- Project #2 - get started!

"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
- definite clause grammars
- natural language interfaces to databases
- computer algebra and calculus

Today

- Knowledge graphs
- Semantic web

Knowledge Graphs and Ontologies

- Is there a flexible way to represent relations?
- How can knowledge bases be made to interoperate semantically?

Choosing Individuals and Relations

How to represent: "Pen #7 is red."

- $red(pen_7)$. It's easy to ask "What's red?"
  Can't ask "what is the color of pen_7?"
- $color(pen_7, red)$. It's easy to ask "What's red?"
  It's easy to ask "What is the color of pen_7?"
  Can't ask "What property of pen_7 has value red?"
- $prop(pen_7, color, red)$. It's easy to ask all these questions.

$prop(Individual, Property, Value)$ is the only relation needed:
called individual-property-value representation
or triple representation

Universality of *prop*

To represent "a is a parcel"

- $prop(a, type, parcel)$, where *type* is a special property. Then *parcel* is a class.
- $prop(a, parcel, true)$, where *parcel* is a Boolean property. Here parcel is the characteristic function of the class.

Reification

- To represent $scheduled(cs312, 101, 1200, dmp310)$. "section 101 of course $cs312$ is scheduled at 12:00 in room $dmp310$."
- Let $b123$ name the booking:

  $prop(b123, course, cs312).$

  $prop(b123, section, 101).$

  $prop(b123, time, 1200).$

  $prop(b123, room, dmp310).$
- We have reified the booking.
- Reify means: to make into an individual.
- What if we want to add the year?
- What if we want to add the instructor?

When you only have one relation, *prop*, it can be omitted without loss of information.

Logic:

$prop(Individual, Property, Value)$ or

$rdf(Individual, Property, Value)$

triple:

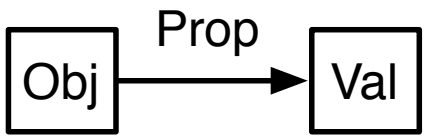
$\langle Individual, Property, Value \rangle$

simple sentence:

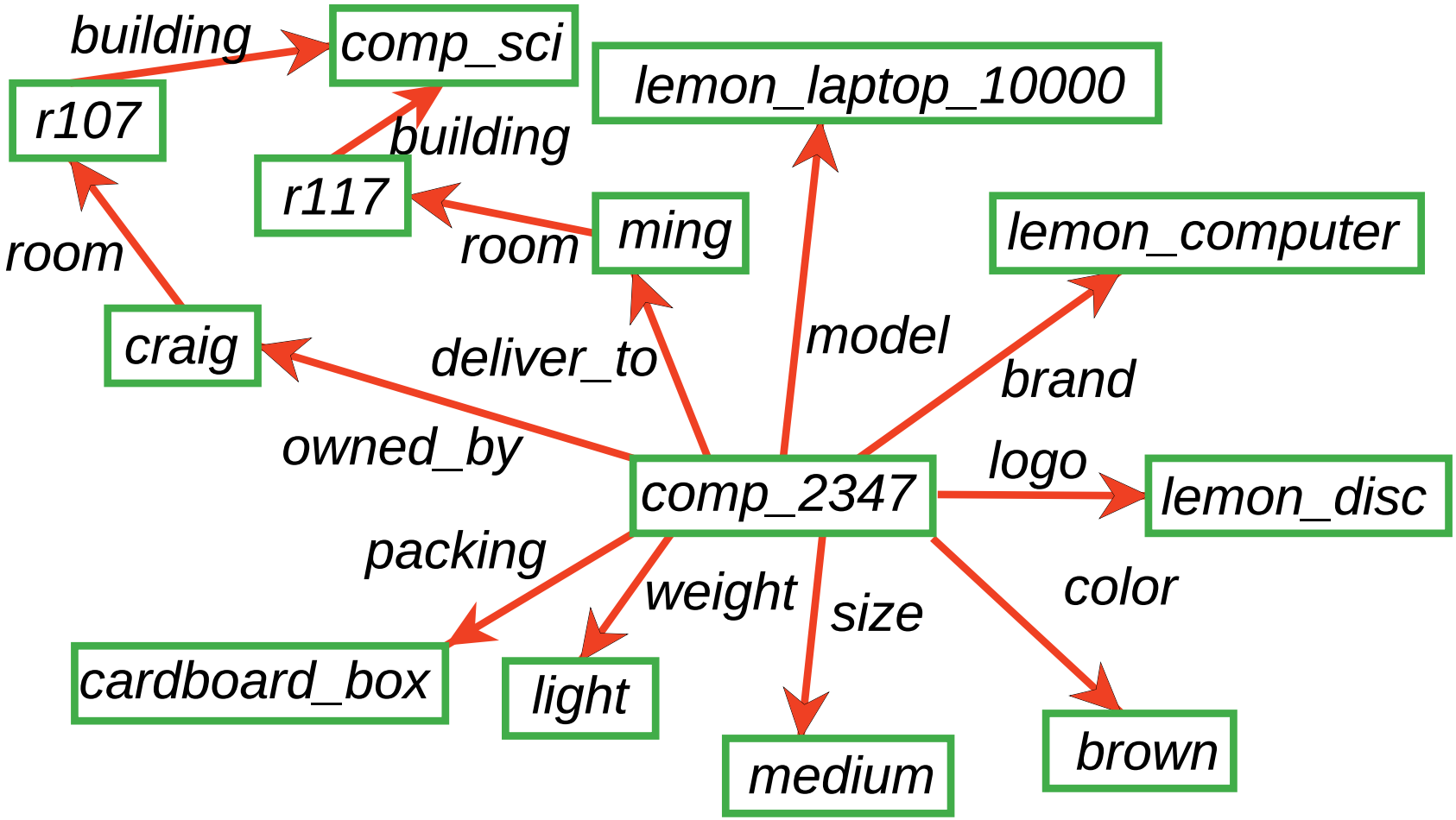
Individual Property Value.

Subject Predicate Object.

graphically:



An Example Semantic Network



Equivalent Logic Program

$prop(comp\_2347, owned\_by, craig).$

$prop(comp\_2347, deliver\_to, ming).$

$prop(comp\_2347, model, lemon\_laptop\_10000).$

$prop(comp\_2347, brand, lemon\_computer).$

$prop(comp\_2347, logo, lemon\_disc).$

$prop(comp\_2347, color, brown).$

$prop(craig, room, r107).$

$prop(r107, building, comp\_sci).$

$\vdots$

Semantic Web Technologies

- XML the Extensible Markup Language provides generic syntax.
  $\langle tag \ldots /\rangle$ or
  $\langle tag \ldots \rangle \ldots \langle /tag \rangle$.
- URI a Uniform Resource Identifier is a constant denoting an individual (resource). This name can be shared. Often in the form of a URL to ensure uniqueness.
  E.g., `https://www.w3.org/People/Berners-Lee/card#i`
  `http://www.cs.ubc.ca/~poole/foaf.rdf#david`
- 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

- Triple store can be implemented very efficiently with eighthow many indexes.?
- SWI Prolog can store 300,000,000 triples, and retrieve them efficiently.
- Wikidata `https://www.wikidata.org/wiki/Wikidata:Main_Page` contains about 37 million triples. (Curated.)
- See `http://www.cs.ubc.ca/~poole/cs312/2018/prolog/sem_web.pl`
- Google's Knowledge Graph, contains 70 billion triples. Much of the data is from marked-up web pages; see `http://schema.org/`.
- Google's Knowledge Vault contains 1.6 billion triples. (Learned).

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

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

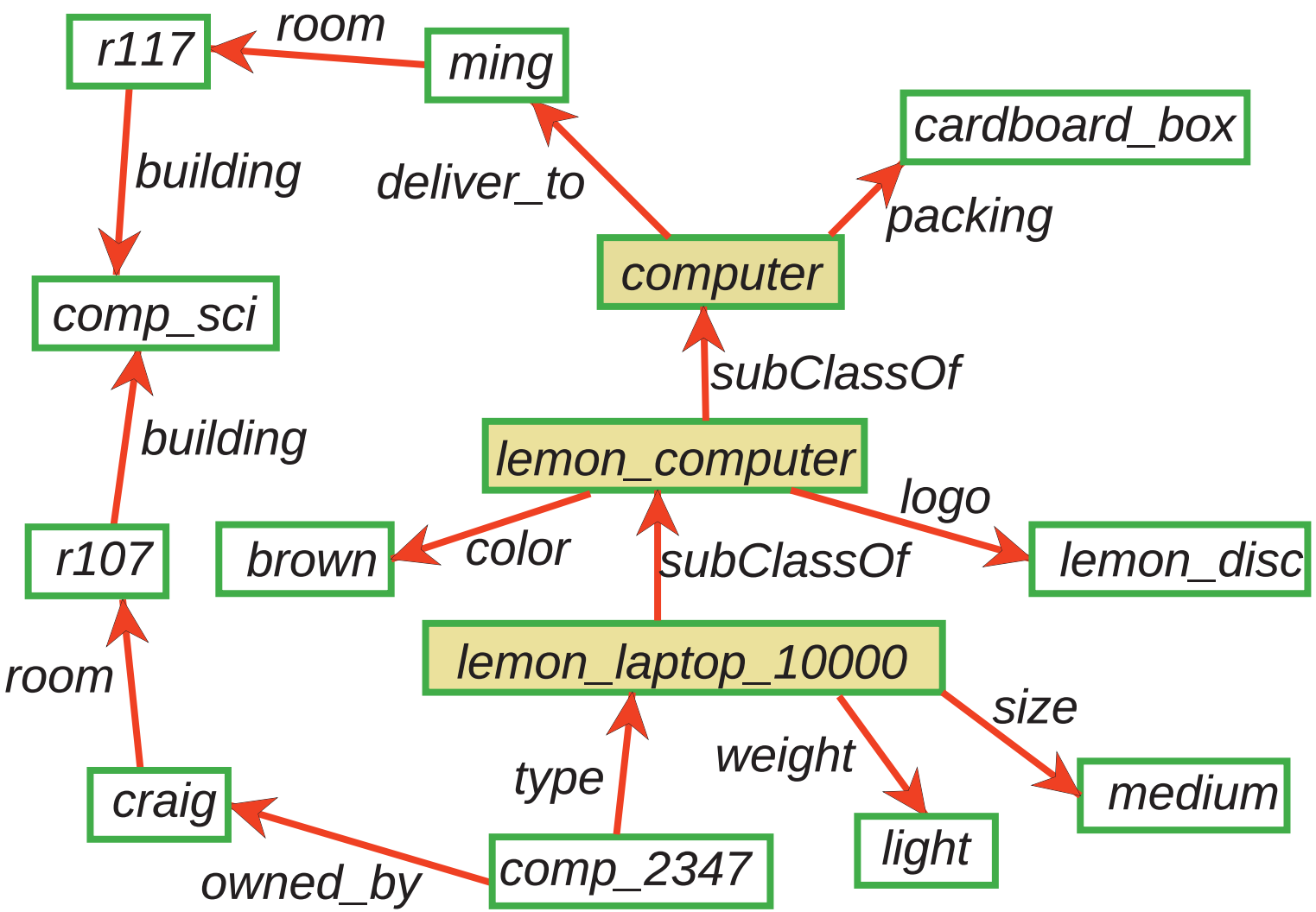
B it has a standard meaning and everyone who uses that constant means the same thing

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

A Structured Semantic Network



Logic of Classes

Distinguish

- Properties of classes
- Properties of individuals in classes

  An arc $c \xrightarrow{p} v$ from a class c with a property p to value v means every individual in the class has value v on property p:

  $$prop(Obj, p, v) \text{ :-}$$
  $$\quad prop(Obj, type, c).$$

  Example:

  $$prop(X, weight, light) \text{ :-}$$
  $$\quad prop(X, type, lemon\_laptop\_10000).$$
  $$prop(X, packing, cardboard\_box) \text{ :-}$$
  $$\quad prop(X, type, computer).$$

Logic of Property Inheritance

You can do inheritance through the subclass relationship:

$$prop(X, type, T) \text{ :-}$$
$$\quad prop(S, subClassOf, T),$$
$$\quad prop(X, type, S).$$

Multiple Inheritance

- An individual is usually a member of more than one class. For example, the same person may be a wine expert, a teacher, a football coach, and a mother.
- The individual can inherit the properties of all of the classes it is a member of: multiple inheritance.
- With default values,what is an individual inherits conflicting defaults from the different classes? multiple inheritance problem.

Choosing Primitive and Derived Properties

- Associate a property value with the most general class with that property value.
- Don't associate contingent properties of a class with the class. For example, if all of current computers just happen to be brown.

Knowledge Sharing

- A conceptualization is a map from the problem domain into the representation. A conceptualization specifies:
  - What sorts of individuals are being modeled
  - The vocabulary for specifying individuals, relations and properties
  - The meaning or intention of the vocabulary
- If more than one person is building a knowledge base, they must be able to share the conceptualization.
  $\longrightarrow$ challenge: inter-operability of separately designed knowledge bases
- An ontology is a specification of a conceptualization.
  An ontology specifies the meanings of the symbols in an information system.