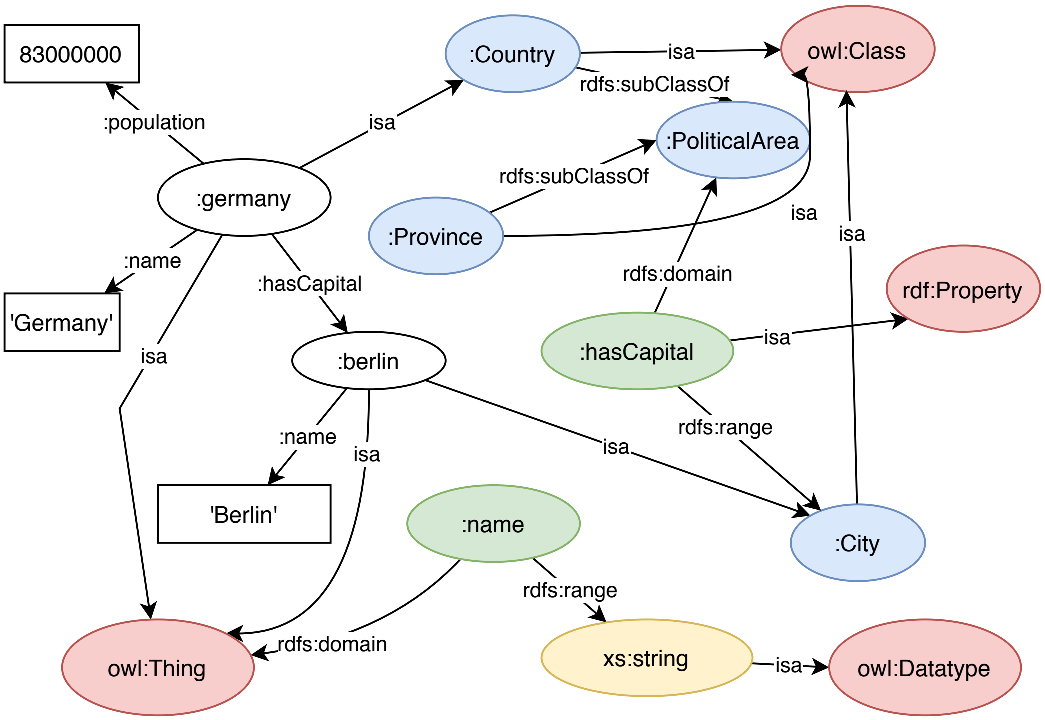
83000000
:Country
owl:Class
isa
rdfs:subClassOf
:population
isa
:PoliticalArea
rdfs:subClassOf
:germany
:Province
isa
isa
:name
:hasCapital
rdfs:domain
rdf:Property
'Germany'
isa
:berlin
:hasCapital
isa
:name
isa
rdfs:range
isa
'Berlin'
:name
:City
rdfs:range
rdfs:domain
owl:Thing
xs:string
isa
owl:Datatype

Yesterday: "Symbolic Reasoning"

Background: Philosophical logics, mathematical logics, model theory aspects:
human reasoning *about* properties of the logic).

Each logic, and thus also First-Order Logic provides a framework
that can be used for symbolic reasoning:

FOL Formulas are strings, FOL reasoning are algorithms that
work on their parse trees.

=> symbolic reasoning: all about Syntax, not Semantics

Formulas are evaluated wrt. first-order-logic
structures/interpretations

Syntax: the symbols used for writing formulas:

* logical symbols: $\wedge$, $\exists$, ...
* variables: x, y, ...
* depending on the application: predicate symbols and function symbols, "signature" Σ

for mondial: Σ = {Country, City, name, hasCapital, ...}

FOL Structure: $\mathcal{S} = (I, \mathcal{D})$

$\mathcal{D}$ is the domain ... the things in the real world.

I maps the symbols from Σ to the domain ...

Example: our "real-world-application" contains a (green) frog, and strings and numbers:

$\mathcal{D}$ = { 🐸 } $\cup$ Strings $\cup$ Numbers ...

Signature to talk about the frog and its properties: (1-ary and 2-ary predicates and constant symbols)

$\Sigma = \{$ Frog/1, Green/1, name/2, bob/c0 $\}$

Interpret the symbols in OUR structure/model $\mathcal{S}$ (=current situation):

I (bob) = (🐸) (an element from $\mathcal{D}$)

I (name) = { (🐸 ,"Bob"), ... } (a set of 2-tuples over $\mathcal{D}$)

I (Frog) = {(🐸), ... } (a set of 1-tuples over $\mathcal{D}$)

Knowledge base $\mathcal{K}$: all frogs are green.

$\forall x$: Frog(x) $\rightarrow$ Green(x)

Our $\mathcal{S}$ must be a model of $\mathcal{K}$:

Tableau calculus: what can we derive?

$\forall x$: Frog(x) $\rightarrow$ Green(x)

Frog(bob)

Frog(X1) $\rightarrow$ Green(X1) (introduce a tableau variable X1)

$\neg$ Frog (X1) $\vee$ Green(X1) equivalent

open two branches

$\neg$Frog(X1)	Green(X1)
$\square$ X1 $\leftarrow$ bob	➤ Green(bob)

=> conclusion by reasoning: bob must be green in our $\mathcal{S}$

=> I(Green) $\supseteq$ I(bob)

I(Green) $\supseteq$ { (🐸)}

I practically is a database, containing unary and binary tables:

(note: DB is only on the syntax level, so bob <-> 🐸)

name	
bob	"Bob"
:	:

Frog
bob

(the constant bob/c0 is like an object identifier)