Primitive versus Derived Relations

**Primitive knowledge** is that which is defined explicitly by facts.

**Derived knowledge** is knowledge defined by rules.

**Example:** All lemon laptops may have have size = *medium*. Associate this property with the class, not the individual.

Allow a special attribute **is_a** between an individual and a class or between two classes that allows for **property inheritance**.
A Structured Semantic Network

- r117 is a room
- ming is a room
- cardio_board_box is a packing
- building is a room
- deliver_to
- computer is a is_a
- logo
- lemon_disc is a color brown
- size medium
- weight light
- packing
- is_a
- medium
- weight
- light
- is_a
- lemon_computer
- lemon_laptop_10000
- comp_sci is_a
- building
- room
- owned_by
- craig
- comp_2347
Logic of Property Inheritance

An arc $p \rightarrow n$ from a class $c$ means every individual in the class has value $n$ of attribute $p$:

$$prop(Obj, p, n) \leftarrow prop(Obj, is\_a, c).$$

Example:

$$prop(X, weight, light) \leftarrow prop(X, is\_a, lemon\_laptop\_10000).$$
$$prop(X, is\_a, lemon\_computer) \leftarrow prop(X, is\_a, lemon\_laptop\_10000).$$
Multiple Inheritance

▶ An individual is usually a member of more than one class. For example, the same person may be a mother, a teacher, a football coach, etc.

▶ The individual can inherit the properties of all of the classes it is a member of: multiple inheritance.

▶ If there are default values, we can have a problem when an individual inherits conflicting defaults from the different classes: multiple inheritance problem.
Choosing Primitive and Derived Relations

- Associate an attribute value with the most general class with that attribute value.

- Don’t associate contingent properties of a class with the class. For example, if all of current computers just happen to be brown.

- Axiomatize in the **causal** direction. You want knowledge that is stable as the world changes.