- Project #2 - proposals due today - discuss with TA!

"In Prolog, as in most halfway decent programming languages, there is no tension between writing a beautiful program an writing an efficient program. If your Prolog code is ugly, the chances are that you either don't understand your problem or don't understand your programming language, and in neither case does your code stand much chance of being efficient. In order to ensure your program is efficient, you need to know what it is doing, and if your code is ugly, you will find it hard to analyse."

Richard A. O'Keefe, "The Craft of Prolog", 1990.

Plan

Last time

- difference lists
- computer algebra and calculus
- definite clause grammars
- natural language interfaces to databases
- Knowledge graphs

Future

- Knowledge graphs
- Semantic web
- Negation as failure
- Proofs

Triples are universal representations of relations

All relations can be represented in terms of triples:

	...	P_j	...
	...	...	...
r_i	...	v_{ij}	...
	...	...	...

can be represented as the triple $\langle r_i, P_j, v_{ij} \rangle$.

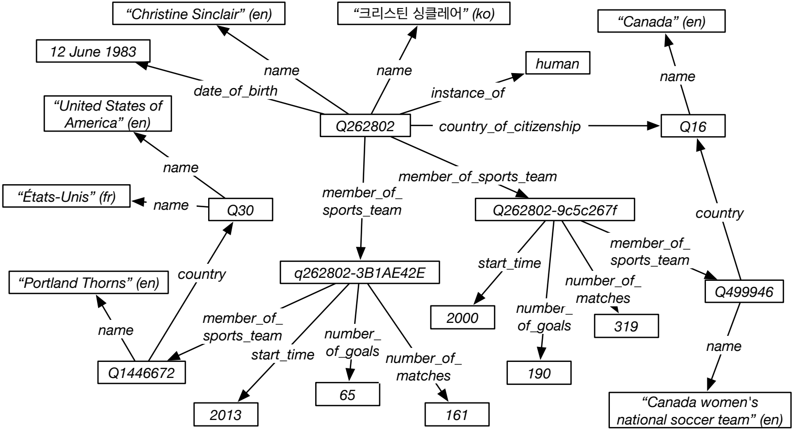
- r_i is either a primary key or a reified entity.
- Examples of reified entities: a booking, a marriage, flight number, transaction number, FIFA Wold Cup Final 2026.

Wikidata IRIs

- Wikidata (`https://www.wikidata.org`) is a free, collaborative knowledge graph with around 1.25 billion triples describing 100 million entities (as of 2022).
- The soccer player Christine Sinclair is represented using the identifier "`http://www.wikidata.org/entity/Q262802`"
- The identifier "http://schema.org/name" is the property that gives the name of the subject
- "`http://www.wikidata.org/prop/direct/P27`" is the property "country of citizenship".
- Canada is "`http://www.wikidata.org/entity/Q16`"
- "Christine Sinclair is a citizen of Canada":
  /entity/Q262802 /prop/direct/P27 /entity/Q16

  but all starting with `http://www.wikidata.org`

Part of the Wikidata Knowledge Graph



Clicker Question

In the query

```
?- rdf('http://www.wikidata.org/entity/Q34086',
        'http://www.wikidata.org/prop/direct/P25',M),
   rdf(M,'http://schema.org/name',MN).
```

the reason to use the constant `'http://schema.org/name'` is:

A to make it look complicated and impressive

B it has a standard meaning and everyone who uses that constant means the same thing

C because schema.org is sponsored by Google, Microsoft, Yahoo and Yandex, and they will be impressed if we use schema.org

D it is part of the semantic web, which is the future of the Internet

E there is no reason to use such a complicated constant when a simple one would do just as well.

Clicker Question

In the query

```
?- rdf('http://www.wikidata.org/entity/Q34086',
       'http://www.wikidata.org/prop/direct/P25',M),
   rdf(M,'http://schema.org/name',MN).
```

What is not a reason to use the constant `'http://www.wikidata.org/entity/Q34086'` instead of using his name `'Justin Bieber'`

- A the constant denotes the person, not the name
- B it has a standard meaning and everyone who uses that constant means the same thing
- C there may be multiple people called 'Justin Bieber' and the constant denotes a particular one
- D the constant is easier for people to find and remember
- E these are all reasons

Semantic Web Technologies

- XML the Extensible Markup Language provides generic syntax.
  $\langle tag \ldots / \rangle$ or
  $\langle tag \ldots \rangle \ldots \langle / tag \rangle$.
- IRI an Internationalized Resource Identifier is a constant denoting an individual (resource). This name can be shared. Often in the form of a URL to ensure uniqueness.
- RDF the Resource Description Framework is a language of triples
- OWL the Web Ontology Language, defines some primitive properties that can be used to define terminology. (Doesn't define a syntax).

Triple Store

A triple store stores triples in a way that allows for efficient retrieval of arbitrary queries.
Example queries:

- $rdf(S, V, O)$.
- $rdf(sub1, v173, o765)$
- $rdf(sub1, V, O)$
- $rdf(sub1, v173, O)$

How many indexes are needed so all such queries can be implemented efficiently?

Triple Store

- Triple store can be implemented very efficiently with eighthow many indexes.?
- SWI Prolog can store about 250 million triples on a 64-bit machine with 64 Gb memory, and retrieve them efficiently.
- Wikidata `https://www.wikidata.org` contains about 1.25 billion triples. (Curated. Anyone can edit.)
- DBpedia `https://www.dbpedia.org` contains 15 billion triples 3.6TB of data reachable (extracts structured data from Wikipedia, and publishes them in RDF)
- Google's Knowledge Graph, contains 500 billion facts on 5 billion entities.

  `https://blog.google/products/search/about-knowledge-graph-and-knowledge-panels/`

  Much of the data is from marked-up web pages; see `http://schema.org/`.

Clicker Question

What is not a reason for using triples as a representation for relations:

A They can be indexed efficiently, whereas arbitrary relations may require too many indexes or are restricted to index on given keys

B Extra arguments to the relation can be added simply

C They allow for more flexible queries

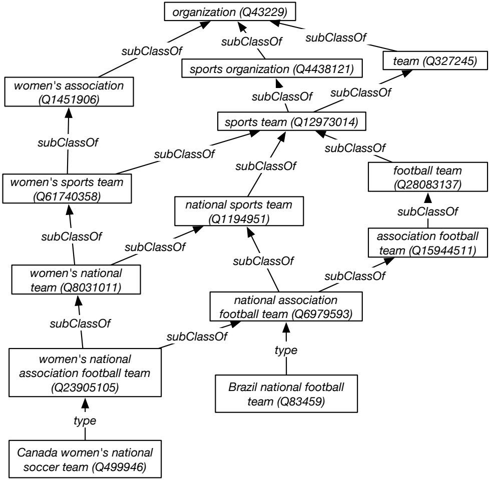
D These are all reasons

Classes and Properties

- Primitive knowledge is knowledge that is specified explicitly.
- Derived knowledge is knowledge that can be inferred from primitive knowledge and other derived knowledge.
- A standard way to use derived knowledge is to specify attributes (property–value pairs) for all members of a class.
- Individuals inherit the attributes associated with the classes they are in.
- A natural kind is a class such that describing individuals using the class is more succinct than describing individuals without the class.
- E.g., Specifying all *mammals* are warm blooded is more concise than specifying each individual is warm blooded.
- A class is the set of those actual and potential individuals that would be members of the class.
  E.g., class of chairs includes chairs that have not been build or may never be built.

Class Hierarchies

- Class S is a subclass of class C means S is a subset of C.
  $subClassOf(S, C) \wedge type(E, S) \rightarrow type(E, C)$.
- Part of the Wikidata class structure:



Property Hierarchies

- Property p_1 is a sub-property of property p_2 if every pair related by p_1 is also related by p_2.
- $subPropertyOf(p_1, p_2) \wedge p_1(x, y) \rightarrow p_2(x, y)$

Some sub-properties in Wikidata are

- "member of sports team" (P54) is a sub-property of "member of" (P463)
- "member of" (P463) is a sub-property of "affiliation" (P1416)
- "affiliation" (P1416) is a sub-property of "part of" (P361)
- "part of" (P361) is a sub-property of "partially coincident with" (P1382)

Property

- The domain of a property is a class such that the subject of a triple with the property has to be in the class.
  $domain(p, C) \wedge p(x, y) \rightarrow type(x, C)$.
- domain of "member of sports team" (P54) is "human" (Q5)
- The range of a property is a class such that the object of a triple with the property has to be in the class.
  $range(p, C) \wedge p(x, y) \rightarrow type(y, C)$.
- range of "member of sports team" (P54) is "sports team" (Q12973014)
- Property p is functional means there is at most one object associated with any subject.
  $p(x, y_1) \wedge p(x, y_2) \rightarrow y_1 = y_2$.
- Is "member of sports team" (P54) functional?
- Is "date of birth" (P569) functional?

These are not (currently) part of Wikidata!

Designing Classes

Aristotle [350 B.C.] suggested the definition if a class C in terms of:

- Genus: a super-class
- Differentia: the attributes that make members of the class C different from other members of the super-class

"If genera are different and co-ordinate, their differentiae are themselves different in kind. Take as an instance the genus 'animal' and the genus 'knowledge'. 'With feet', 'two-footed', 'winged', 'aquatic', are differentiae of 'animal'; the species of knowledge are not distinguished by the same differentiae. One species of knowledge does not differ from another in being 'two-footed'."

Aristotle, *Categories*, 350 B.C.

Designing Classes

The art of ranking things in genera and species is quite important, and greatly helps our judgment as well as our memory. . . . Order largely depends on it, and many good authors write in such a way that their whole account could be divided and subdivided according to a procedure related to genera and species. This helps one not merely to retain things in one's memory, but also to find them there. Writers who have laid out all sorts of notions under certain headings or categories have done something very useful.

– G. W. Leibniz [1705]

Designing Classes

To design classes based on Aristotelian definitions:

- For each class: determine a relevant superclass and then select those attributes that distinguish the class from other subclasses.
- Each attribute gives a property and a value.
- For each property, define the domain to be most general class for which it makes sense.
- Make the range of the property another class that makes sense (perhaps requiring this range class to be defined, either by enumerating its values or by defining it using an Aristotelian definition).

Designing Classes

Aristotelian definitions result in class structure that is a lattice, and rarely a tree.

Example: Consider definitions of rectangle, rhombus, and square:

- A quadrilateral is a planar figure made up of four straight sides.
- A rectangle is a quadrilateral where all inside angles are right angles (90°).
- A rhombus is a quadrilateral where all four sides have the same length.
- A square is a quadrilateral where all four sides have the same length and all inside angles are right angles.

What is a most specific superclass of square?
A square is both a rectangle and a rhombus.
Neither is more specific than the other.