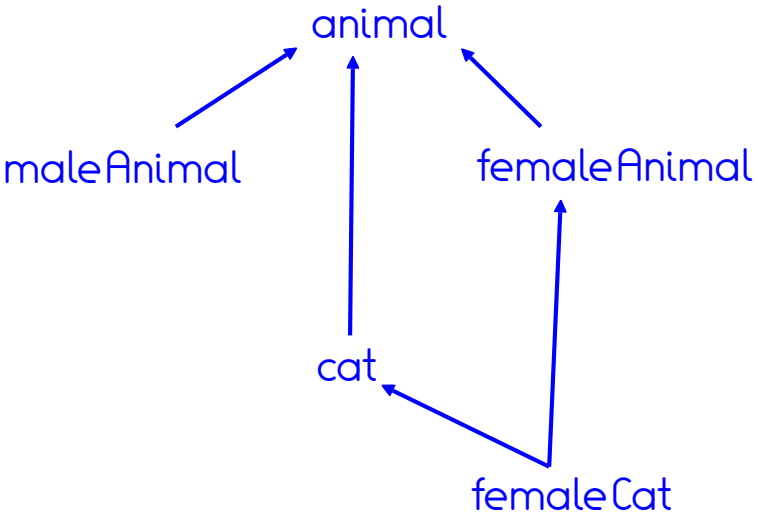
How do we deal with "male" and "female"?



Examples

Consider a taxonomy of the animal kingdom.

How do we 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 (tough)

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.

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

$$D(born) = \{ <x,y,z> \mid x\ was\ born\ in\ y\ in\ year\ z \}$$

Digression: Semantics

But these are again symbols.

$$D(born) = \{ <x,y,z> \mid x \; was \; born \; in \; y \; in \; year \; z \}$$

What is their semantics?

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:

$$bornInCity(Atkinson, Consett)$$

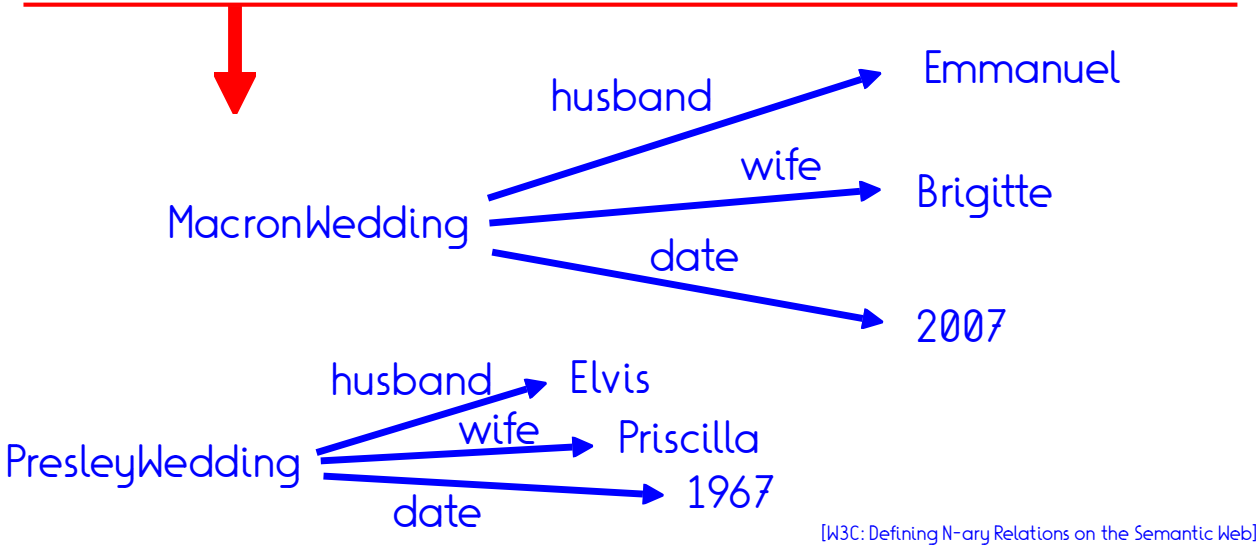
The first argument of a fact is the subject, the second the object.

n-ary facts as binary facts

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

WEDDINGS

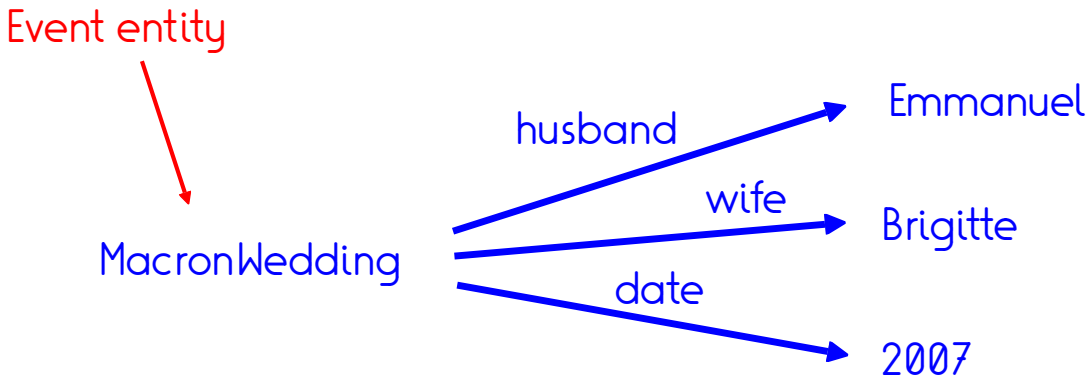
Husband	Wife	WeddingDate
Emmanuel	Brigitte	2007
Elvis	Priscilla	1967
...	...	...



[W3C: Defining N-ary Relations on the Semantic Web]

Def: Event Entity

An event entity represents an n-ary fact.



>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.

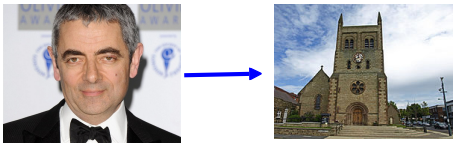
(There may be multiple solutions)

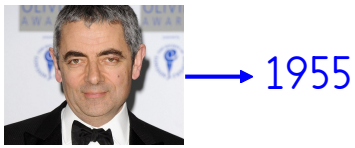
Binary relations are flexible

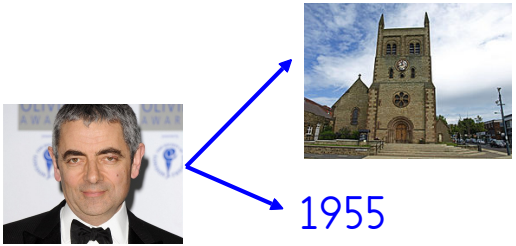
n-ary relations enforce the presence of all arguments:

born	Person	City	Year
	Atkinson	Consett	1955

Binary relations don't:





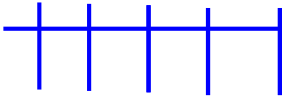


Binary vs n-ary

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

binary

- more relations
- less arity
- more flexibility



n-ary

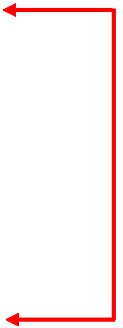
- less relations
- more arity
- more control

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.

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

hasNationality
hasSpouse (in medieval ages)
hasNumberOfTeeth

Def: Inverse Functional Rel.

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

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

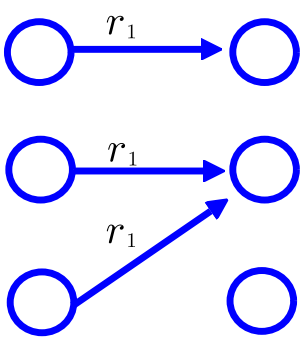
hasCitizen

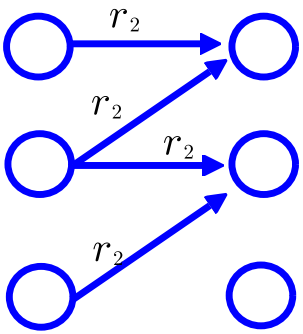
hasSpouse (in medieval ages)

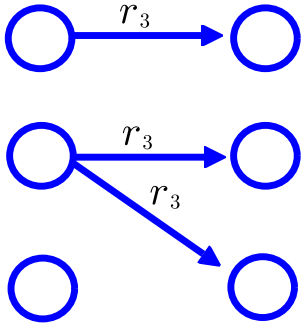
>task

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.

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

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

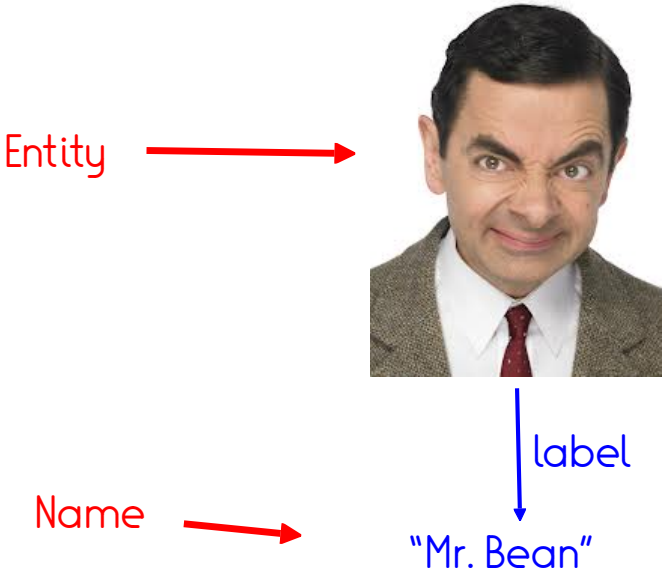
$\Rightarrow MrBean = Bean$

$born(Bean, 1955)$

$born(MrBean, 1955)$ (Nothing follows)

Def: Name

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

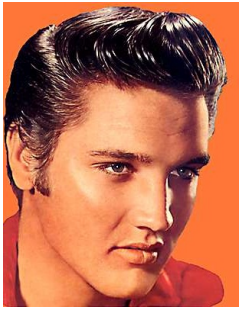


>ambiguity

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")



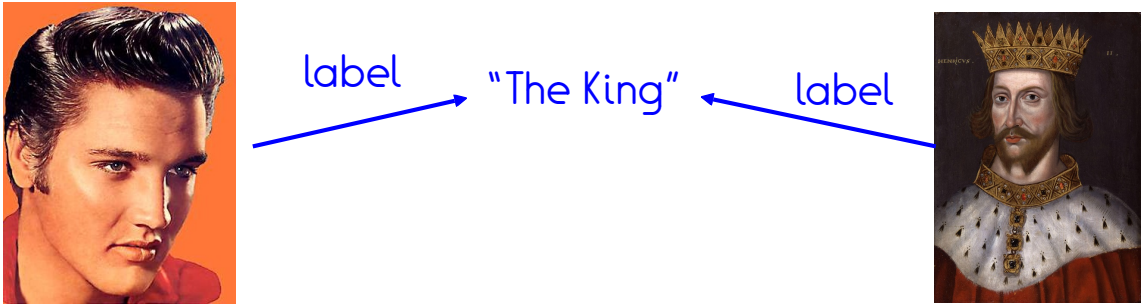
label → "The King"

label → "Elvis"

Def: Ambiguity

If a name is attached to multiple entities,
the name is called ambiguous.

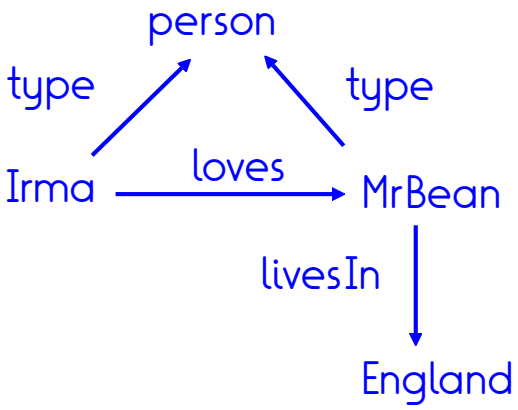
(The adjective for the names is "ambiguous", the phenomenon is called "ambiguity")



Def: Knowledge Graph

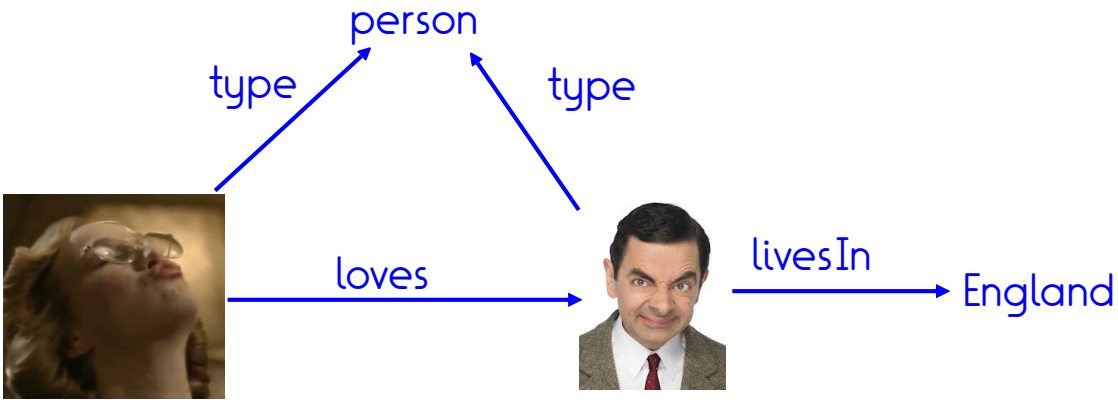
A knowledge graph (also: Entity-Relationship graph, Knowledge base, KB) is a directed labeled multi-graph that has an edge x->y with label r, iff r(x,y).

$loves(Irma, MrBean)$
$type(Irma, person)$
$type(MrBean, person)$
$livesIn(MrBean, England)$



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).



-> schema

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")

Popular triple stores are:

- BlazeGraph
- Jena
- Virtuoso
- ...or classical databases

->schema

Classes as binary relations

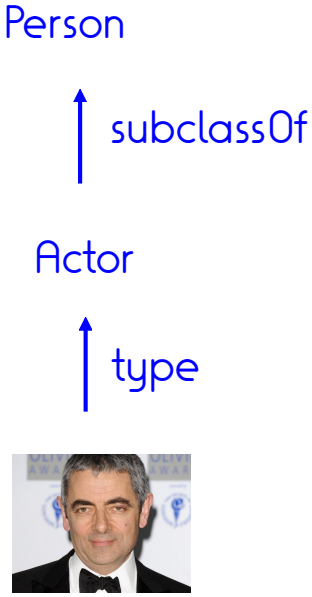
One way to represent a class is by the binary relations $type$, $subclassOf$.

$$type \subseteq entity \times class$$

$$type(Atkinson, actor)$$

$$subclassOf \subseteq class \times class$$

$$subclassOf(actor, person)$$



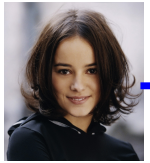
("entity" is the class of entities, "class" is the class of classes. See later in the lecture for details.)

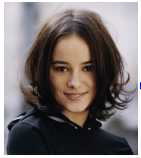
Digression: Classes and Relations

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


type → woman


gender → female


type → singer


job → singer

Domains as binary relations

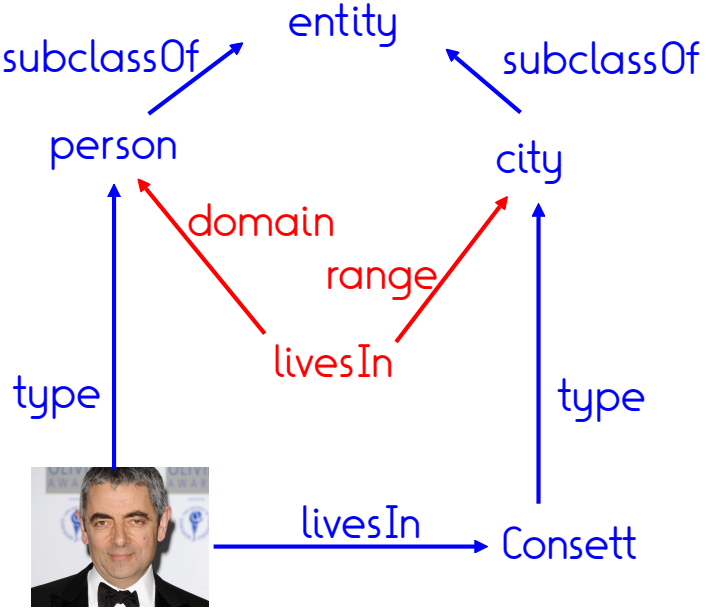
The domain and range of relations can be expressed by binary relations $domain$ and $range$.

$domain \subseteq relation \times class$

$domain(livesIn, person)$

$range \subseteq relation \times class$

$range(livesIn, city)$

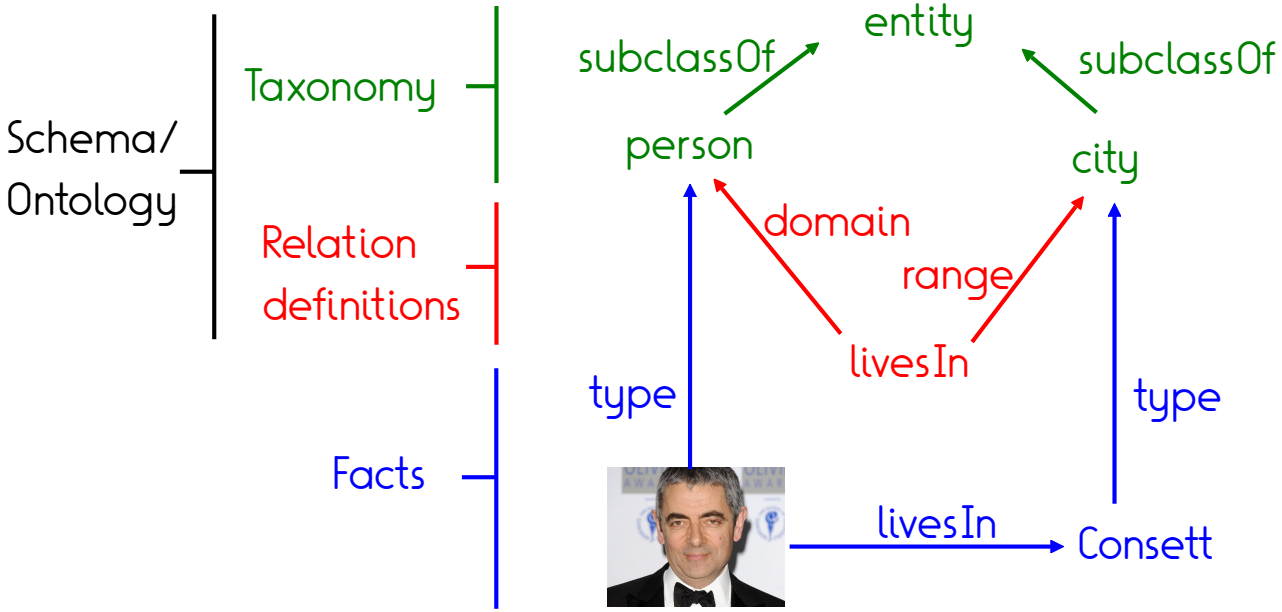


("relation" is the set of relations. See later in this lecture for details.)

Def: Schema

The schema/ontology is the part of a knowledge graph that consists of

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



Task: Schema

Define a schema for the domain of politics.

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

Digression: Task: Class entities

Draw a knowledge graph with the relations $domain$ and $range$.
Can $domain$ and $range$ appear as nodes?

Digression: The problem with classes

type(class, class) "The class "class" is a class"

class ∈ class ...in a naïve set-theoretical interpretation

class = {cars, frenchPeople, class, ...}

class = {cars, frenchPeople, {cars, frenchPeople, { cars, ...

Digression: The problem with classes

We distinguish:

1) the class/concept/ generalization

The class of French people

2) the extension of the class

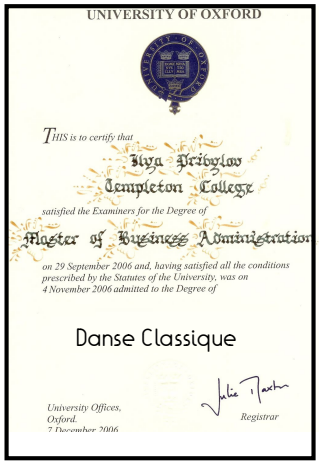
The extension of the class of French people

{Emmanuel Macron, Alizee, Napoleon Bonaparte, Pierre Dupont, ...}

(This means that two classes can have the same extension and still be different. Relations are treated in the same way. See RDF Semantics for a discussion.)

Reified statements

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


UNIVERSITY OF OXFORD

THIS is to certify that

Ilya Pribylov

Templeton College

satisfied the Examiners for the Degree of

Master of Business Administration

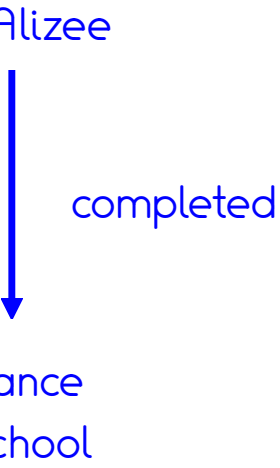
on 29 September 2006 and, having satisfied all the conditions prescribed by the Statutes of the University, was on 4 November 2006 admitted to the Degree of

Danse Classique

University Offices, Oxford. 7 December 2006

Registrar

represents



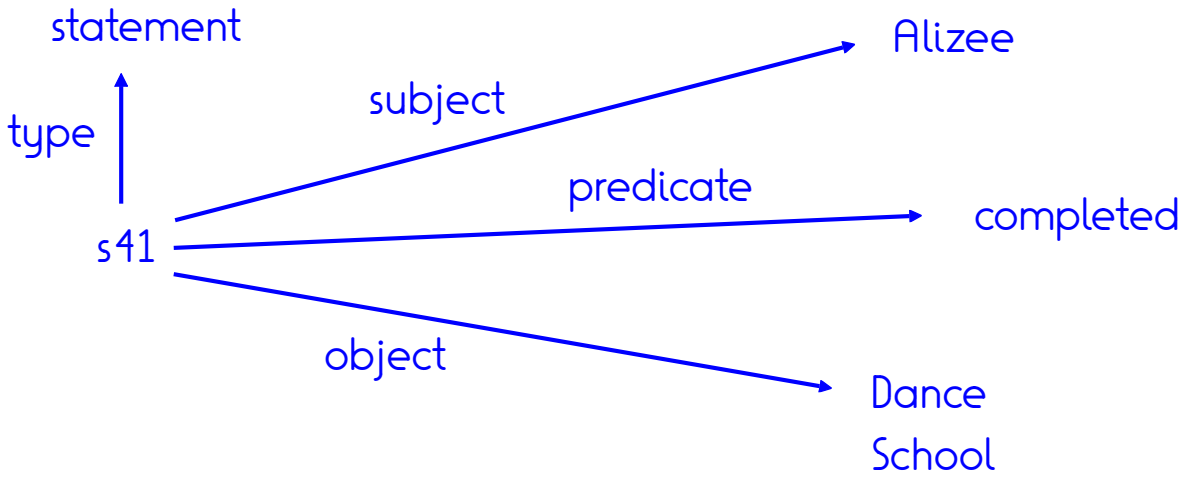
Reification Vocabulary

$statement$ = class of reified statements

$subject \subseteq statement \times entity$

$predicate \subseteq statement \times relation$

$object \subseteq statement \times entity$



Example: Reification

$hopes(Pierre, s42)$

$subject(s42, Alizee)$

$predicate(s42, type)$

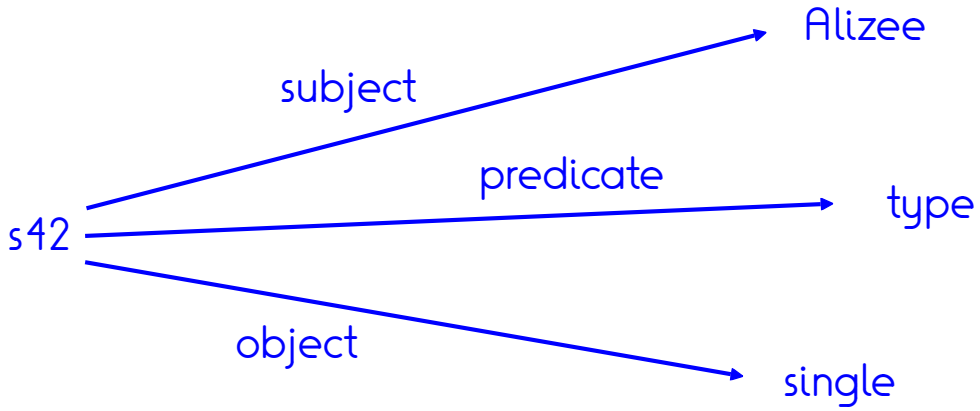
$object(s42, single)$

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: Reification

Write down a knowledge base
with some reified facts.
Can you reify facts that have reified arguments?

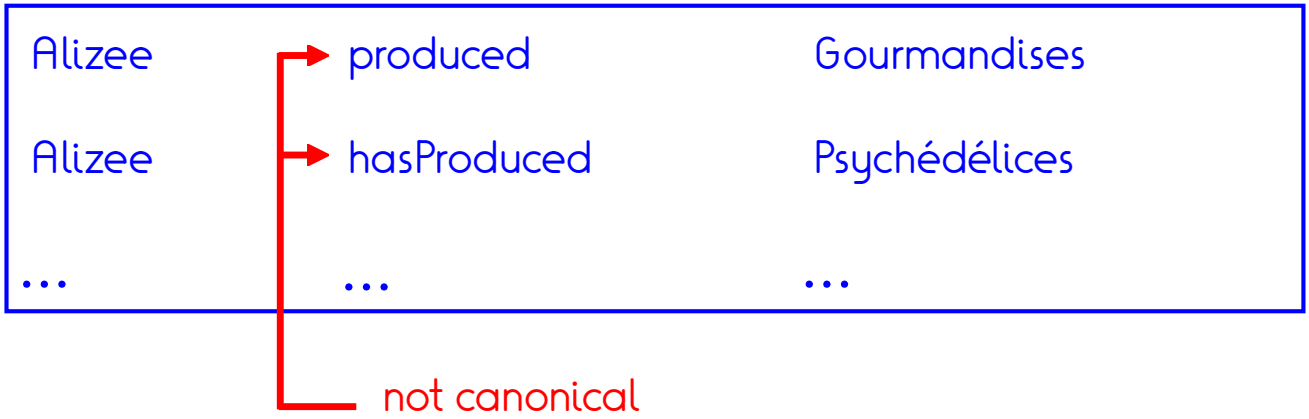
Def: Canonic Entities

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

(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.

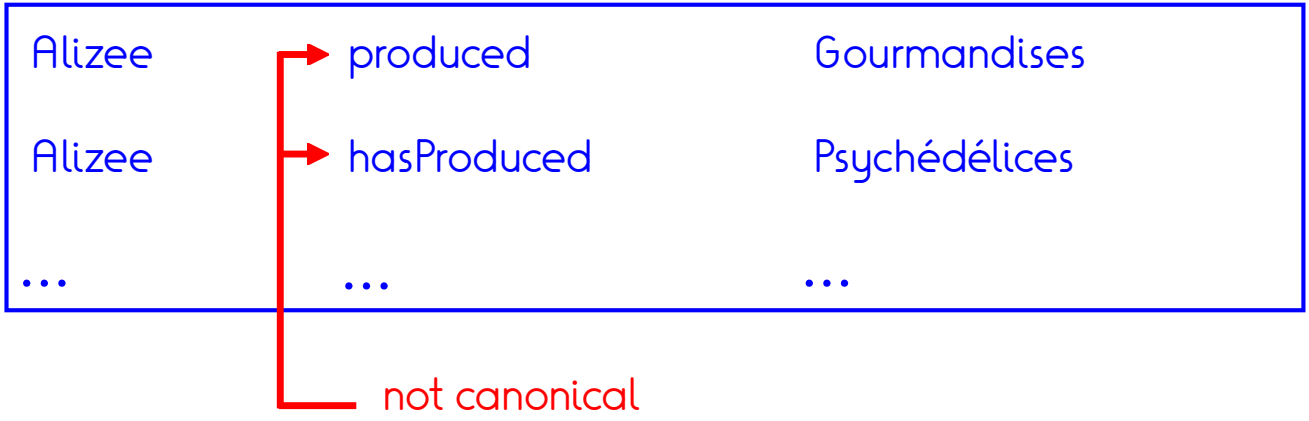


(Here, we distinguish exceptionally between a relation in the KB, and the real world relation. This distinction is correct, but rarely necessary otherwise, which is why this lecture does not make the difference)

Use of Canonicity

Canonicity is essential for

- counting
- answering queries
- constraint satisfaction



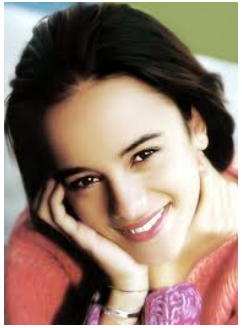
Canonicity and Names

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

Open Information Extraction

"Tell me about Alizée"

Argument 1: Alizee Relation:

24 answers from **46 sentences**

near duplicates

is a french singer (8)

fakes urdukahaniustt 2006 (4)

not Alizee

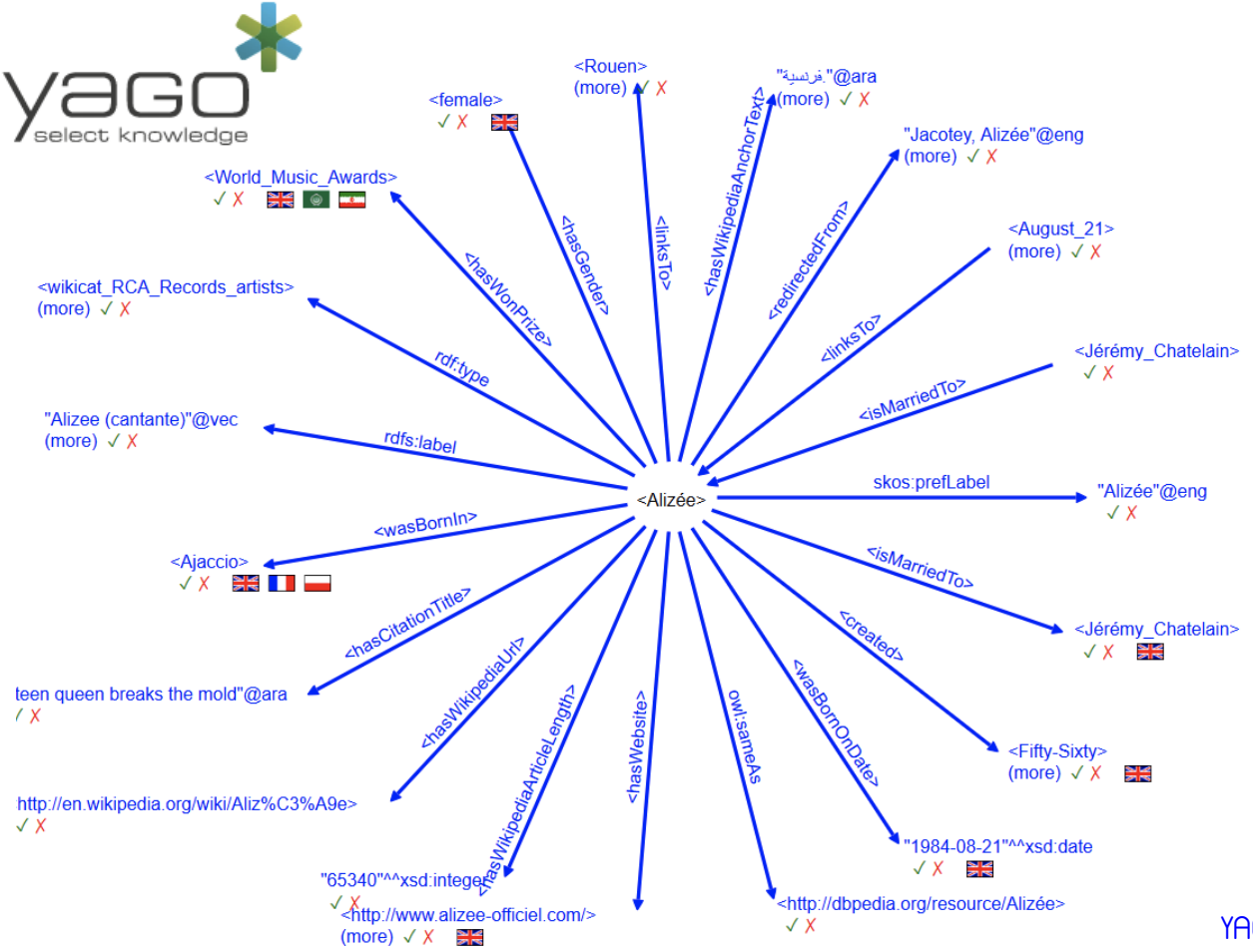
was booed by some of the 3,400 fans (2)

is *French pop singer* (3)

not an entity

does a little strip-tease move (3)

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

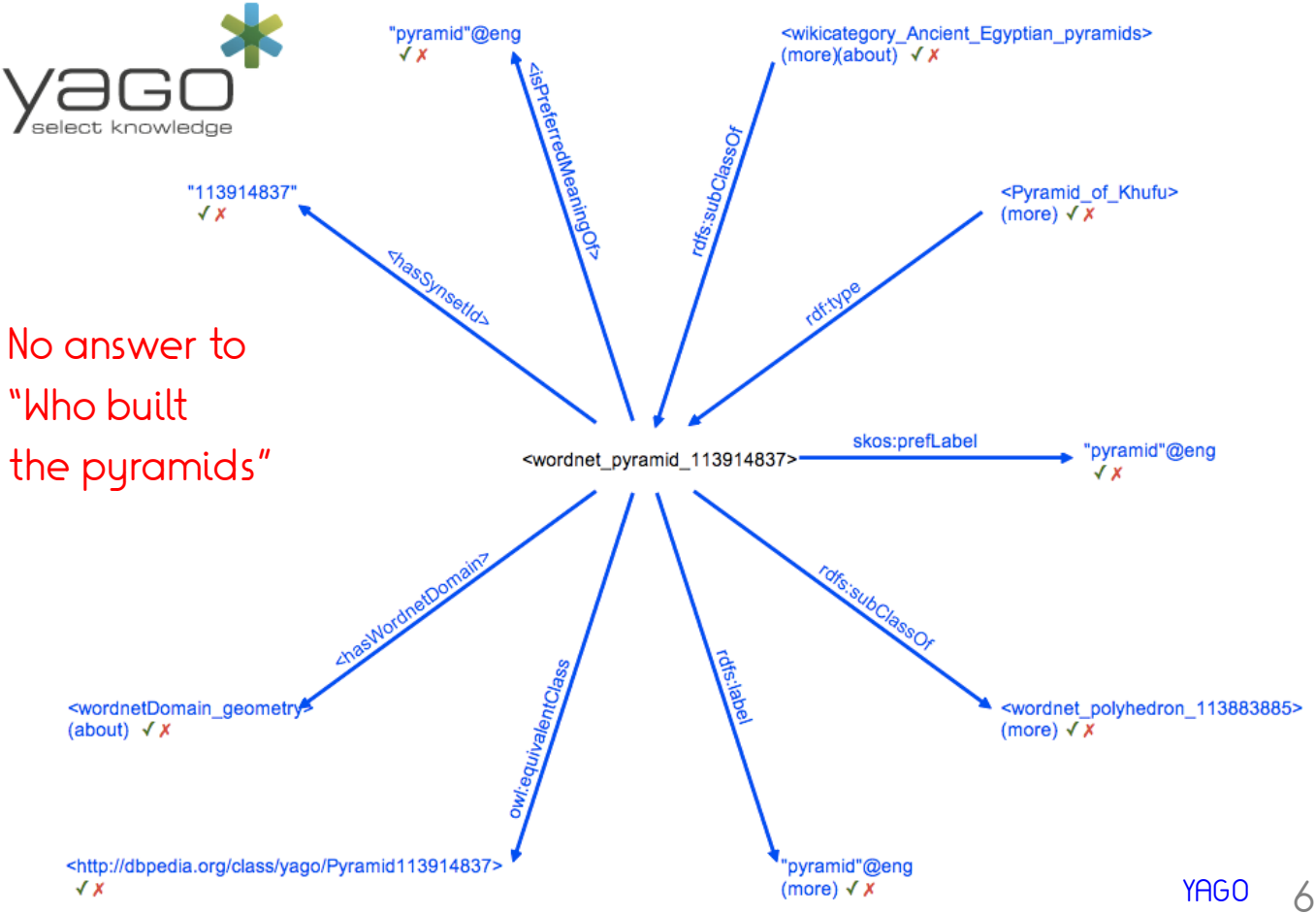
Aliens (35) → less likely

the Ancient Egyptians (31) → duplicate

people (25)

correct, not bad → useful

Example: Canonicity



Canonicity as Trade-Off

non-canonic

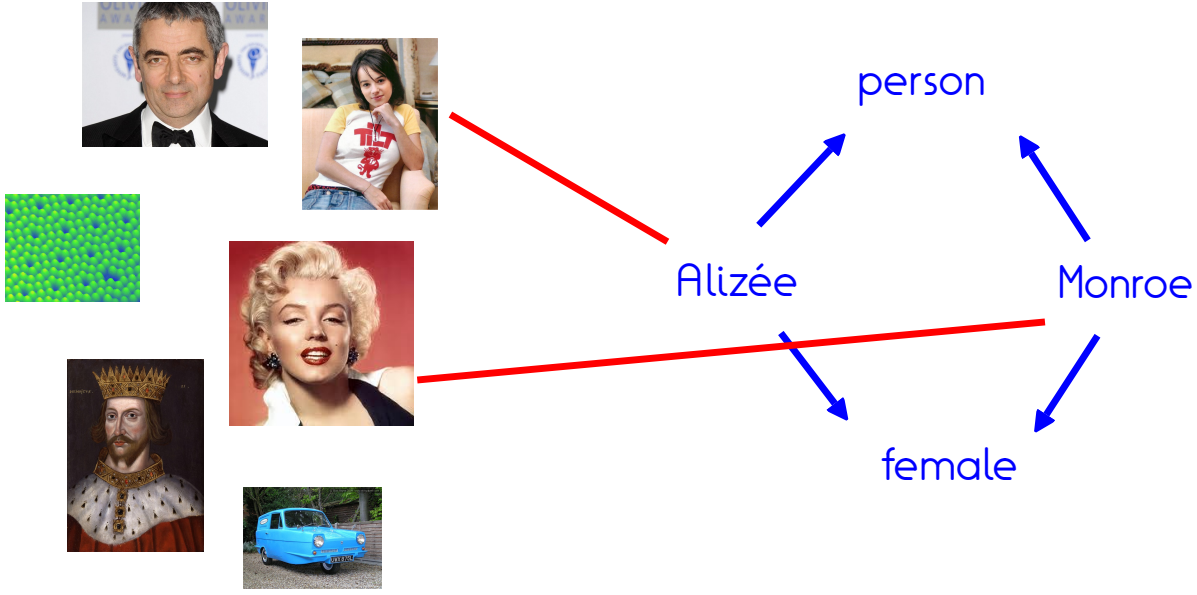
canonic

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

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

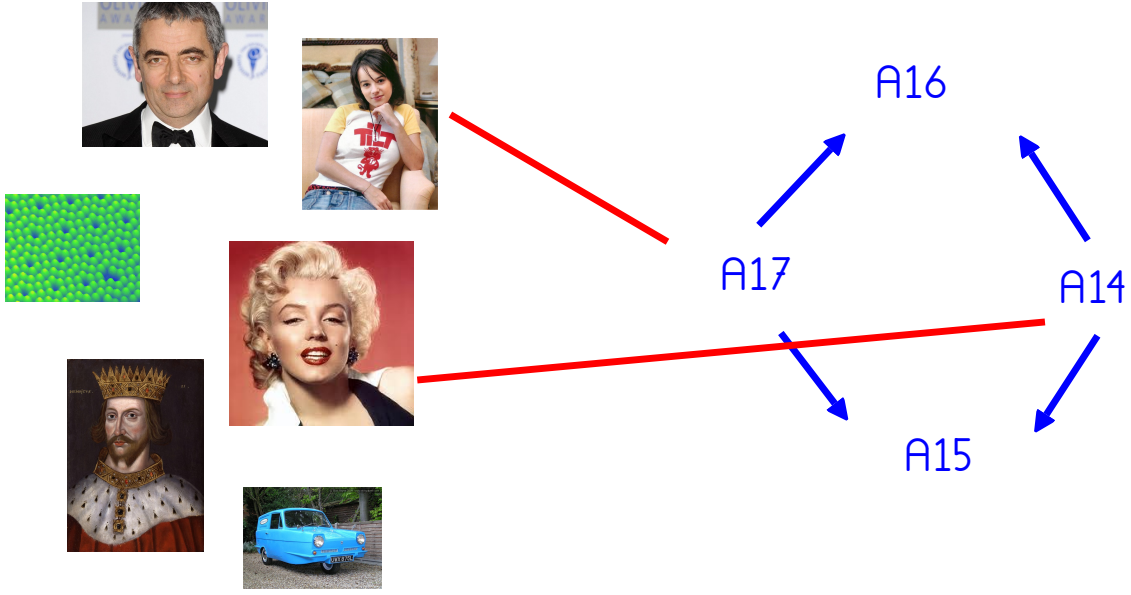
Digression: Reality

We model reality by a representation.



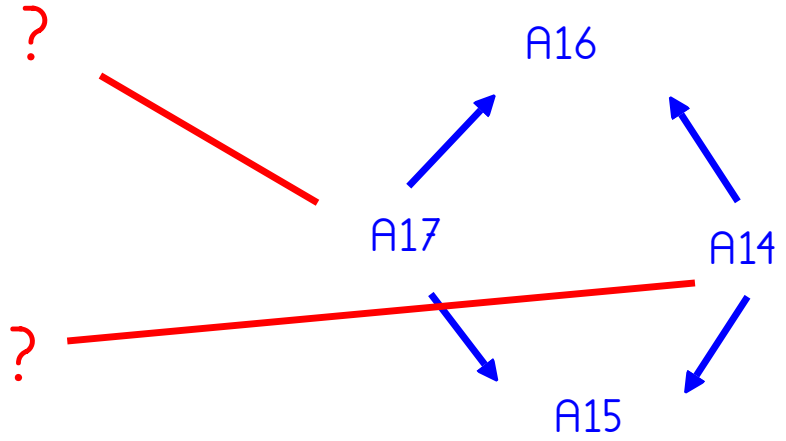
Digression: Reality

Our identifiers are arbitrary names.



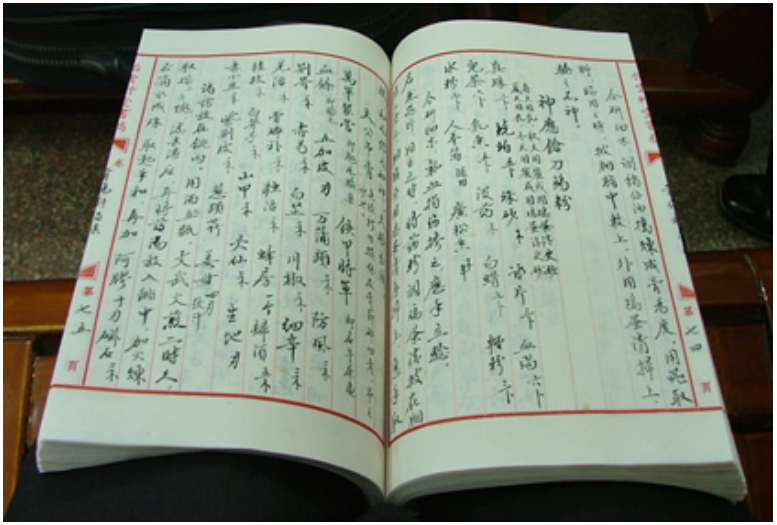
Digression: Reality

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.

->introduction-nlp
->corpus ->GAFA
->character-encodings
->named-entity-recognition
->semantic-web

References

The knowledge representation model
we saw today is RDF without URIs.

See also:
Knowledge Representation in Entity-Centric Knowledge Bases

->introduction-nlp
->corpus ->GAFA
->character-encodings
->named-entity-recognition
->semantic-web