QUEEN'S UNIVERSITY BELFAST

MSc Artificial Intelligence

Knowledge Engineering

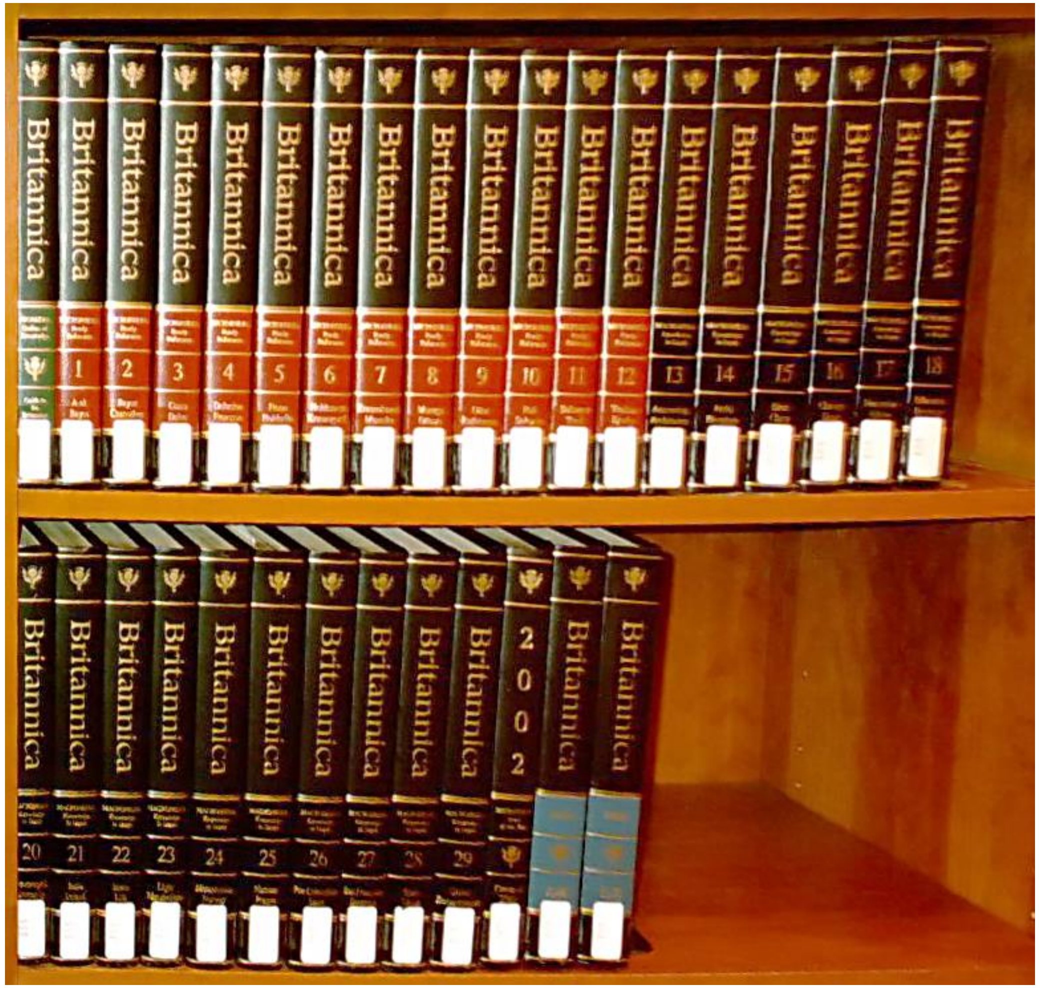
22 November 2023

A look back at knowledge bases

- Knowledge Bases provide one means of representing our knowledge of the world
  - Robust mechanisms for querying
  - Can produce new knowledge (new sentences)
  - Easily updated
  - Rigourous inference
  - Human readable in principle

Limitations of Knowledge Bases

- Not scalable
- Labour intensive
- Relationships not especially transparent
- No “types”
- No annotations



Overcoming these Limitations

There are many things we would like KB to do:

- Explicitly model relationships between entities
- Allow the “atomic units of knowledge” to be annotated with additional information
- Support performant and accurate inference
- Scale to very large numbers of “atoms”.
- Support highly parallel querying
- Interpretable by humans
- Readable by machines

- Enter the **knowledge graph**

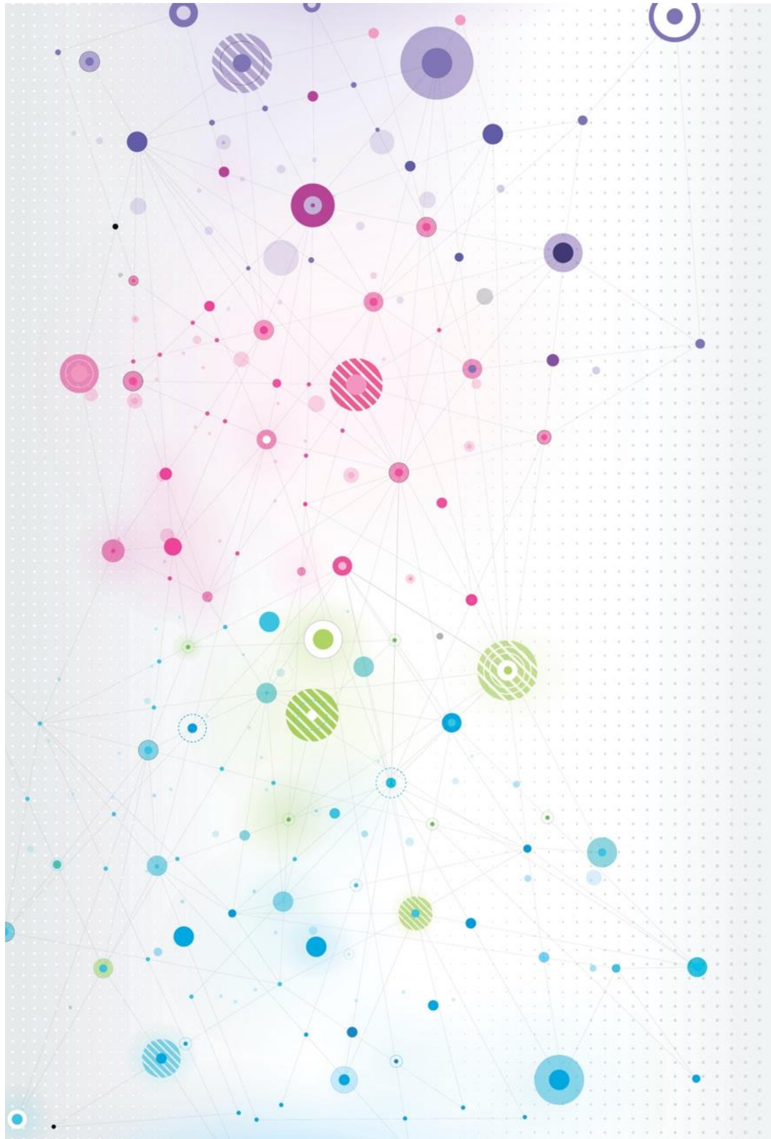
Rule-based systems as graphs

- If x is a dog then x is an animal.
- If x is a cow then x is an animal.
- If x is an animal then x is a living thing.
- If x is a cow then x eats herbs.
- If x is a herb then x is a plant.
- If x is a plant then x is a living thing.

What does the graph miss?

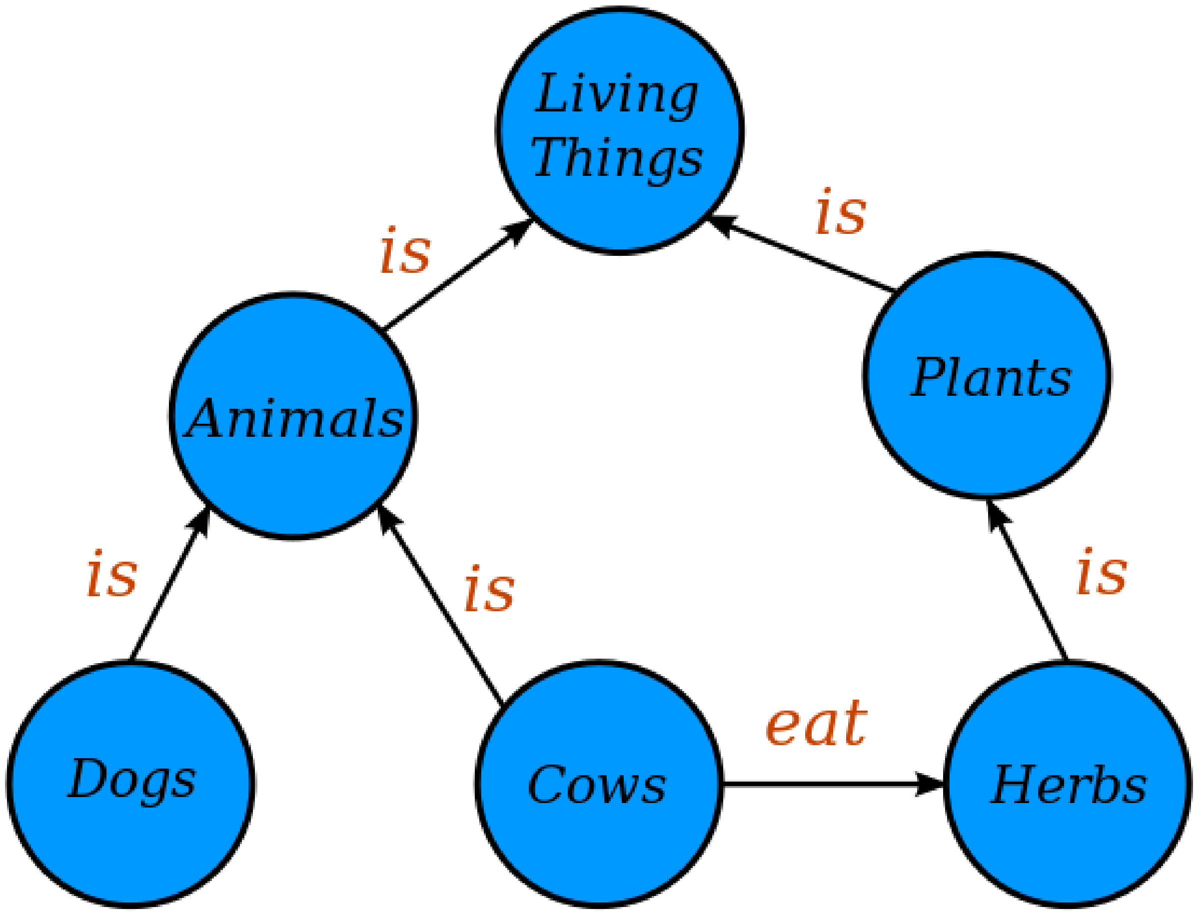
What is a Knowledge Graph

- A Knowledge Base, structured on a graph
- Defines the “things” that our knowledge is about – the **nodes** of the graph
- Defines the relationships between the “things’” – the **edges** of the graph
- Defines the properties of the “things” – the **attributes** of the nodes
- Rich structure, often with substructure (clustering of nodes)
- Normalised
- Explicit and declarative – has an intended meaning and is itself meaningful
- Often large – millions of nodes or more
- Human and machine readable



A simple example

- Obeys a “grammar”
- Relationships are explicitly defined
- Loose hierarchy defined by the relationships, not imposed
- Nodes can be directly annotated
- Normalised
- Has a clear meaning



A More Complex Example

- How might you model the relationships between:
  - Me, Lu Bai, Hui Wang
  - All of you (A. Turing, G. Hopper)
  - Knowledge Engineering, Machine Learning, Into to AI modules

  - What nodes do we need?
  - What relationships do we need to define?

Exercises 9: A simple knowledge graph

QUEEN'S UNIVERSITY BELFAST

MSc Artificial Intelligence

Knowledge Engineering

22 November 2023

What can Knowledge Graphs do?

- Explicitly model relations between things
- Model relations between classes of things
- Model hierarchies of things
- Be used to identify patterns
- Be used directly to derive new knowledge by running logical inference on top of a knowledge graph

What’s wrong with a Relational DB?

- All of this could be done with a SQL database
- The information is there, but it is hardly transparent
- Information spread over many tables
- Not really human readable

uid (PK)	staffID	Name
1	12345	Iain Styles
2	54321	Hui Wang
3	12321	Lu Bai

uid (PK)	moduleID	Name	Taught By
11	45678	Knowledge Engineering	12345
12	56789	Machine Learning	54321
33	67890	Intro to AI	12321

Why Graphs?

- Graphs are a natural way of representing relationships
  - “Things” = vertices on the graph
  - “Relationships” = edges on the graph

- Rich mathematical structure entail many powerful algorithms

- Totally flexible abstract data structure and can be structured to model

- Useful to have in mind some basic properties of graphs

Graph Basics

- A graph consists of a set *V* of nodes or vertices and a set *E* of edges
- Each edge is a tuple of two vertices
- Vertices v_1 and v_2 are *adjacent* (ie, connected) if (v_1, v_2) is in E
- Edges can be directed
- Can represent in terms of matrices

How to structure graphs

- Graphs provide a powerful mathematical foundation
- But we need to structure them to model a specific problem
- Specifically, we need to decide what features of a problem a graph should represent
- And what relationships it should include.
- This can be specified by an **ontology**

Ontologies

- An Ontology is a formal way of specifiying the “things” in a domain, their properties, and the relationships between them.

- It is a schema for modelling the problem

- Derived from domain knowledge

- Knowledge graphs are structured according to an ontology

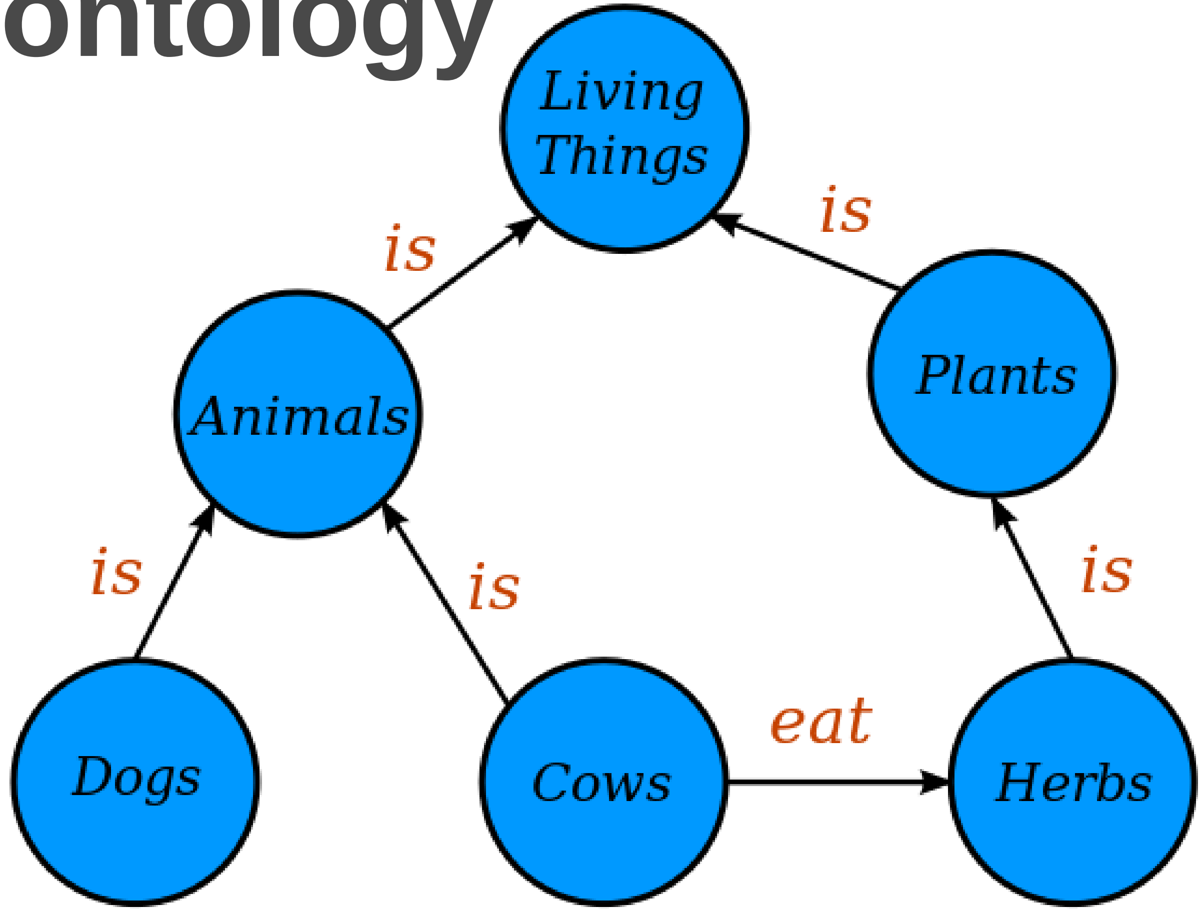
- Specific facts/data are then instantiated

Steps in creating an ontology

1. Define the entities – the fundamental concepts in the domain?

2. What are the relationships between the entities?

3. What properties do the entities have?

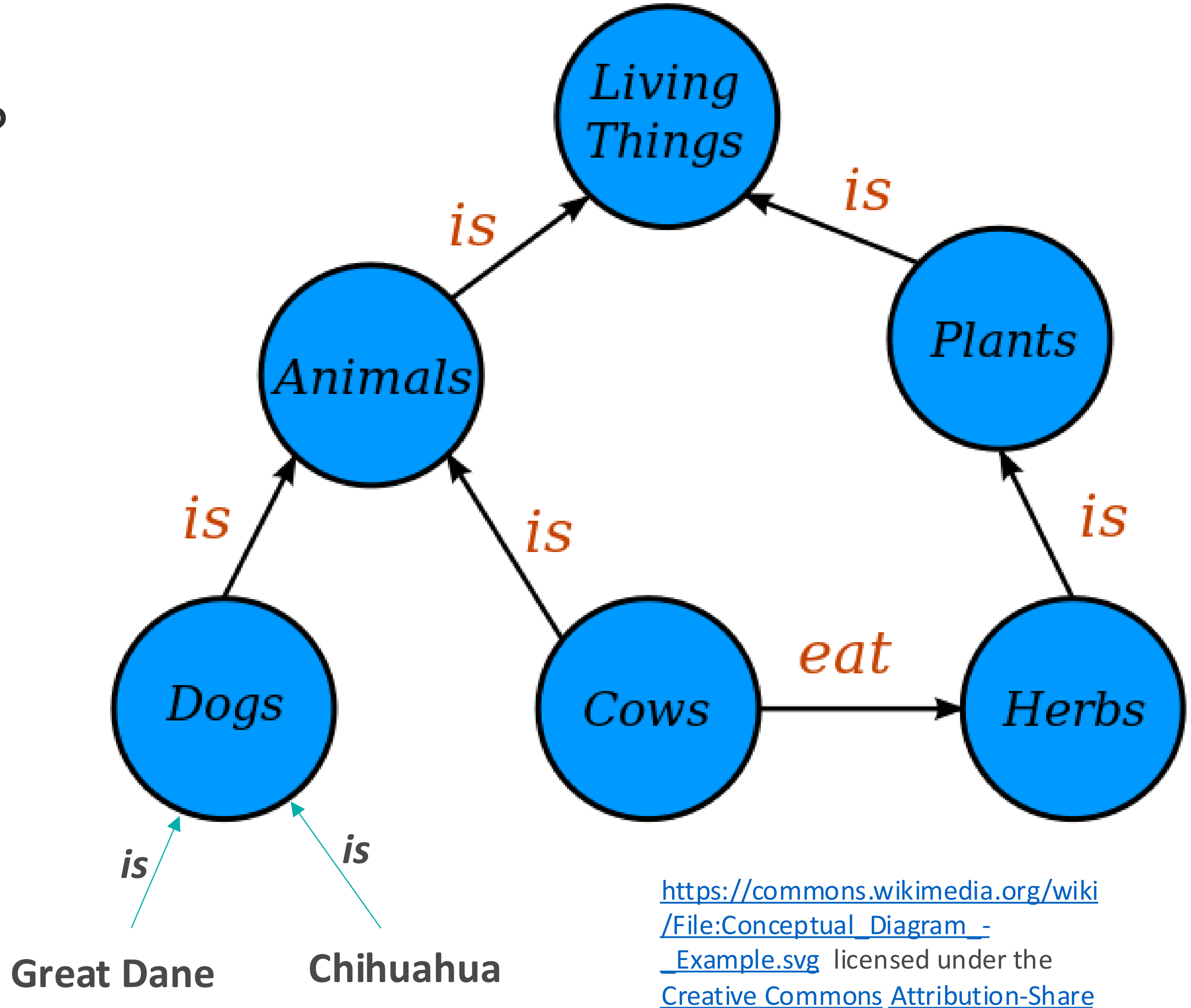


Entities

What are the fundamental concepts?

- Living things?
- Animals?
- Plants?
- Dogs?
- Cows?
- Herbs?

It depends…



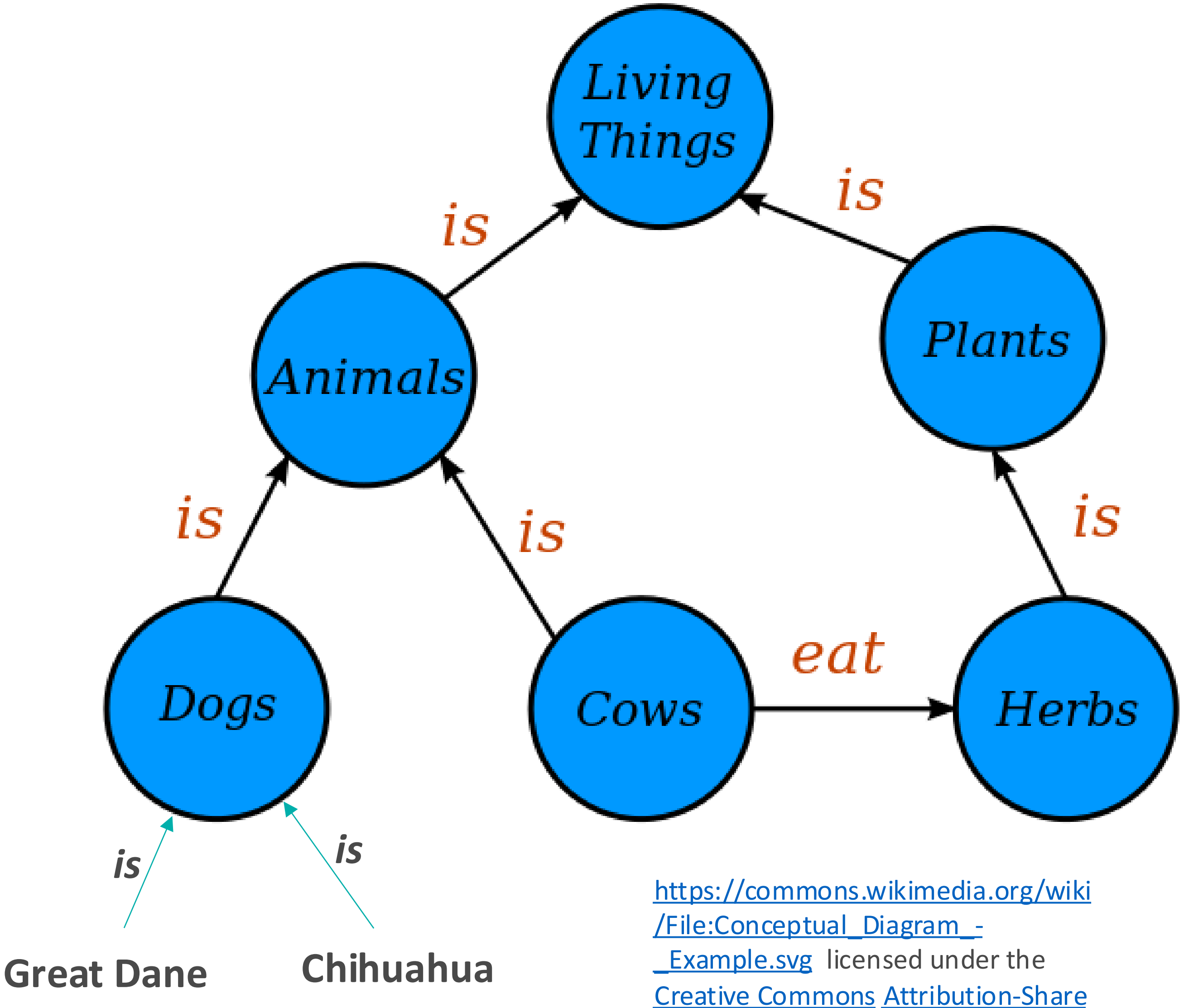
Relationships

What are the fundamental relations between entities?

- Is? (belongs to)
- Eats?
- Is toxic to?

- Are relationships meaningfully bidirectional?

It depends…

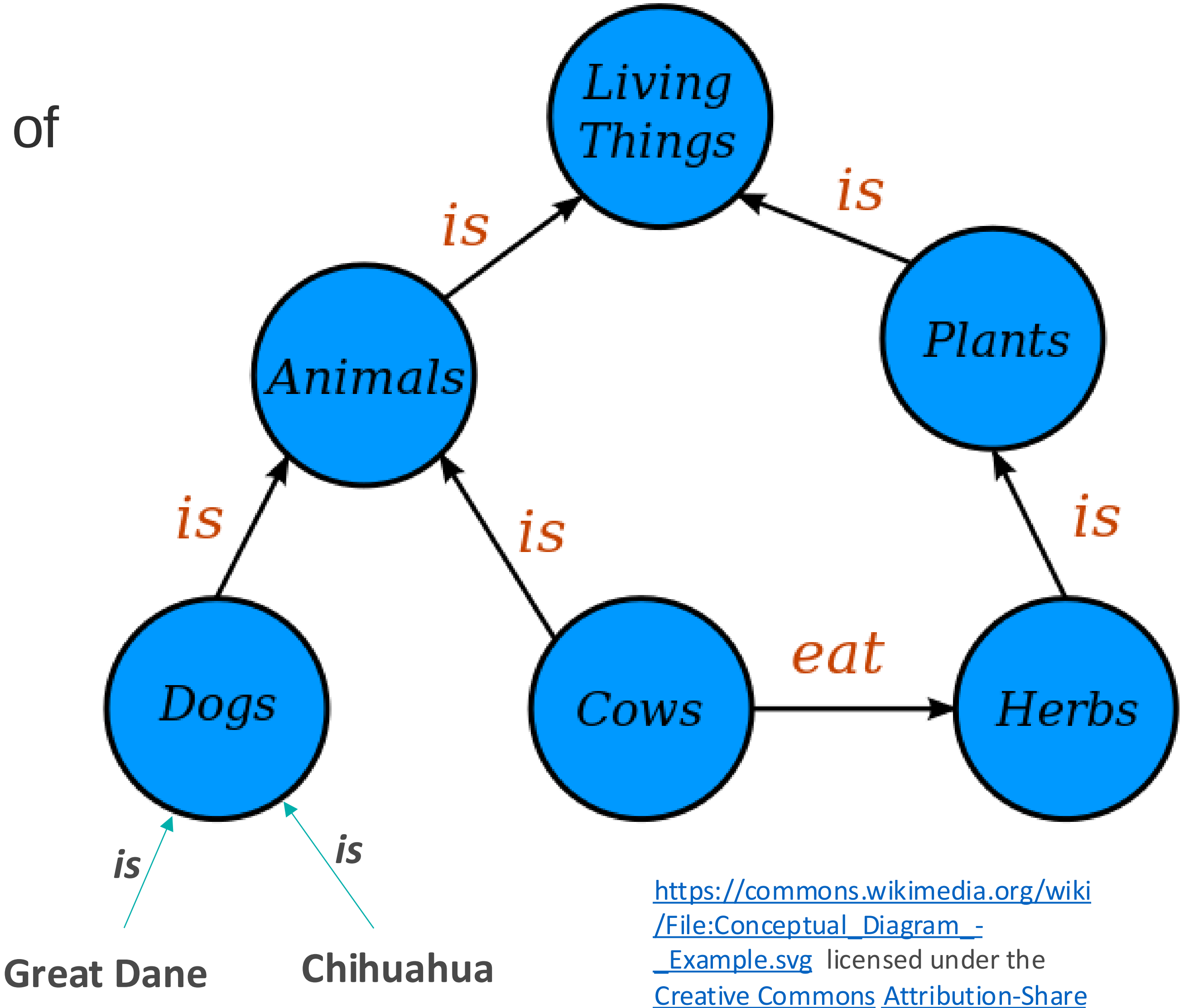


Properties

What are the fundamental properties of the entities?

- Latin name?
- Typical size?
- Where it is found?

It depends…



Ontology Guidelines

- Does not need to represent everything we know
- Not all relationships will matter
- Not all properties will be important

- Is the colour of a tablet relevant?
  - No, if you are a doctor or a patient
  - Yes, if you are responsible for manufacturing it

Exercises 10: Designing an Ontology

QUEEN'S UNIVERSITY BELFAST

MSc Artificial Intelligence

Knowledge Engineering

22 November 2023

Examples of different ontology formats

(is_a, dog, mammal)

(is_a, cat, mammal)

(chases, dog, cat)

- Simple Triples format
- Easy to parse
- Portable
- Lacks metadata
- Extensible

Examples of different ontology formats

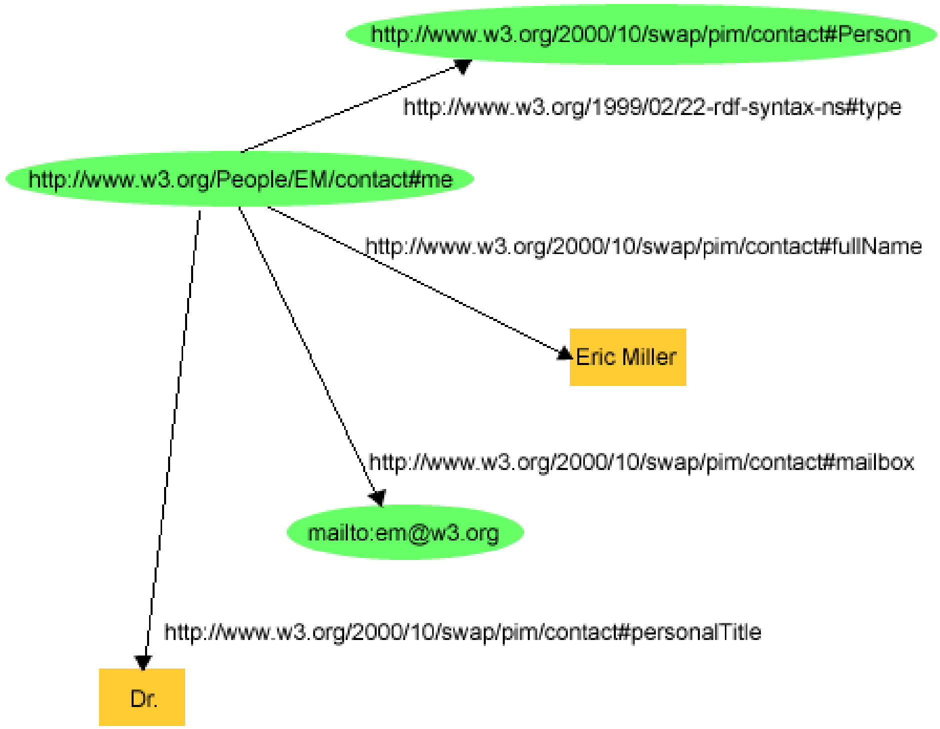
<http://www.w3.org/People/EM/contact#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/2000/10/swap/pim/contact#Person> .

<http://www.w3.org/People/EM/contact#me> <http://www.w3.org/2000/10/swap/pim/contact#fullName> "Eric Miller" .

<http://www.w3.org/People/EM/contact#me> <http://www.w3.org/2000/10/swap/pim/contact#mailbox> <mailto:em@w3.org> .

<http://www.w3.org/People/EM/contact#me> <http://www.w3.org/2000/10/swap/pim/contact#personalTitle> "Dr." .

- W3C N-Triples format
- Subject– Predicate – Object
- Designed for web data
- Referenced against external W3C definitions



http://www.w3.org/TR/rdf-primer/fig1dec16.png

Turtle: Terse RDF Triple Language

```
prefix ns0: <http://www.w3.org/2000/10/swap/pim/contact#> .

http://www.w3.org/People/EM/contact#me a <http://www.w3.org/2000/10/swap/pim/contact#Person> ;
ns0:fullName "Eric Miller" ;
ns0:mailbox <mailto:em@w3.org> ;
ns0:personalTitle "Dr." .
```

- Less repetitive and verbose than N-Triples
- Namespaces and nesting

RDF/XML

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

- Very similar but a little more verbose than Turtle
- XML-based so can use generic parsers

OWL: Web Ontology Language

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:owl="http://www.w3.org/2002/07/owl#"
        xml:base="http://www.dummy.info/new.owl"
        xmlns="http://www.dummy.info/new.owl#">

<owl:Ontology rdf:about="http://www.dummy.info/new.owl"/>

<owl:Class rdf:about="#Animal">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>

<owl:Class rdf:about="#Dog">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>

<owl:Class rdf:about="#Cat">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>

<owl:ObjectProperty rdf:about="#is_a">
  <rdfs:domain rdf:resource="#Dog"/>
  <rdfs:range rdf:resource="#Animal"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#is_a">
  <rdfs:domain rdf:resource="#Cat"/>
  <rdfs:range rdf:resource="#Animal"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="#chases">
  <rdfs:domain rdf:resource="#Dog"/>
  <rdfs:range rdf:resource="#Cat"/>
</owl:ObjectProperty>

</rdf:RDF>
```

- Like many XML formats, can be very verbose.
- XML tooling.
- Referenced against external definitions.

Notes on formats

- Ontology representations designed to be “human readable” and machine parseable.

- In practice, nearly always read/written by tools and read by humans only when strictly necessary.

QUEEN'S UNIVERSITY BELFAST

MSc Artificial Intelligence

Knowledge Engineering

22 November 2023

How do we specify an ontology?

- Web Ontology Language (OWL)

- We'll study this by example later

- We do not write this by hand: use a tool: https://owlready2.readthedocs.io/en/v0.42/

conda install -c conda-forge owlready2

- Let's code…

Exercises 11: Extending an Ontology

Where to find out more?

- There is a lot more you can do

- I found all of the following helpful

- https://owlready2.readthedocs.io/en/v0.42/
- https://kgtutorial.github.io/
- https://jingdongsun.medium.com/creating-knowledge-graph-step-by-step-a383231acf2d
- https://iccl.inf.tu-dresden.de/web/Knowledge_Graphs_(WS2021)/en

Exercises 12: Implementing a Complete Ontology