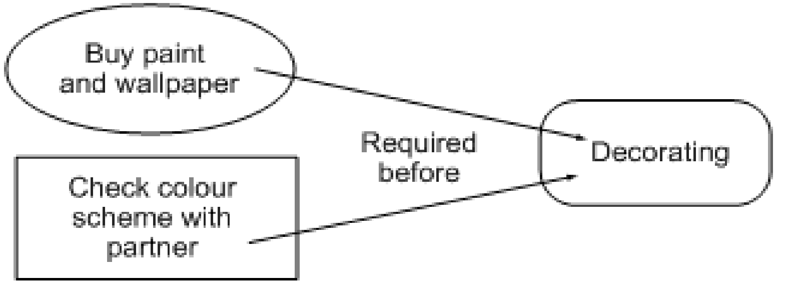
Other Knowledge Representation

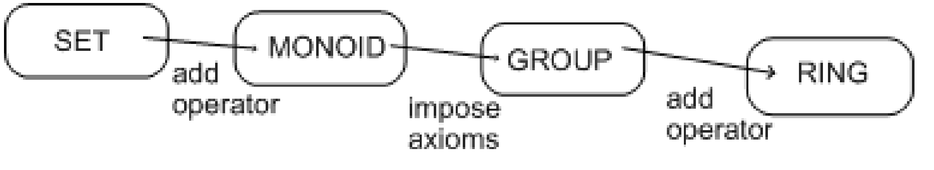
- Different Knowledge Representation Schema
- Semantic Net
- Frames
- Rule based systems

Graphical Representations

- Humans draw diagrams all the time, e.g.,
  – E.g. causal relationships:

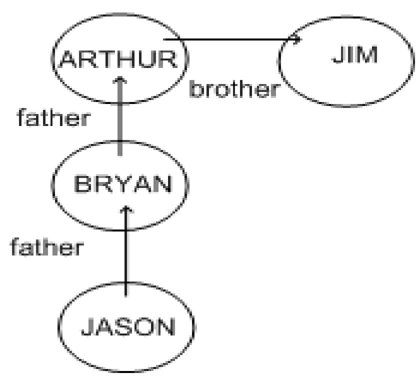


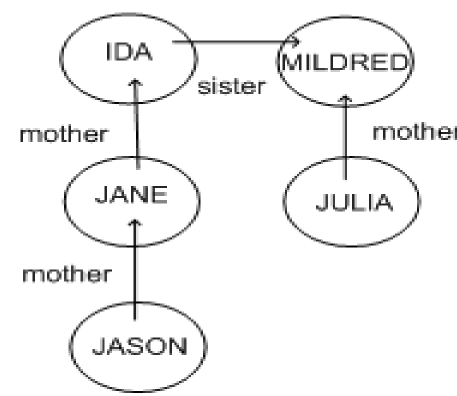
  – And relationships between ideas:



Graphical Representation(cont.)

- Graphs are very easy to be stored inside a computer
- For information to be of any use
  - We must impose a formalism on the graphs





  - Jason is 15, Bryan is 40, Arthur is 70, Jim is 74
  - How old is Julia?

Hierarchy and taxonomy

- Hierarchy or taxonomy is a natural way to view the world
  - importance of *abstraction in remembering and reasoning*
  - groups of things share properties in the world
  - we do not have to repeat representations
- **Example:**
- Saying “elephants are mammals” is sufficient to know a lot about them
- **Inheritance is the result of reasoning over paths in a hierarchy**

Semantic Nets

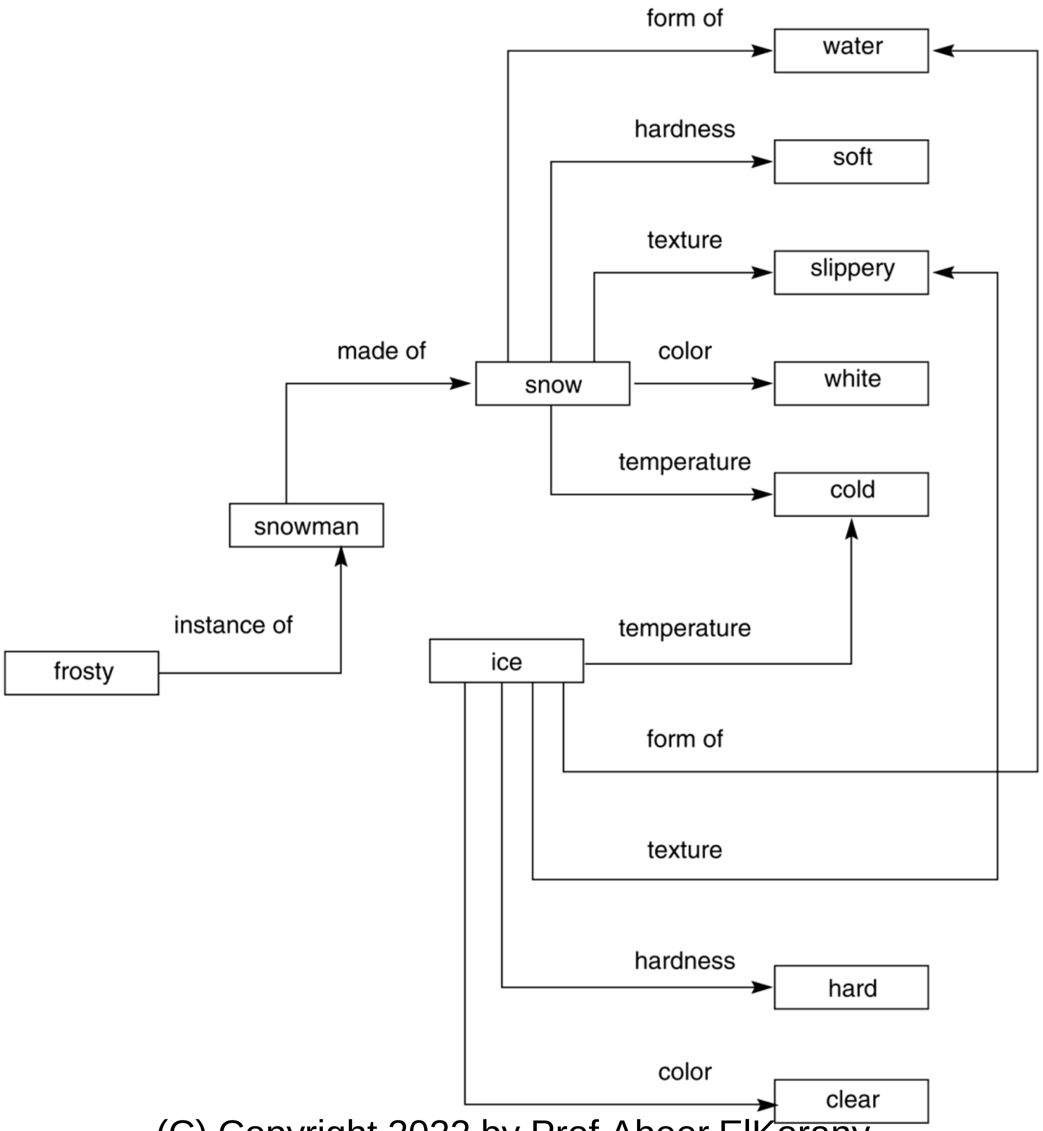
- Graphical way for representing declarative knowledge
- It is a labeled, directed graph consisting of vertices and edges.
- Knowledge represented as a set of nodes connected by labeled arcs
- Nodes stand for **objects**, **concepts**, **attributes** or **events**
- Arcs represent relationships between the nodes
  - the relationships contain the structural information of the knowledge to be represented
  - the label indicates the type of the relationship

Good for:

- **Semantic associations (taxonomies)**
- **Physical and causal structures**

A brief look at semantic networks

- A semantic network is an irregular graph that has concepts in vertices and relations on arcs.
- Relations can be ad-hoc, but they can also be quite general, for example, “is a” (ISA), “a kind of” (AKO), “an instance of”, “part of”.
- Relations often express physical properties of objects (colour, length, and lots of others).
- Most often, relations link two concepts.



Semantic Nets (cont.)

- Object representation
  - Two kinds of node to represent objects:
    - classes and instances (objects)
  - Two kinds of relationships between class/objects:
    - IsA relates an instance to its class
    - AKindOf relates a sub-class to its superclass
- Property representation
  - Attributes are represented by nodes and relationship to their concept are labeled on the arcs between them

Types of Relationships

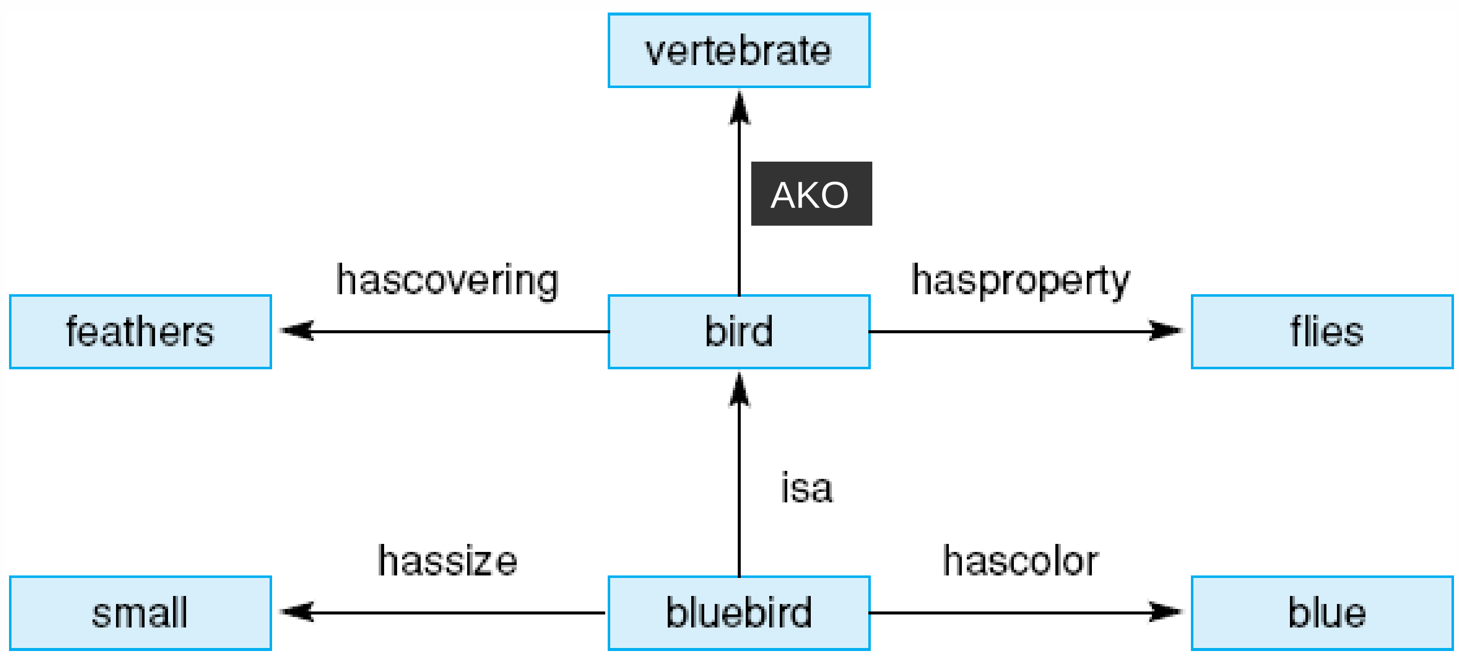
- Relationships can be arbitrarily defined by the knowledge engineer
  - allows great flexibility
  - for reasoning, the inference mechanism must know how relationships can be used to generate new knowledge
    - inference methods may have to be specified for every relationship
- Frequently used relationships
  - IS-A
    - relates an instance (individual node) to a class (generic node)
  - AKO (a-kind-of)
    - relates one class (subclass) to another class (superclass)

Objects and Attributes

- Attributes provide more detailed information on nodes in a semantic network
  - often expressed as *properties*
    - combination of attribute and value
  - attributes can be expressed as relationships
    - e.g. has-attribute

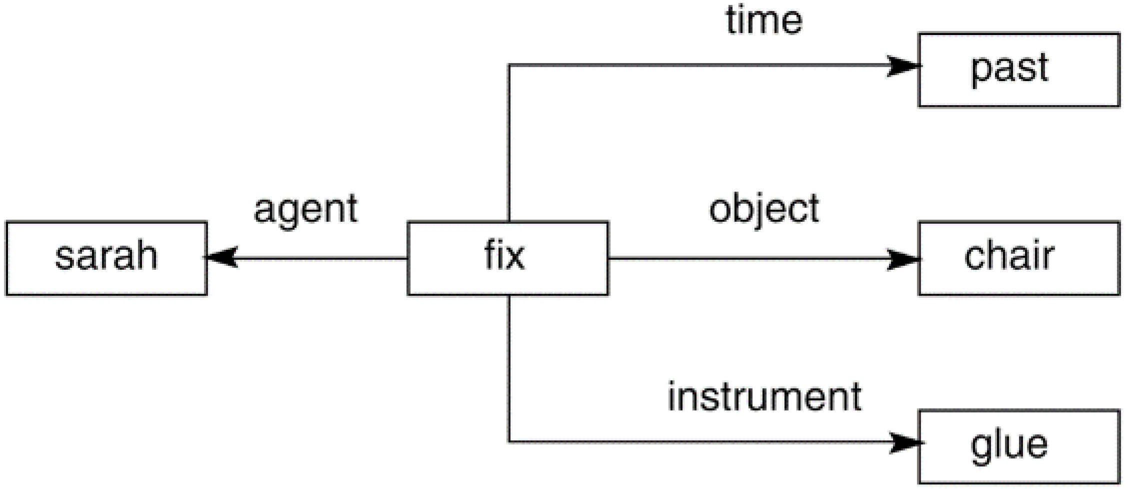
Semantic Nets (cont)

- Semantic nets allow you to define relations between objects, including class relations (X isa Y).
- objects and values of attributes of objects.



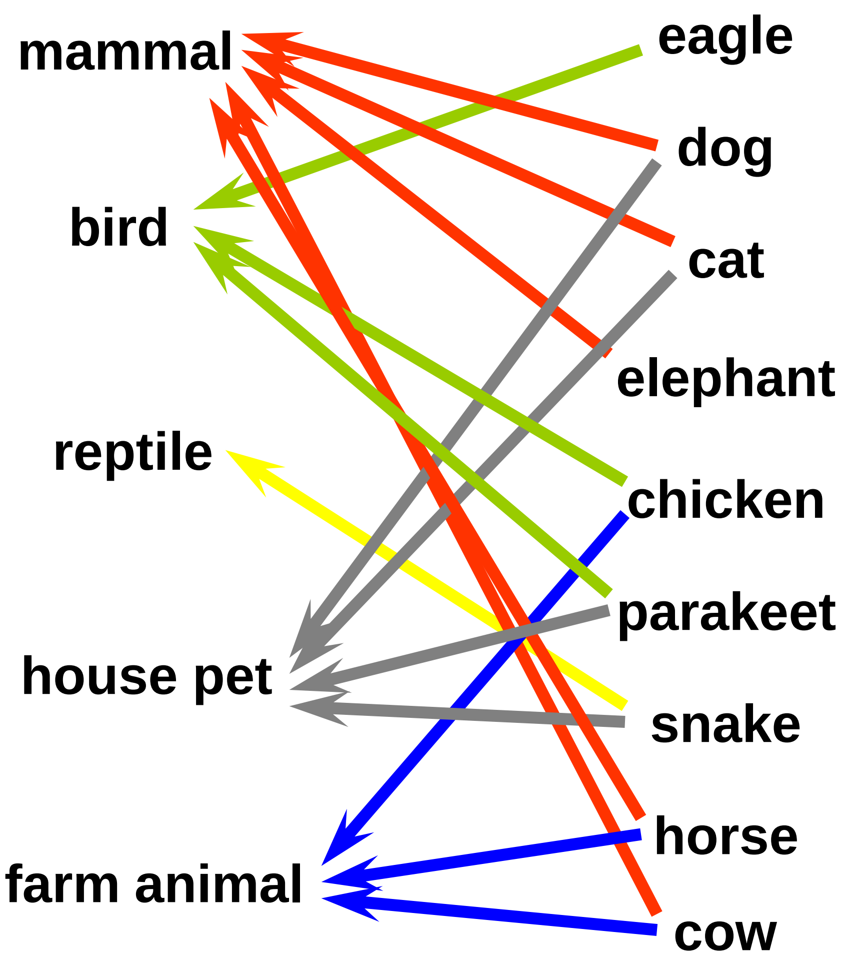
Semantic Network (cont.)

- A sentence is centered on a verb that expects certain arguments.
- For example, verbs usually denotes actions (with agents) or states (with passive experiencers, for example, “he dreams” or “he is sick”).



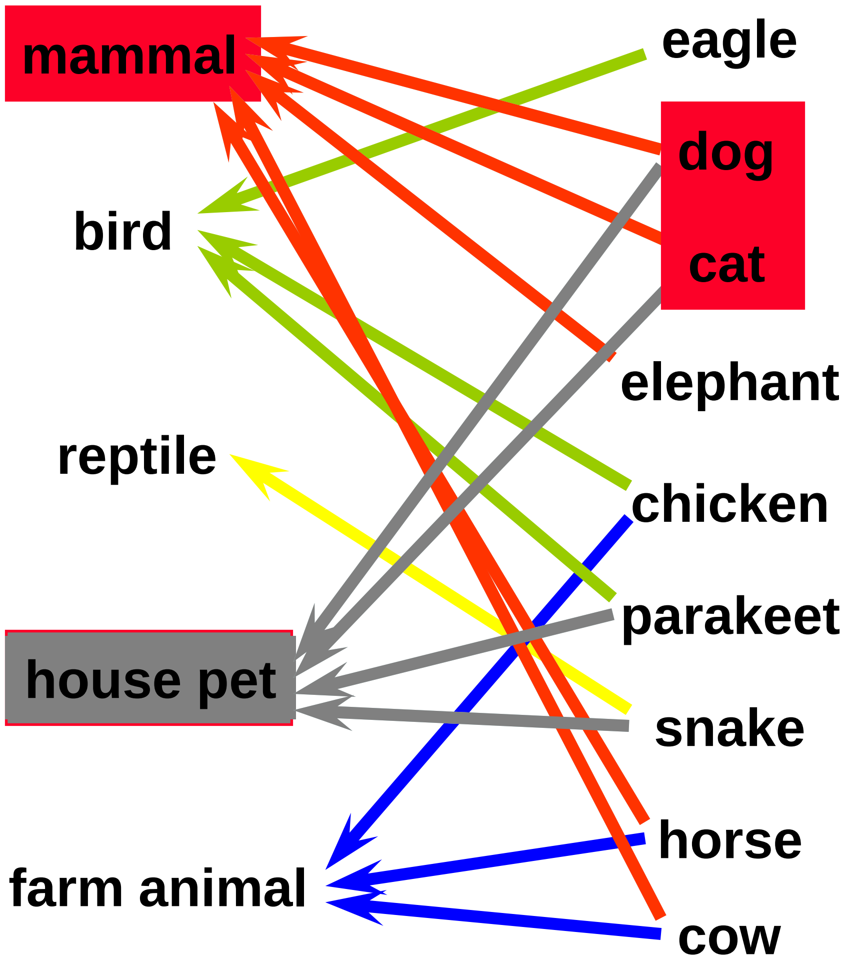
Taxonomy of Animal Knowledge As Semantic Network

eagles are birds
snakes are reptiles
elephants are mammals
dogs are mammals
cat are mammals
cows are mammals
cows are farm animals
chickens are farm animals
horses are farm animals
cats are house pets
dogs are house pets
chickens are birds
snakes are house pets
horses are mammals
parakeets are birds
parakeets are house pets

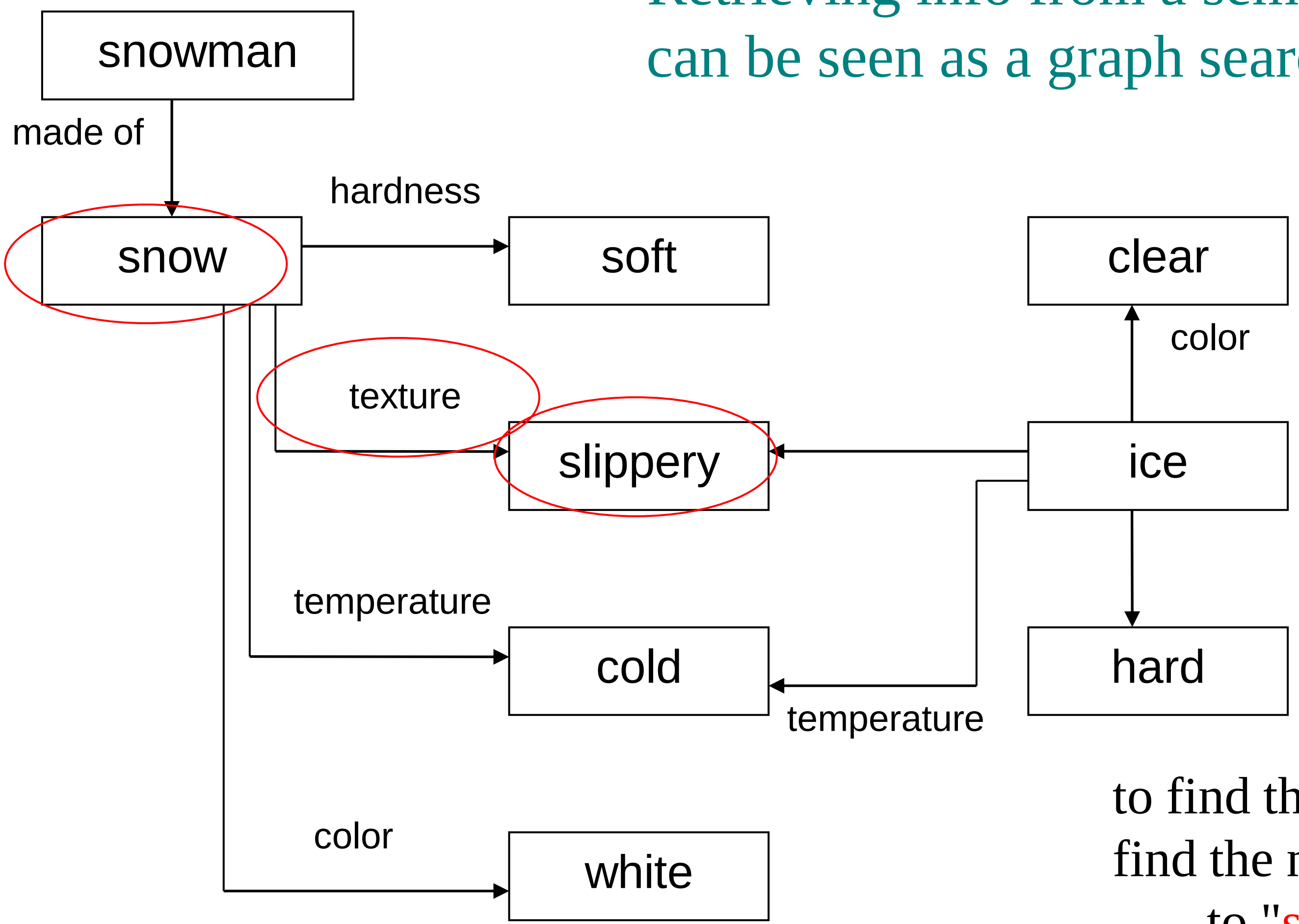


Taxonomy of Animal Knowledge As Semantic Network

What kind of animals are both dogs, cats?



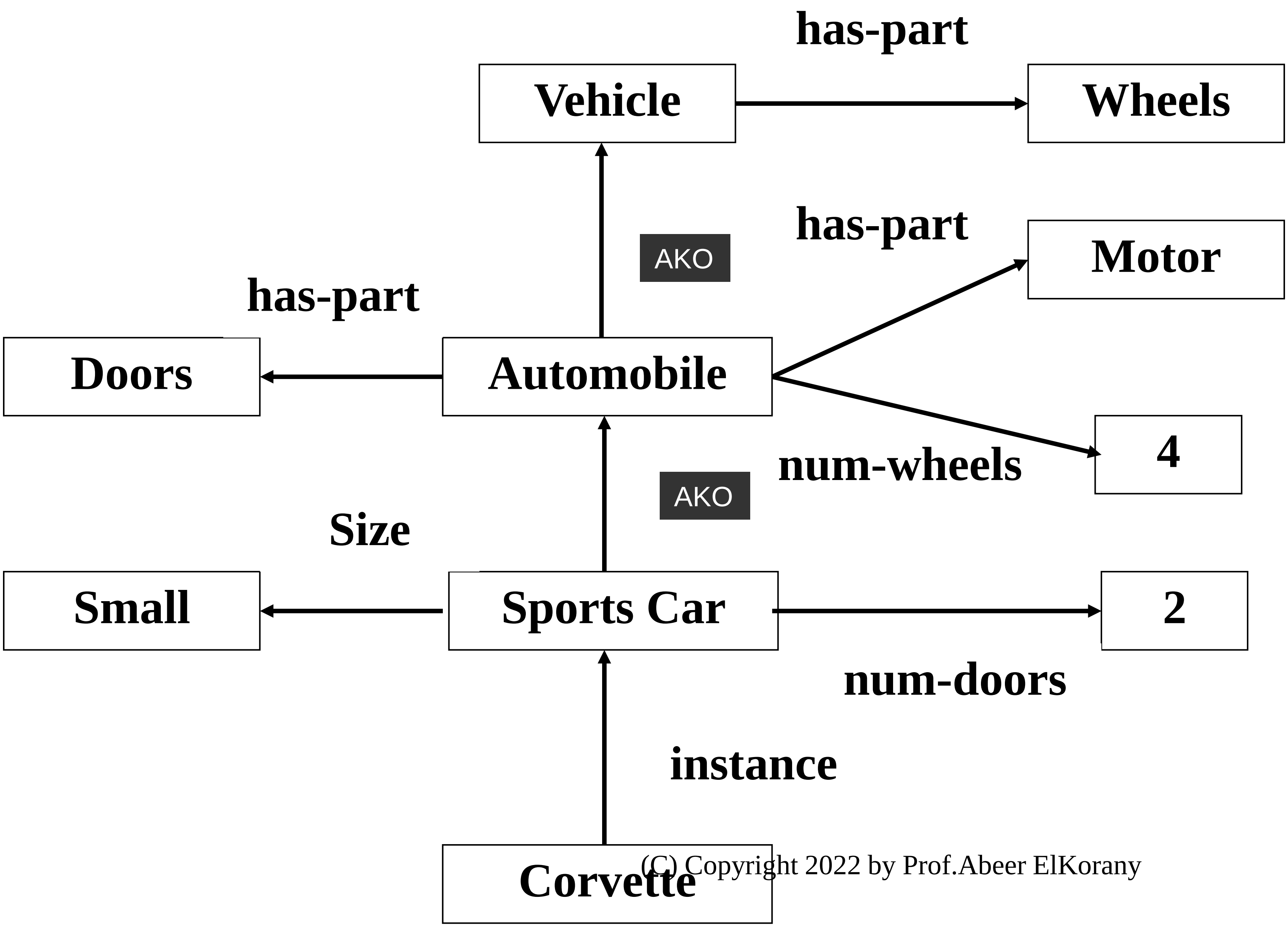
Network Representation of properties of snow and ice



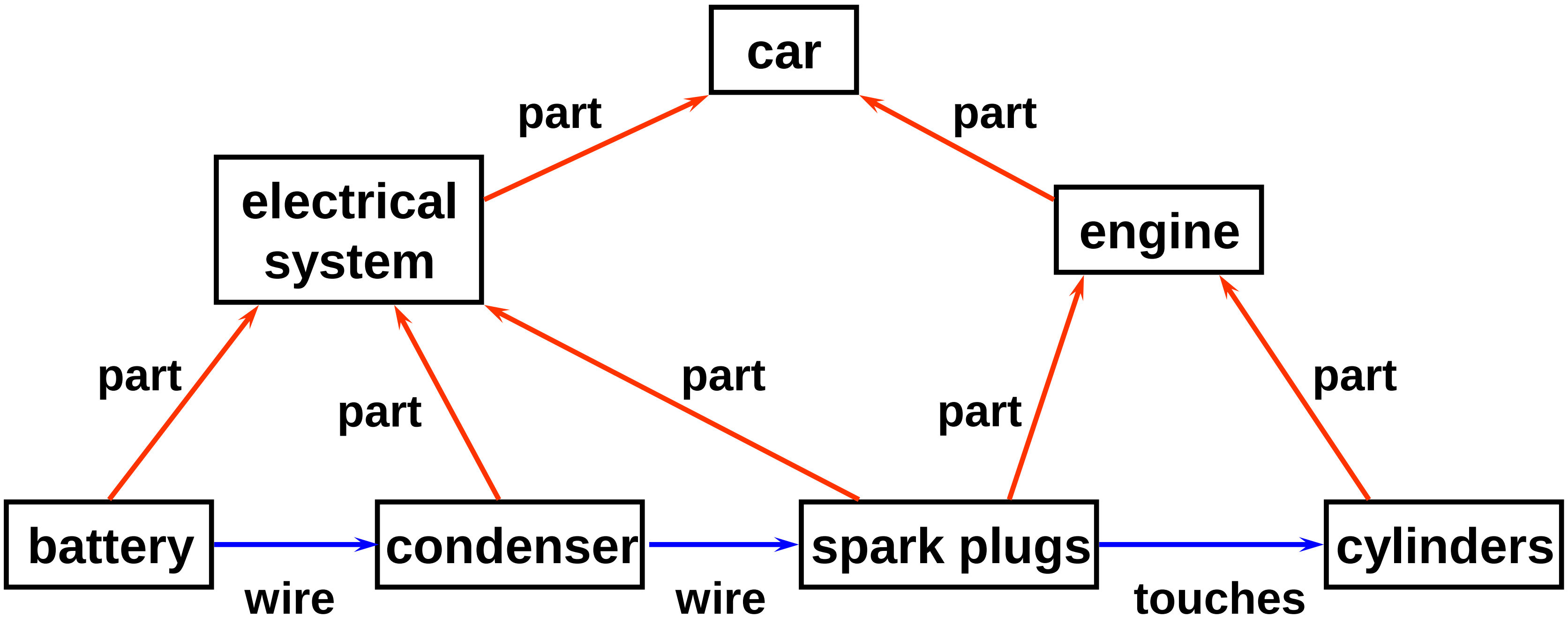
Retrieving info from a semantic net can be seen as a graph search problem

to find the texture of snow
find the node corresponding to "snow"
find the arc labeled "texture"
follow the arc to the concept "slippery"

Example of Semantic Net



A Physical/Causal Model: As Semantic Network



Advantages:

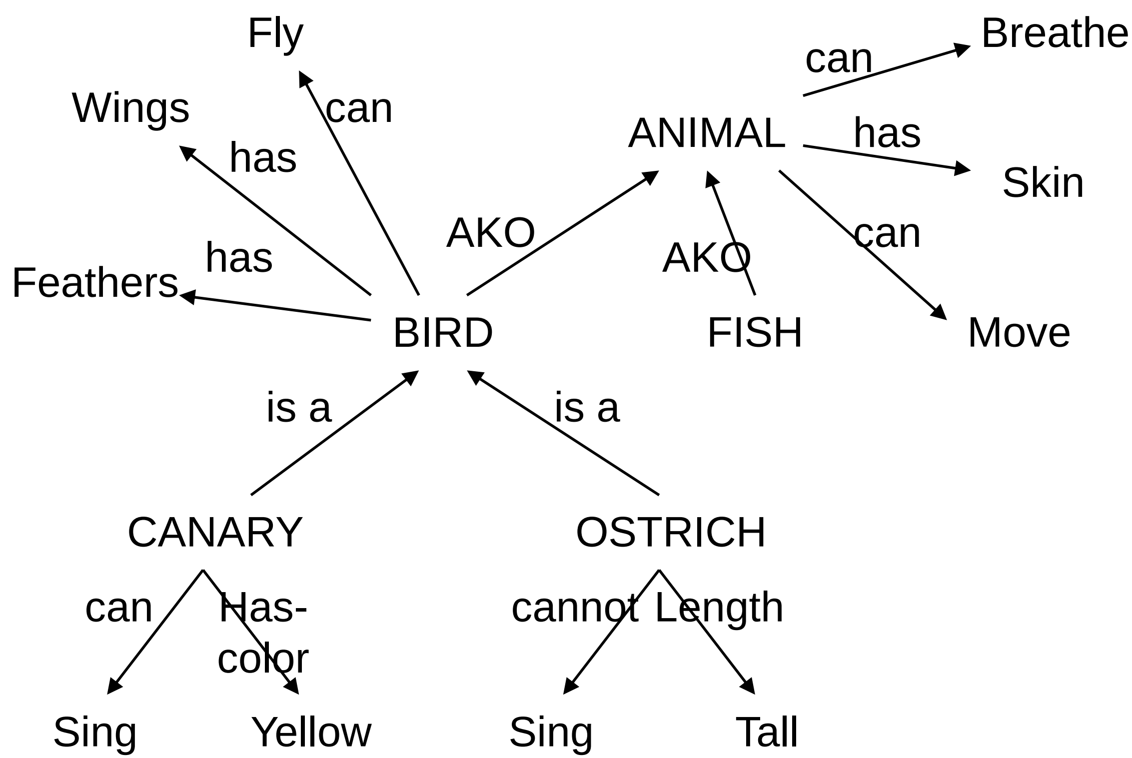
- allows reasoning about electrical connectivity
- allows reasoning about parts

Inference in a Semantic Net

- Basic inference mechanism: *follow links between nodes*. Two methods to do this:
- **Intersection search**
  - One of the earliest ways that semantic networks were used was to find relationships between objects by spreading ***activation*** from each of two nodes and seeing where the activations met. This process is called ***intersection search***.
- **Inheritance**
  - The AKO, *isa* and *instance* representation provide a mechanism to implement this.

Semantic nets & inheritance

- Nodes can inherit properties from other nodes (ISA,AKO)
- Usually only particular properties can be inherited over particular arcs



Semantic nets & inheritance (cont.)

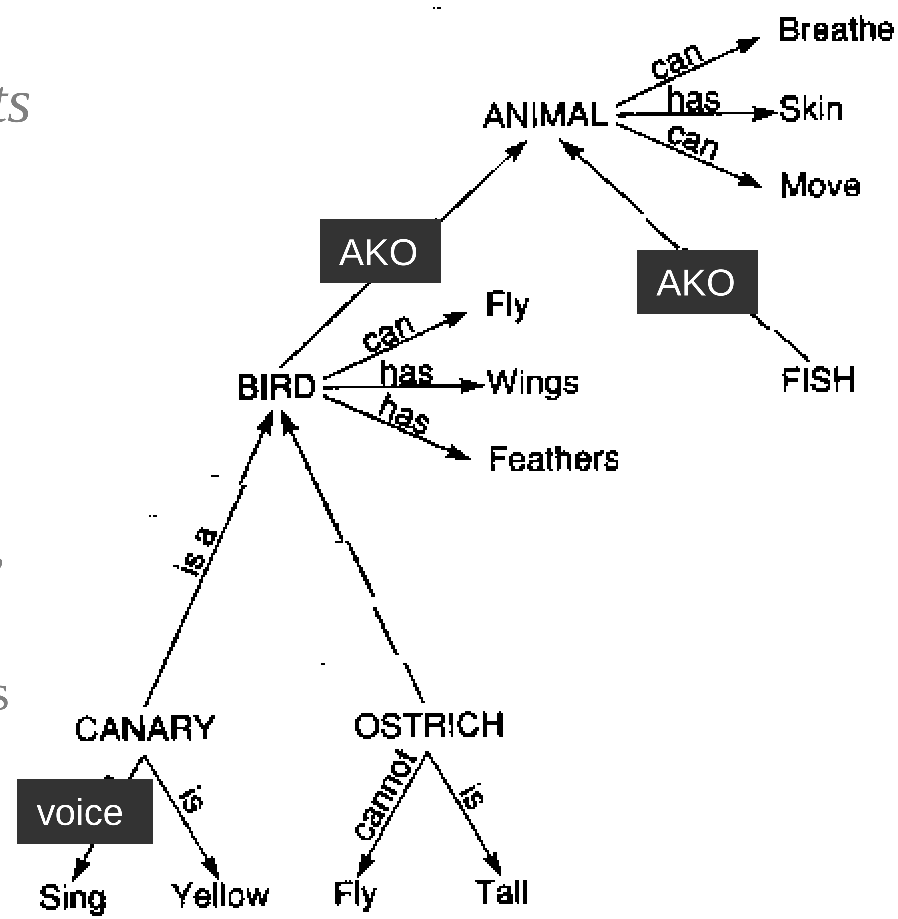
- In addition to data retrieval, semantic nets can provide for *deduction* using inheritance

since a canary is a bird, it *inherits* the properties of birds (likewise, animals)

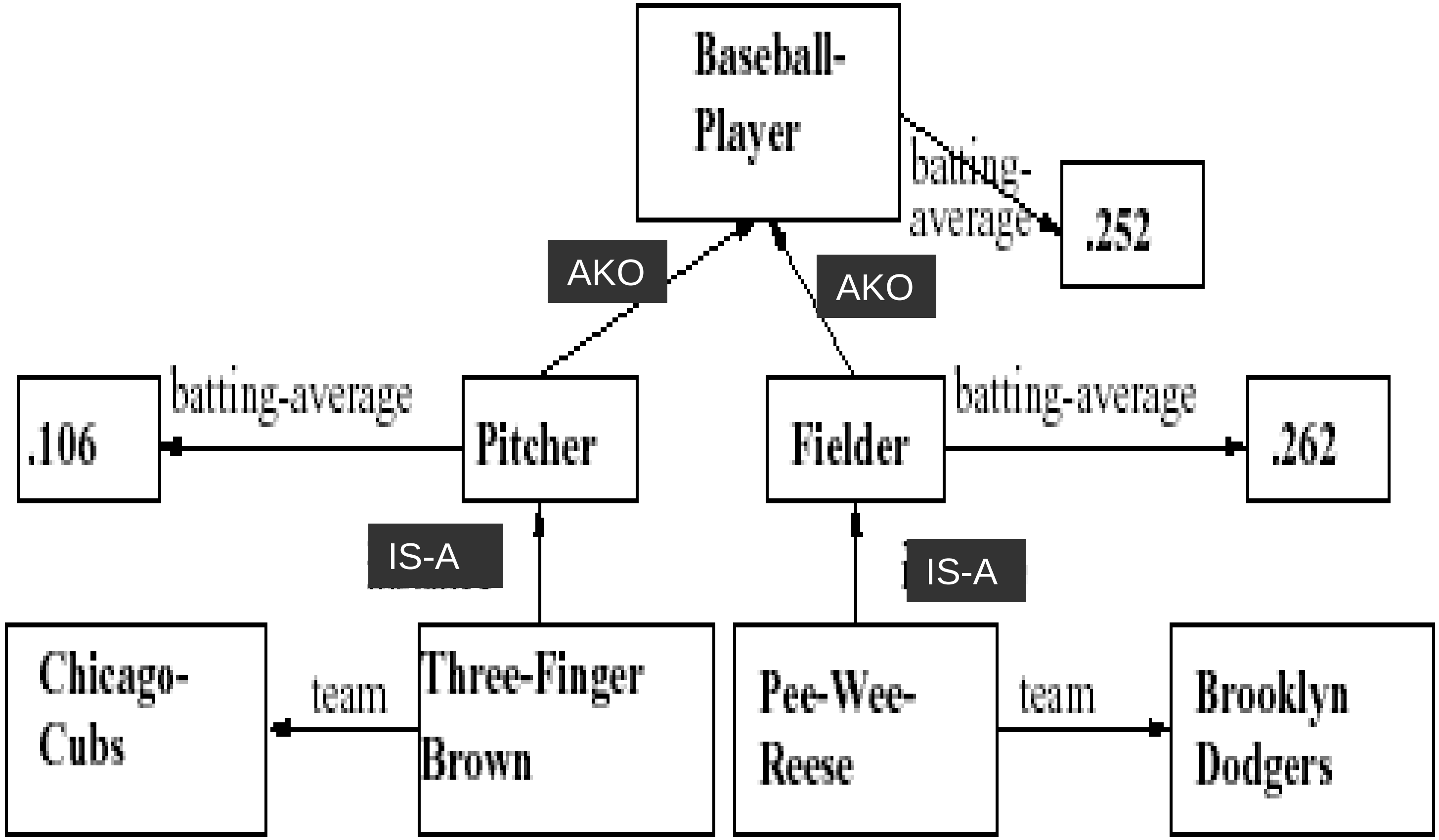
e.g., canary can fly, has skin, …

to determine if an object has a property,

- look for the labeled association,
- Follow *is_a* link to parent class and (recursively) look there



Question: “What is the relation between Chicago cubs and Brooklyn Dodgers?”



Answer: “They are both teams for baseball players.”

Inference in a Semantic Net

- By traversing network we can find:
  - That certain concepts related in certain ways (e.g., apples and elephants).
  - A concept has properties
- BUT: Meaning of semantic networks was not always well defined.
  - Are all Elephants big, or just typical elephants?
  - Do all Elephants live in Africa?
  - Do all animals have the same head?
- For machine processing these things must be defined. (Use of Frames)

Advantages and Disadvantage of Semantic nets

Advantage

- Easy to visualize
- Formal definitions of semantic networks have been developed.
- Related knowledge is easily clustered.
- Efficient in space requirements
  - Objects represented only once

Disadvantage

- Inheritance (particularly from multiple sources and when exceptions in inheritance are wanted) can cause problems.
- No standards about node and arc values

Object-Oriented representation

Objects: a natural way to organize the knowledge about:

- **Physical objects:**
  - a desk has a surface-material, # of drawers, width, length, height, color, procedure for unlocking, etc.
- **Situations:**
  - A class room: participants, teacher, day, time, seating arrangement,

lighting, procedures for registering, grading, etc.

- –A trip: destination, origin, conveyance, procedures for buying ticket, getting through customs, reserving hotel room, etc.

- **Important: Objects enable grouping of procedures for determining:**
  - properties of objects, their parts, interaction with parts

Frame Systems

- Frames were the next development, after semantic network
- A frame is viewed as a data structure for representing stereotyped situations to which are attached various kind of information including the object or events
- A frame system attempts to integrate
- Declarative notions about objects and events and their properties,
- Procedural notions about how to retrieve information and achieve goals.
    - frame = nodes and links grouped together
    - slot = link
    - slot values = destination of links

Frames

- Frames allow more convenient "packaging" of facts about an object.
- Frames look much like modern classes
- We use the terms "slots" and "slot values"
- Information in a frame:
  - Frame identification
  - Relationship to other frames
  - Descriptors of the requirements
  - Procedural information
  - Default information

mammal:
superclass: animal

elephant:
superclass : mammal
size: large
has-part: trunk

Nellie:
instance: elephant
likes: apples

Frames (cont.)

- Frames often allowed you to say which things were just typical of a class (default value), and which were definitional, so couldn't be overridden.

```
Elephant:
   superclass: mammal
   has-part: trunk
   color: grey
    size: large
```

- Frames often allowed slots to contain procedures.
- Area slot could contain code to calculate the area of polygon
- Frames also allow multiple inheritance

Slots

- Each slot contains one or more facets
- Facets may take the following forms:
  - values
  - default
    - used if there is not other value present
  - range
    - what kind of information can appear in the slot
  - if-added
    - procedural attachment which specifies an action to be taken when a value in the slot is added or modified (data-driven, event-driven or bottom-up reasoning)
  - if-needed
    - procedural attachment which triggers a procedure which goes out to get information which the slot doesn't have (expectation-driven; top-down reasoning)
  - other
    - may contain frames, rules, semantic networks, or other types of knowledge

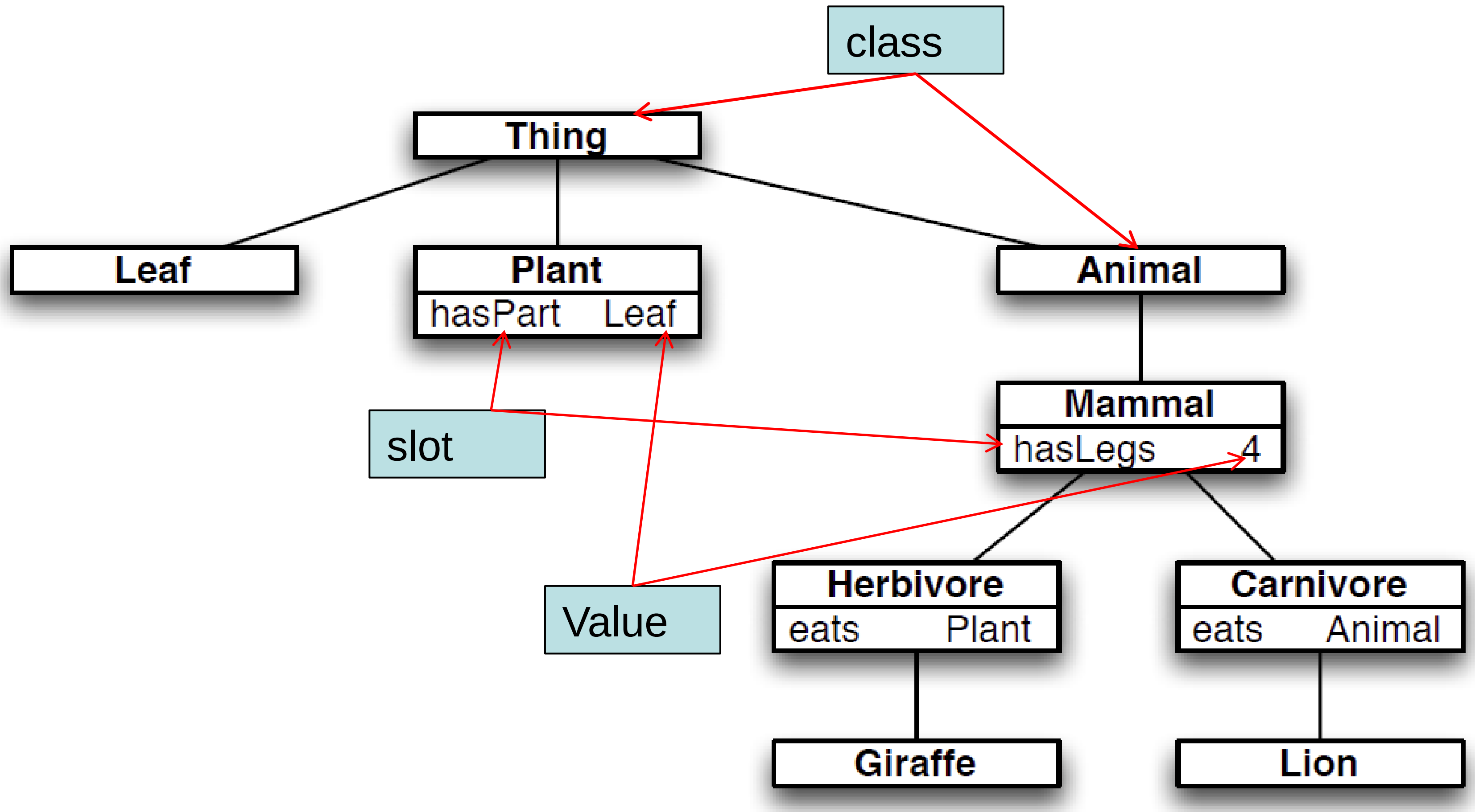
Usage of Slots

- Filling slots in frames
  - can inherit the value directly
  - can get a default value
  - these two are relatively inexpensive
  - can derive information through the attached procedures (or methods) that also take advantage of current context (slot-specific heuristics)
  - filling in slots also confirms that frame or script is appropriate for this particular situation

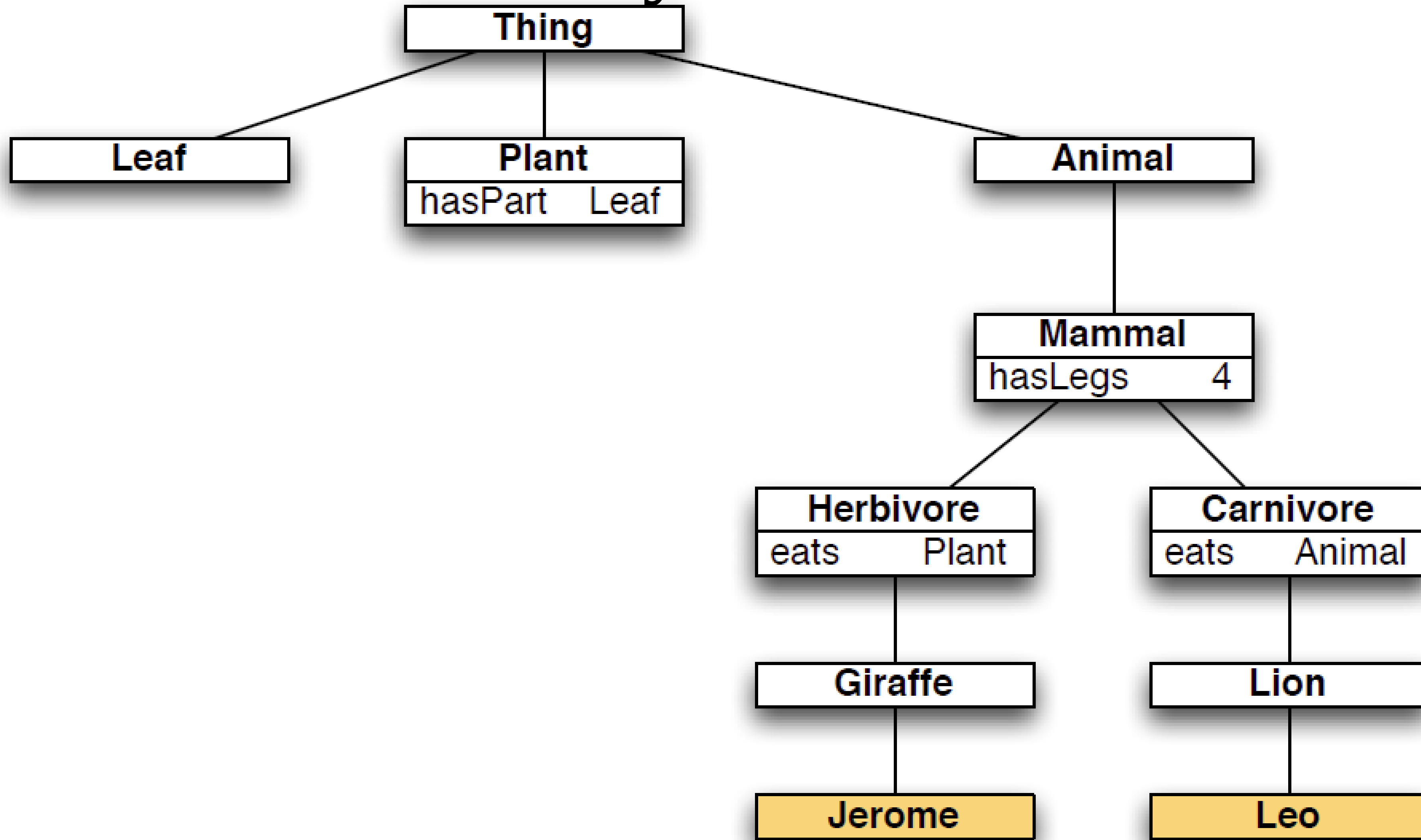
Hierarchy of Frames

- Frames are typically arranged in a hierarchy in which “lower” frames can inherit values from “higher” frames in the hierarchy.
- Properties and procedures for “higher” frames are more or less fixed whereas “lower” frames may be filled with more contingent information.

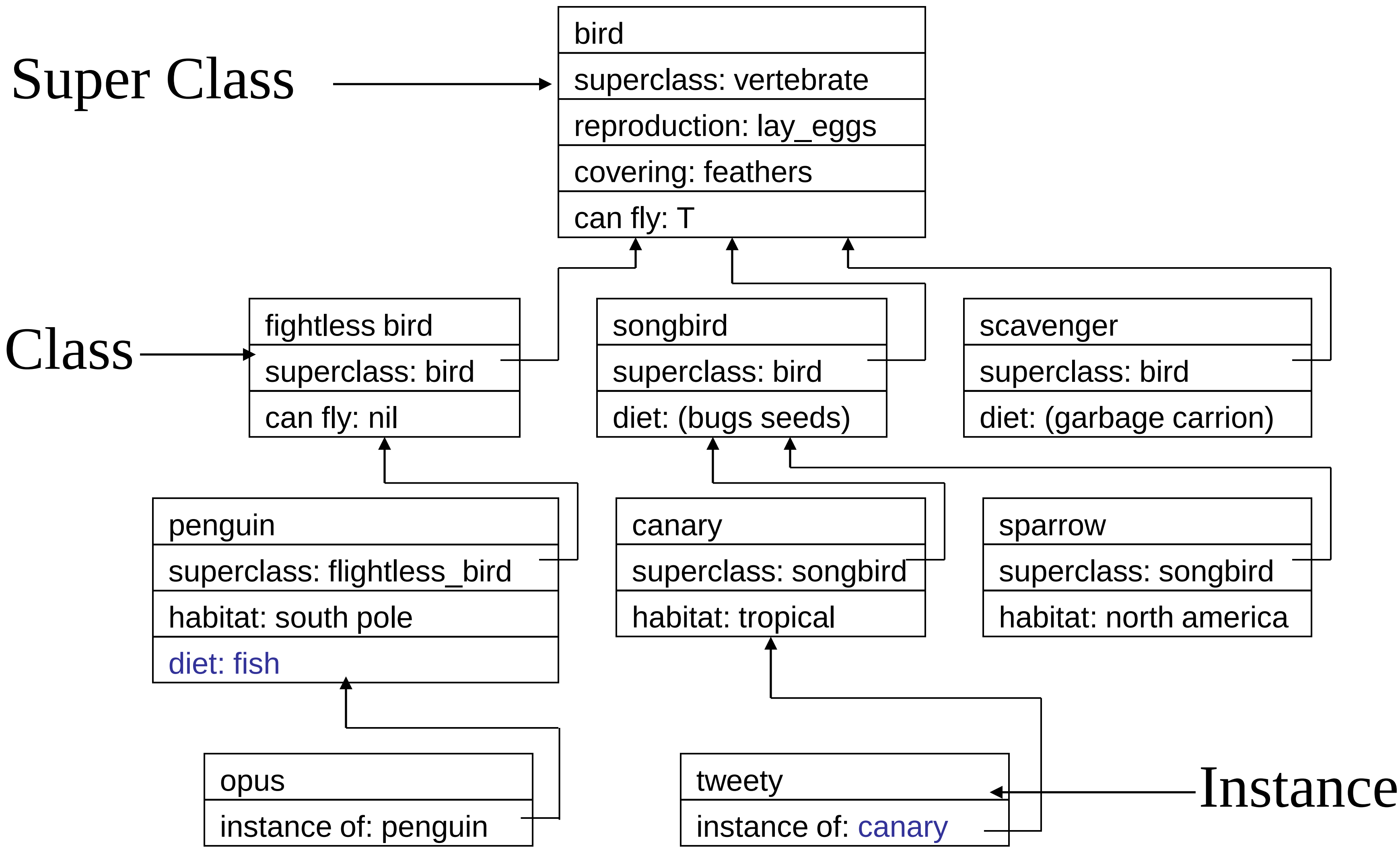
Frame Example



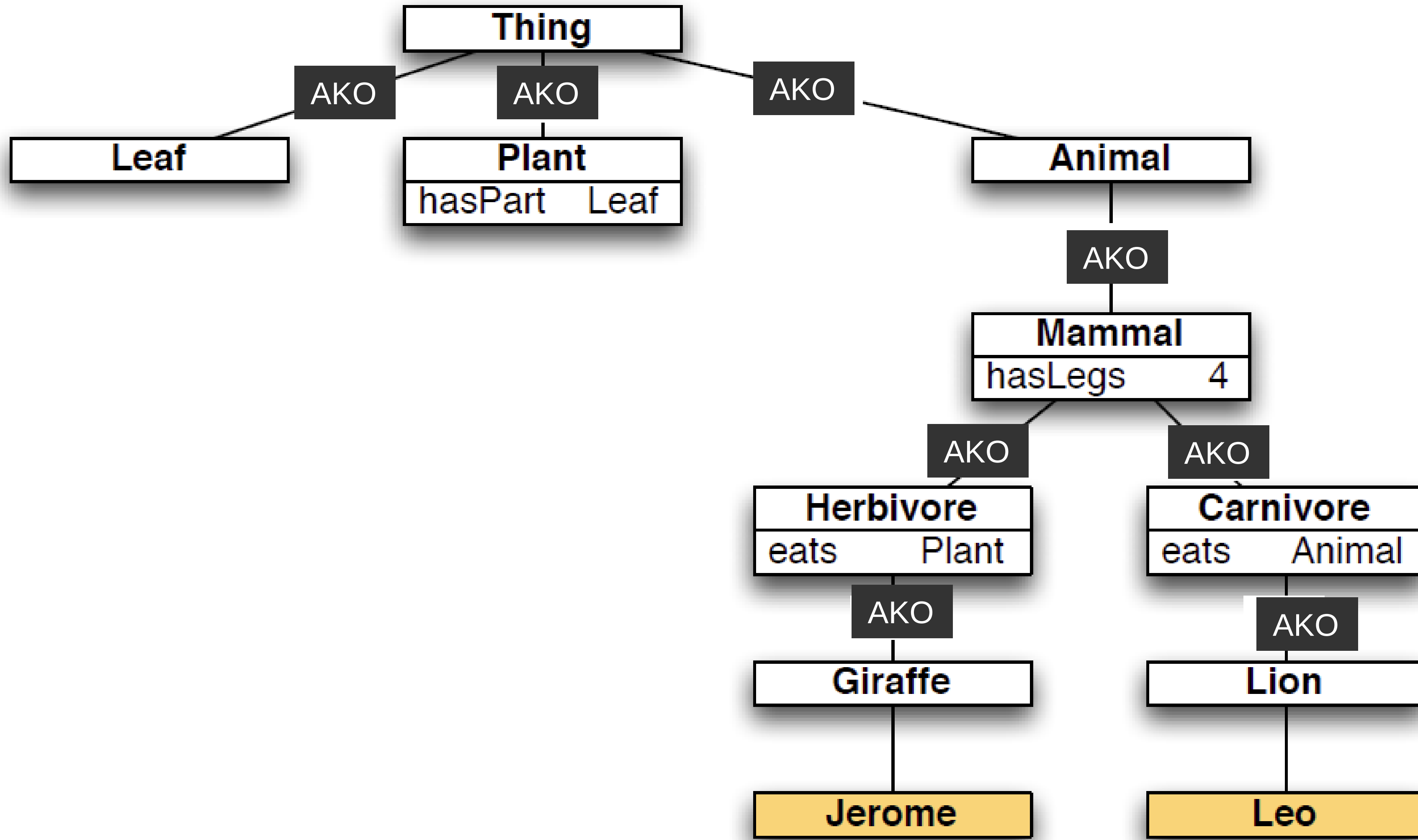
Inheritance System of animals



Inheritance System of Birds



Inheritance as Inference mechanism



Inheritance as Inference mechanism

Inheritance

How many legs does Jerome have? 4

⇒ Jerome is an instance of Giraffe

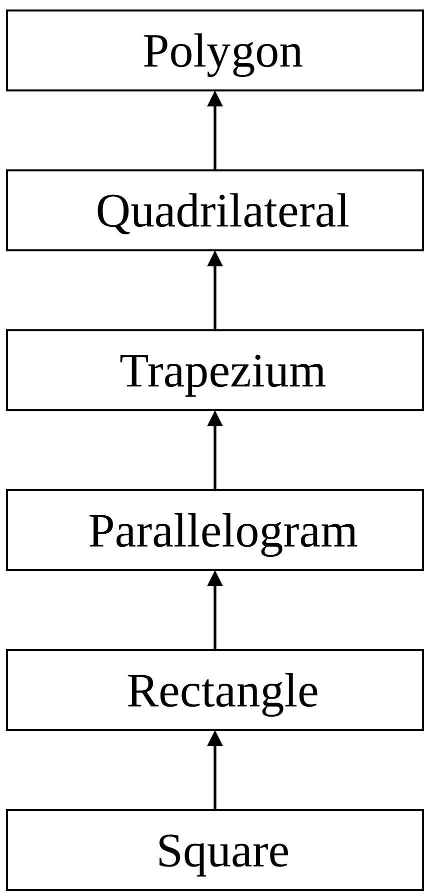
⇒ Every instance of Giraffe is an instance of Herbivore

⇒ Every instance of Herbivore is an instance of Mammal

⇒ Mammals have 4 legs

So the attribute of having 4 legs is inherited by Giraffe from Mammal

Hierarchical Representation for Geometric Figures of Land



- Each node in the hierarchy consists of a structure with the following format:

Name:

Number of Sides:

Length of Sides:

Size of Angles:

Area:

Price of Land:

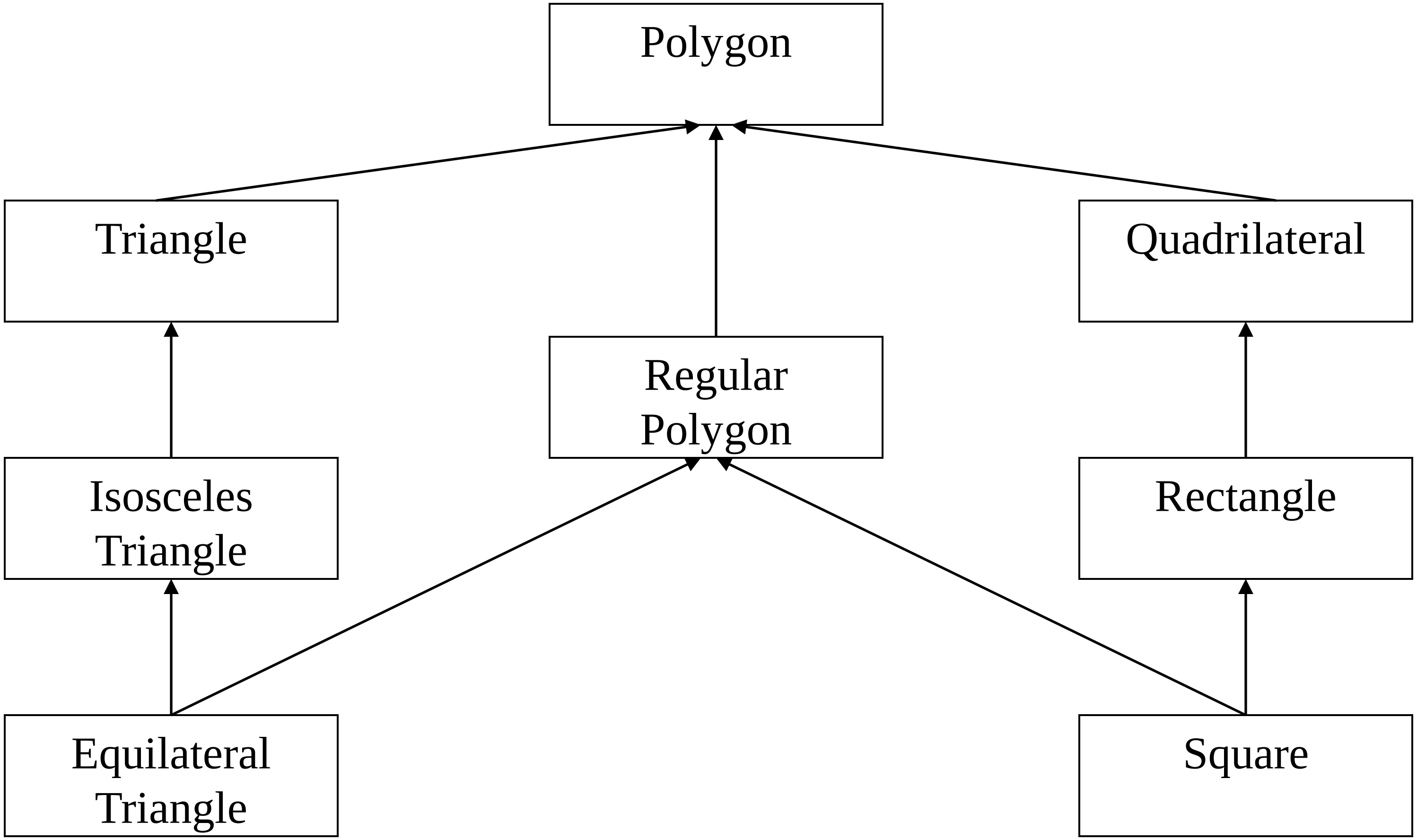
Some Polygon's slots have the following fillers:

Number of Sides: 4 (default)

Area: compute_area

Price= compute_price(Area)

Multiple Inheritance

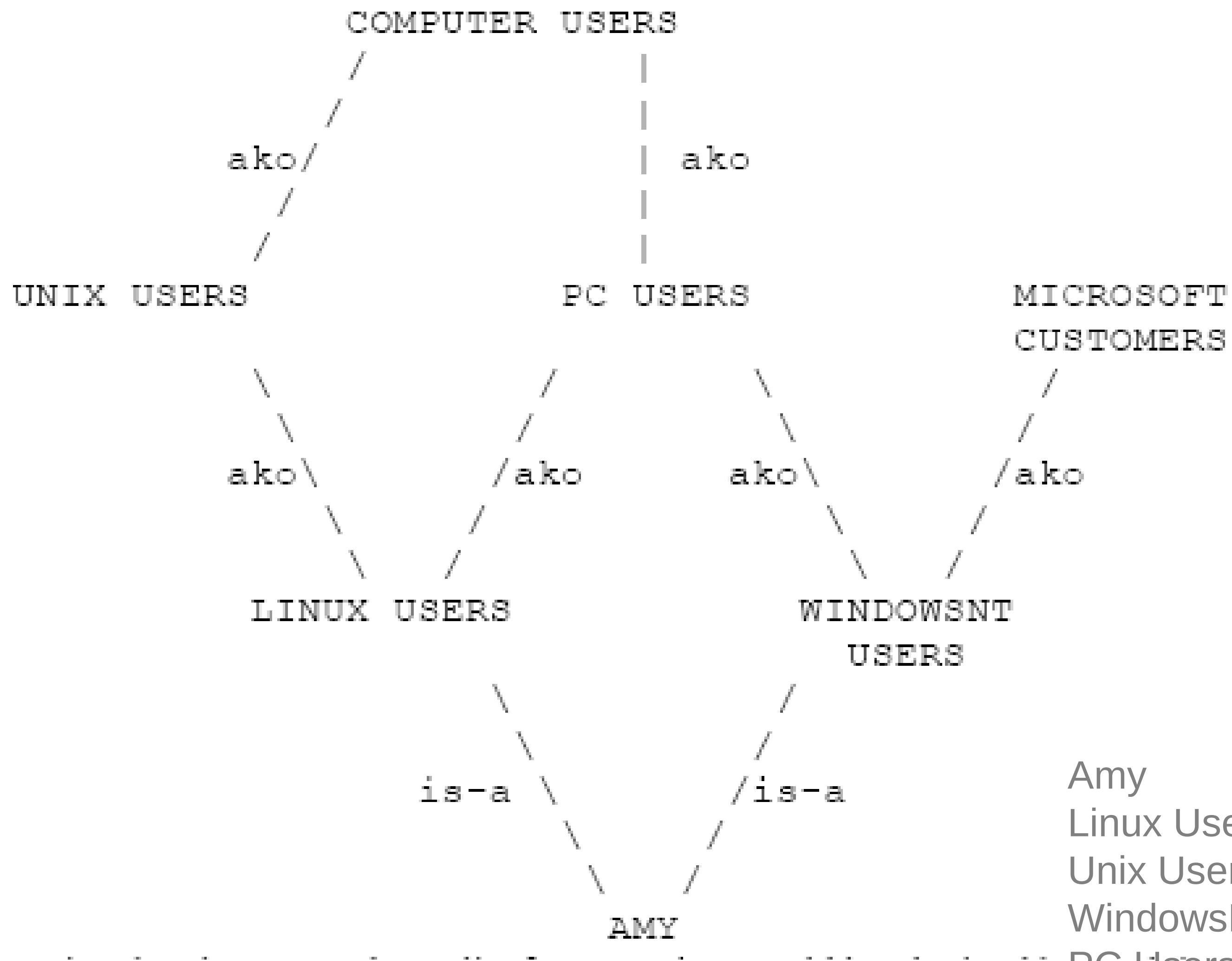


There is no ambiguity if “Square” inherits different information from “Regular Polygon” and “Rectangle”. The same applies to “Equilateral Triangle”

Frame Multiple inheritance

- If two or more ancestors have defined the same slot, it is crucial to know which method any instance of those ancestors inherit.
- Inheritance procedure (topological-sorting algorithm)
- Define <u>class precedence list.</u>
  - Get the direct parent of a class in left to right order
  - Compute the partial ordering of all the ancestors in the inherited hierarchy
  - Derive the total ordering of the <u>class precedence list</u> through a topological sort such that
    - **each class should appear on class-precedence list before any of its superclasses**
    - **Each class come first to those to its right**

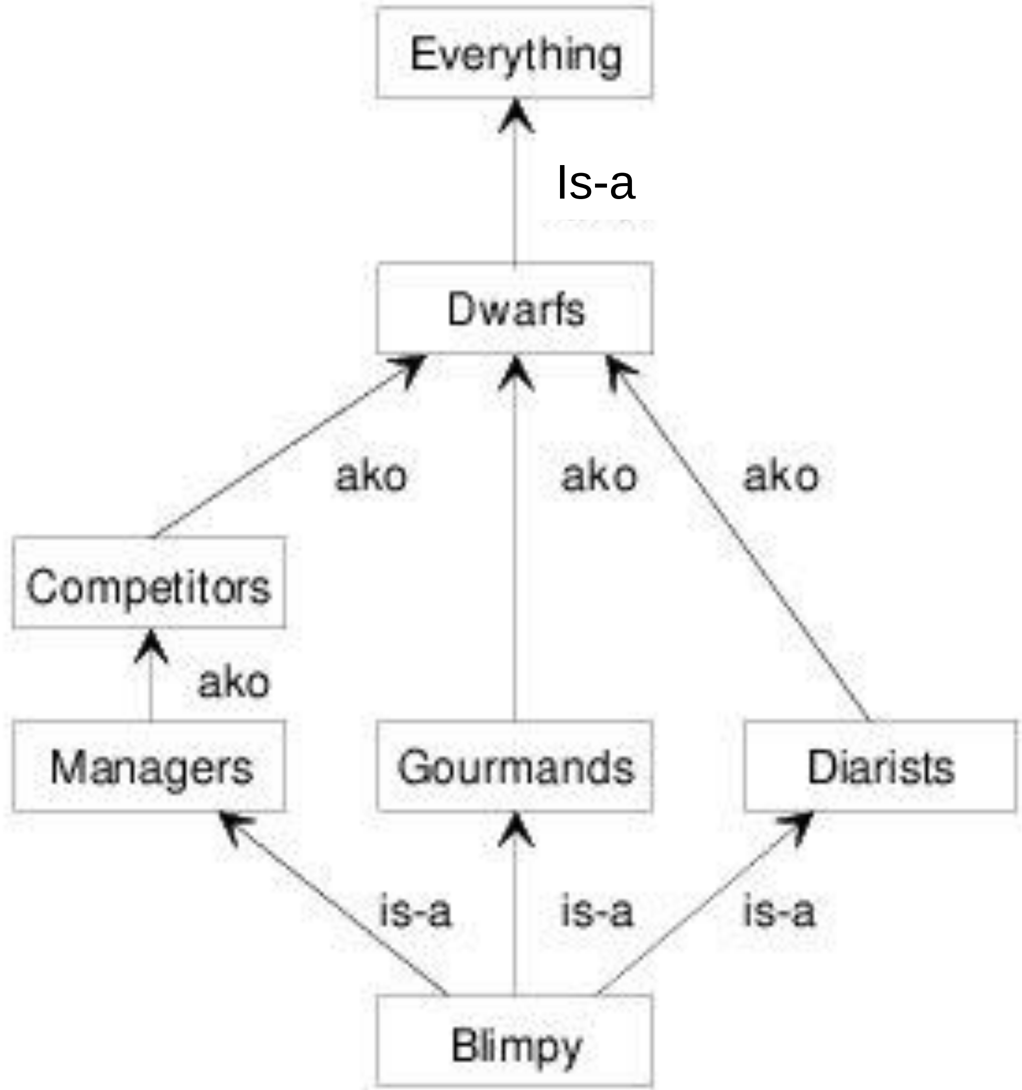
1- Give the class-precedence list for “AMY”



Amy
Linux Users
Unix Users
WindowsNT Users
PC Users
Computer Users
Microsoft Customers

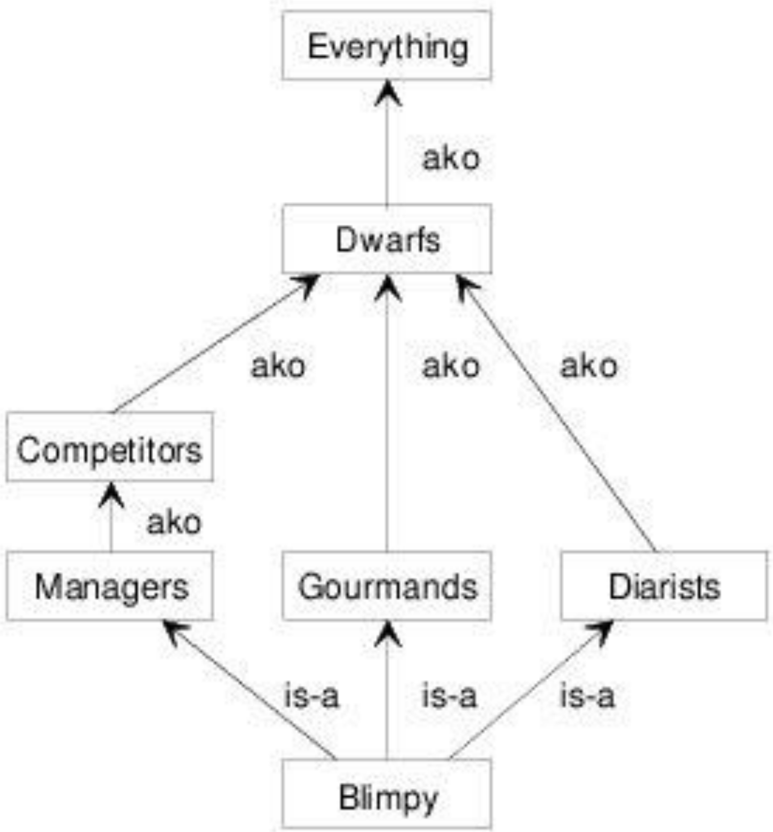
Branching Hierarchy

- Consider the value of appetite attribute for
  - Dwarfs: Small
  - Gourmand: Huge
- Constructing a class precedence list is not as simple
- DFS provides a search procedure for this
  - Each node appears once
- Left-to-right search satisfies convention



Exampls

- Inheritance procedure for Blimpy produces the following class precedence list
  - Blimpy
  - Managers
  - Competitors
  - Gourmands ← all food
  - Diarists
  - Dwarfs ← banana
  - Everything



What blimpy like to eat ->all food

- Represent the following into semantic netwrok
- 1. monkeys climb trees
- 2. monkeys eat bananas
- 3. programs manipulate bits
- 4. programs search trees
- 5. termite climb trees
- 6. termites eat trees

Complete the following semantic net:

Tom is a cat.
Tom caught a bird.
Tom is owned by John.
Cats like rats.
The cat sat on the mat.
A cat is a mammal.
A bird is an animal.
All mammals are animals.
Mammals have fur.

Complete the following semantic net .