Knowledge Representation

Because the world is more complicated than a 3x3 Array...

Need for Knowledge Representation

- Situations for AI which are more complex than puzzles and games
  - States of Human Affairs
  - Representation of Complex Rules
  - Understanding of Language
  - Representing Mechanisms for Design and Diagnosis
  - Bureacratic Rules

Knowledge Representation

- For Real World Problems:
  - Lots of Knowledge
  - Complex Interactions
  - Partial State Information
  - Ambiguity
  - Un-knowable
  - Dynamically changing

Evaluating KR Schemes

- how can we get that knowledge into machines?
- Is knowledge domain-specific or general?
- How can we get at it?
- How flexible is it?
- How can we change it?

Representational Adequacy

- How can facts about the world be represented in data-structures?
- Is our representation adequate to the domain?

Representational Adequacy

- Under the "law of representation" we know that if you need to represent 2^k things, you need k bits
  - –How many "ideas" are there?
- If representation is too weak, you cannot get at necessary discriminations e.g.
  - –Representing visitors to a website by the IP address, misses different users in cybercafe or berry patch
  - –Representing air-flight without plane-type means you can't do seat assignments

Knowledge Accessibility

- How can the represented knowledge be indexed and used?
- How can it be manipulated and processed?
- How can it be incremented?

Inferential Ability

- How can inference be accomplished?
- Will it be efficient?
- How will inference be *controlled*?

It is very easy to create infinite variations of knowledge from a finite set of statements, using rules and laws of logic

e.g. how to avoid combinatorial explosion?

KR Methods

- Procedural Representations
  - –e.g. Computer Programs
- Feature-Value Systems
- Database Systems (indexed Tables)
- First Order Predicate Logic
- Semantic Networks

Example: Computer Programs A Procedural Representation

- Adequate?

  Yes, Turing Universal
- Accessible?

  No, compiled and distributed
- Translation?

  yes, but need BS in CS
- Inference?

  Not usually, Needs Upgrade

Feature Systems

- Each object is represented by a vector
- Each position in the vector represents a feature
- Each number in a position represents the objects value of that feature

Color	Size	Shape	Weight
1=red	0 small, 1 big	3 triangle, 4 square	in oz.

– Organizes data well, but allows no compositionality or flexibility as the situation changes.

Database Systems

- Scalable
- Transaction Based
- Arbitrary complexity of representation, but some Tables get large
- Change of representation tables can require "Database Architect" and downtime.

flight	from	to	time
6550	chicago	ny	4.45
231	ny	chicago	9:45
101	ny	tokyo	12:00

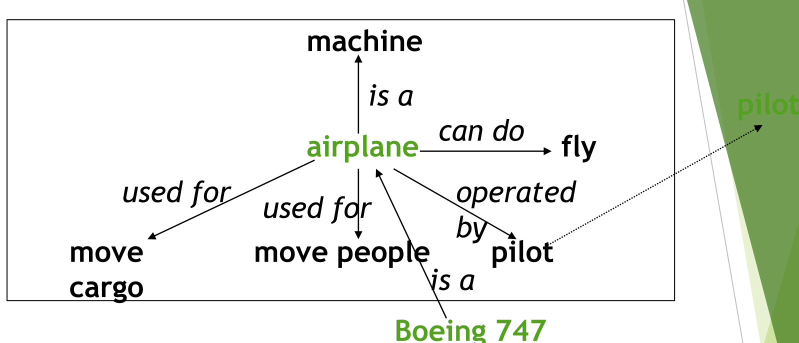
Semantic Networks

- A semantic (or associative) network is a simple representation scheme which uses a graph of *labeled* nodes and *labeled, directed* arcs to encode knowledge.
  - Labeled nodes: objects/classes/concepts.
  - Labeled links: relations/associations between nodes
  - Labels define the semantics of nodes and links
  - Large # of node labels (there are many distinct objects/classes)

    Small # of link labels (types of associations can be merged into a few)

    e.g., buy, sale, give, steal, confiscation, etc., can all be represented as a single relation of "*transfer ownership*" between recipient and donor
  - Usually used to represent static, taxonomic, concept dictionaries
- Semantic networks are typically used with a special set of accessing procedures which perform "reasoning"
  - e.g., inheritance of values and relationships
- often much less expressive than other KR formalisms

13



- Nodes for words
- Directed links for relations/associations between words
- Each link has its own meaning
- You know the meaning (semantics) of a word if you know the meaning of all nodes that are used to define the word and the meaning of the links connecting them
- Otherwise, follow the links to the definitions of related words

14

Semantic Networks

- First introduced by Quillian back in the late-60s

  M. Ross Quillian. "Semantic Memories", In M. M. Minsky, editor, *Semantic Information Processing*, pages 216-270. Cambridge, MA: MIT Press, 1968

- **Semantic network** is simple representation scheme which uses a graph of labeled nodes and labeled directed arcs to encode knowledge
  - Nodes – objects, concepts, events
  - Arcs – relationships between nodes
- **Graphical depiction** associated with semantic networks is a big reason for their popularity

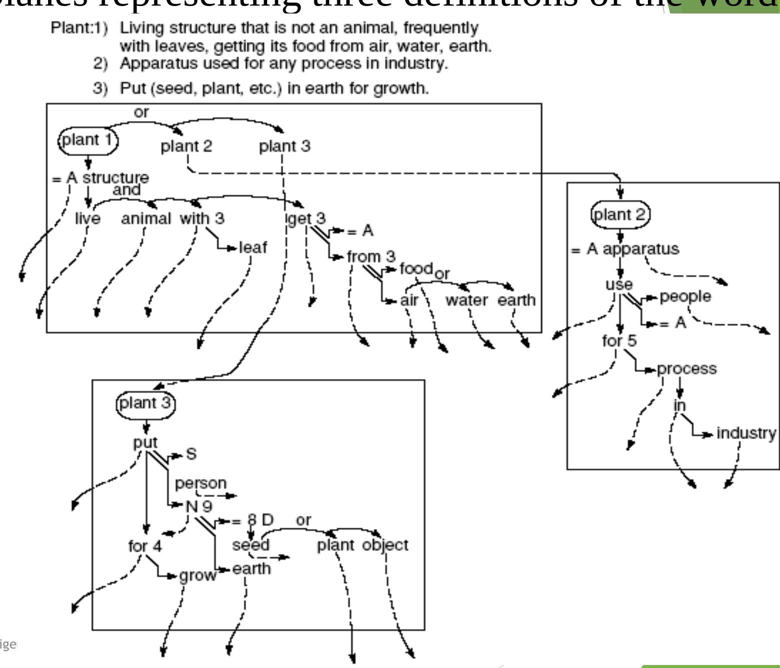
M Ross Quillian
Invented Semantic Nets in 1968

- Used a labelled graph to represent meanings of words in an electronic dictionary
- Technology was intensely developed throughout the 1970's
  - Woods, Brachman, Shapiro, Fahlman

Semantic Network in Natural Language Understanding

- First implementation of semantic networks in machine translation
- Quillian's semantic network
  - Influential program
  - Define English words in a dictionary-like, but no basic axioms
  - Each definition leads to other definitions in an unstructured and sometimes circular fashion
  - When look up a word, traverse the network

Three planes representing three definitions of the word "plant"



Elements of Semantic Networks

- Nodes
  - Labeled circles
  - Represent Concepts

Types (classes)

Tokens (individuals)

Elements of Semantic Networks

- Links
  - Labeled Arrows
  - Represent Relations Between Concepts
  - Only a finite set of link types are allowed

Standard Links

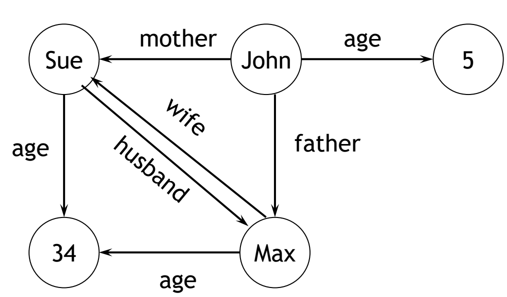
IS-A

has-part

- Different Link types are set up for knowledge in specific domains

Nodes and Arcs

- Arcs define binary relations which hold between objects denoted by the nodes.



FOPL Equiv

mother (john, sue)
age (john, 5)
wife (sue, max)
age (max, 34)
...

Advantages of Semantic nets

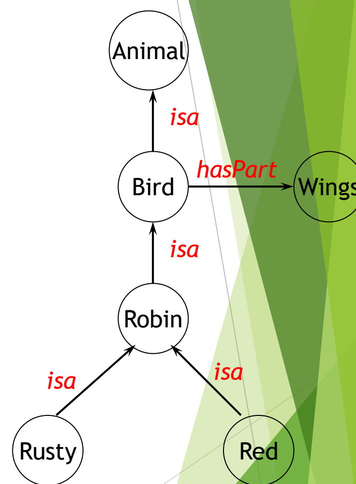
- Easy to visualize
- Formal definitions of semantic networks have been developed.
- Related knowledge is easily clustered.
- Efficient in space requirements
  - Objects represented only once
  - Relationships handled by pointers

Disadvantages of Semantic nets

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

Inheritance

- Inheritance is one of the main kind of reasoning done in semantic nets
- The **ISA** (is a) relation is often used to link a class and its superclass.
- Some links (e.g. **haspart**) are inherited along **ISA** paths
- The semantics of a semantic net can be relatively informal or very formal
  - Often defined at the implementation level

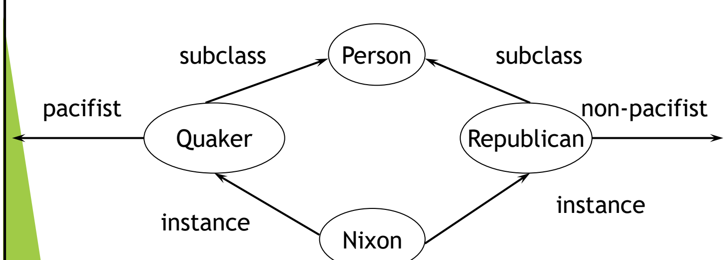


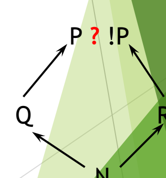
Multiple Inheritance

- A node can have any number of superclasses that contain it, enabling a node to inherit properties from multiple *parent* nodes and their ancestors in the network. It can cause conflicting inheritance.

Nixon Diamond

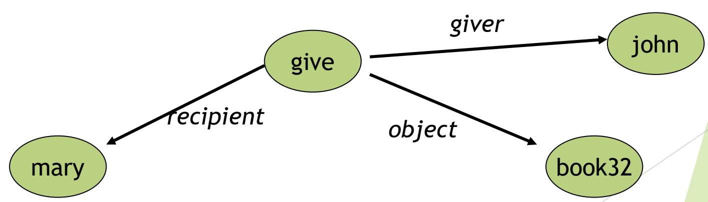
(two contradictory inferences from the same data)





Non-Binary Relationships

- Non-binary relationships can be represented by "turning the relationship into an object"
- We might want to represent the generic "give" event as a relation involving three things: a giver, a recipient and an object, give(john, mary, book32)



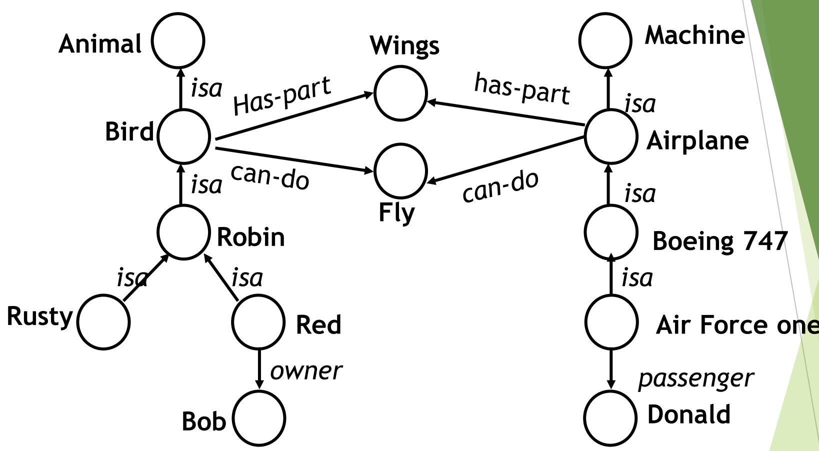
27

Graph Algorithms

- Marker passing
  - Each node has an unique marker
  - When a node is activated (from outside), it sends copies of its marker to all of its neighbors (following its outgoing links)
  - Any nodes receiving a marker sends copies of that marker to its neighbors
  - If two different markers arrive at the same node, then a connection is found between originating nodes
- Spreading activation
  - Instead of passing labeled markers, a node sends labeled activations (a numerical value), divided among its neighbors by some weighting scheme
  - A node usually consumes some amount of activation it receives before passing it to others
  - The amount of activation received by a node is a measure of the strength of its association with the originator of that activation

28

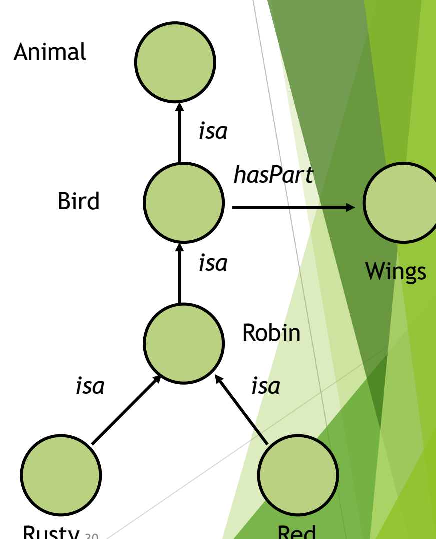
Inference by graph algorithm



- Red (a robin) is related to Air Force One by association (as directed path originated from these two nodes join at nodes Wings and Fly)
- Bob and Donald are not related (no paths originated from them join in this network

ISA hierarchy

- The ISA (is a) or AKO (a kind of) relation is often used to link a class and its superclass.
- And sometimes an instance and it's class.
- Some links (e.g. has-part) are inherited along ISA paths.
- The semantics of a semantic net can be relatively informal or very formal
  - often defined at the implementation level



30

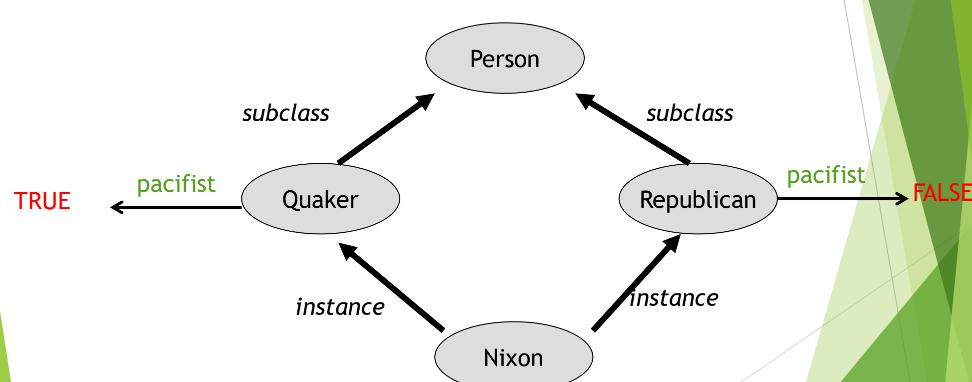
Inference by Inheritance

- One of the main types of reasoning done in a semantic net is the inheritance of values (properties) along the subclass and instance links.
- Semantic Networks differ in how they handle the case of inheriting multiple different values.
  - All possible properties are inherited
  - Only the value or values of the "lowest" ancestor are inherited

31

Multiple Inheritance: the Nixon Diamond

- This was the classic example of inheritance conflict

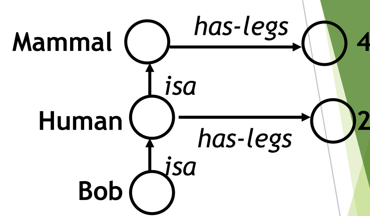


32

Exceptions in ISA hierarchy

- Properties of a class are often default in nature (there are exceptions to these associations for some subclasses/instances)

- Closer ancestors (more specific) overriding far way ones (more general)

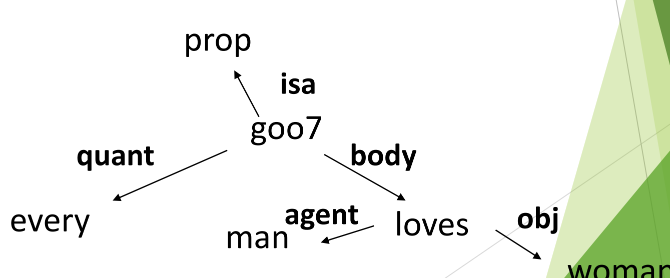


33

Representational Adequacy of SN's?

- As good as FOPL
- Can Graphical Representation limit explosion?

- Sometimes you cannot make any sense out of the network!



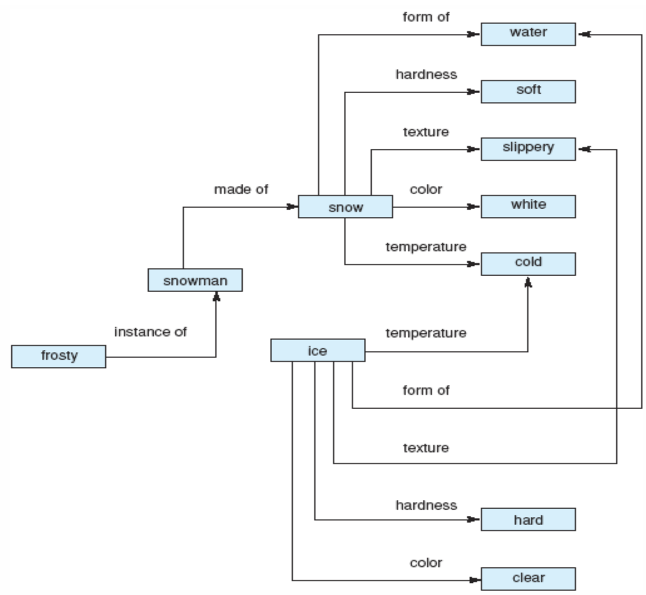
Inferences in Semantic Networks

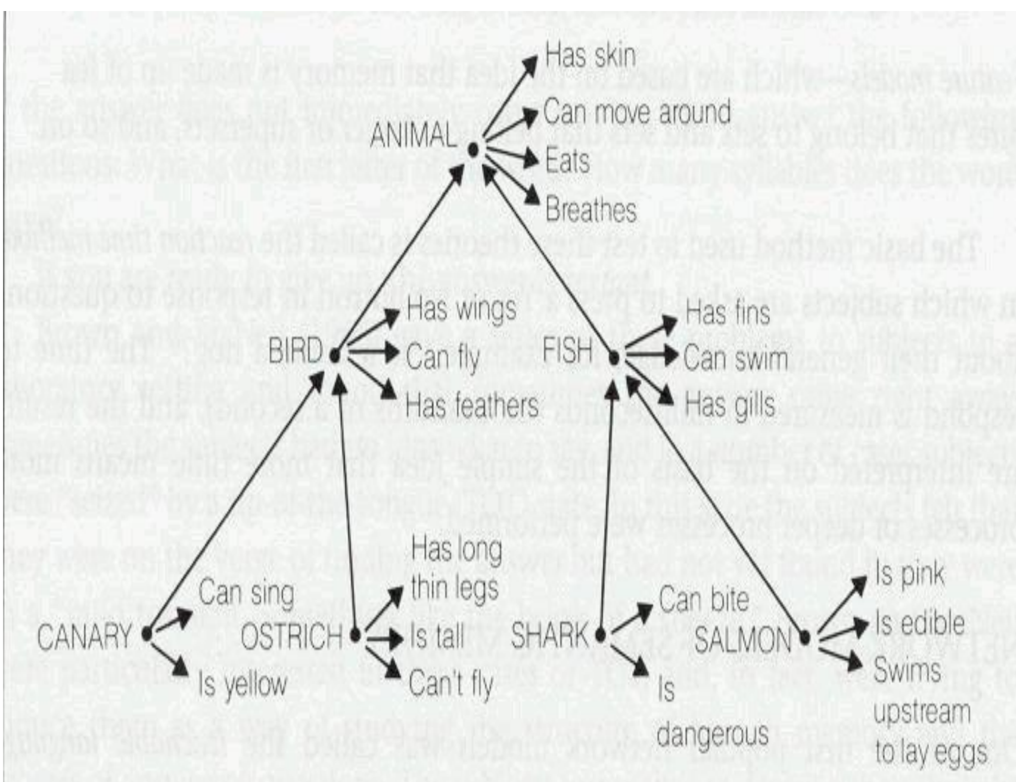
- Inference along associational links
- Find relationships between pairs of words
  - Search graphs outward from each word in a breath-first fashion
  - Search for a common concept or intersection node
  - The path between the two given words passing by this intersection node is the relationship being looked for

Analysis of Semantic Networks

- For a particular Domain, you
  - make up a set of link-types
  - Create a set of nodes
  - connect them together
- Ascribe Meaning
- Write Programs

A Semantic Network Representation of Properties of Snow and Ice



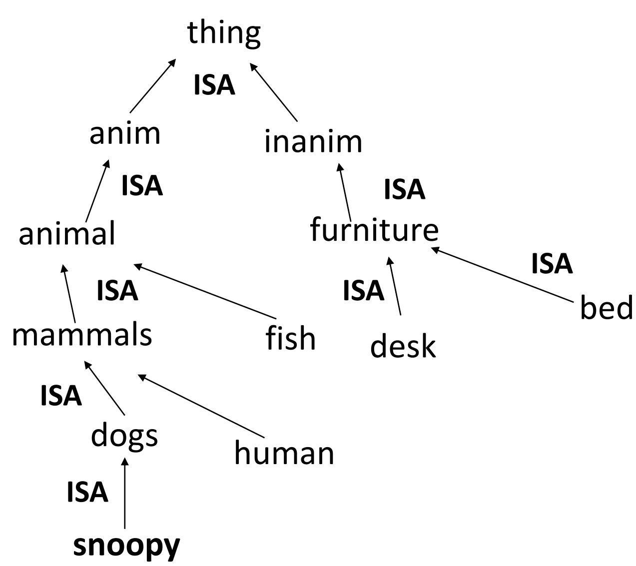


Basic Semantic Network

- ;makes an atomic value
- (defun attr (entity attribute value)
-   (setf (get entity attribute) value))

- ;makes a list value for multiple inheritance
- (defun isa (entity1 entity2)
-   (setf (get entity1 'isa)
-     (cons entity2 (get entity1 'isa))))

Example of Semantic Network



Setting up a hierarchy

```
(defun isa (entity1 entity2)
  (setf (get entity1 'isa)
    (cons entity2 (get entity1
'isa))))
```

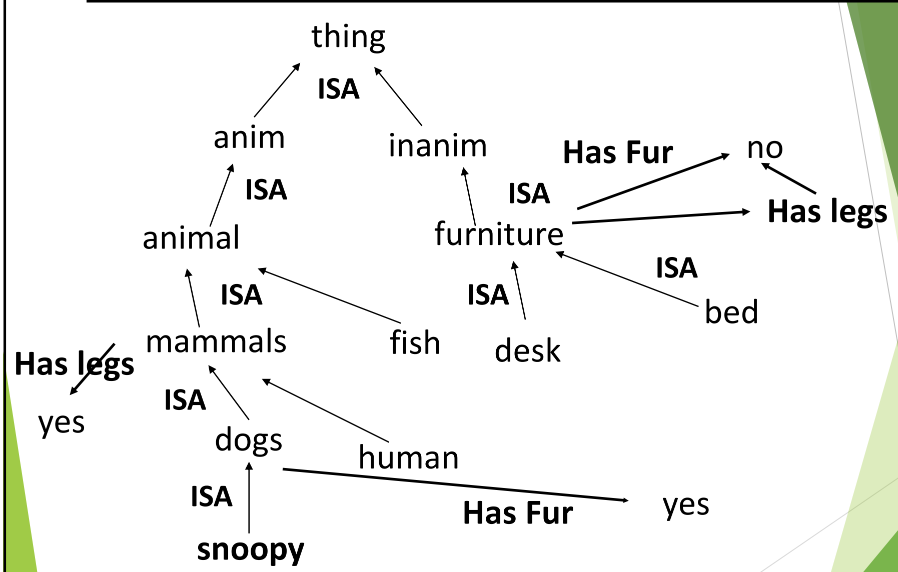
- (isa 'snoopy 'dog)
- (isa 'dog 'mammal)
- (isa 'human 'mammal)
- (isa 'mammal 'animal)
- (isa 'animal 'anim)
- (isa 'anim 'thing)
- (isa 'inanim 'thing)
- (isa 'furniture 'inanim)
- (isa 'desk 'furniture)
- (isa 'bed 'furniture)

```
(defun attr (entity attribute
value)
  (setf (get entity attribute)
value))
```

- (attr 'dog 'has-fur 'yes)
- (attr 'human 'has-skin 'yes)
- (attr 'mammal 'has-legs 'yes)

- (attr 'furniture 'has-legs 'no)
- (attr 'furniture 'has-fur 'no)

Example of Semantic Network



Get a local attribute

- The native "get" from property lists works ok

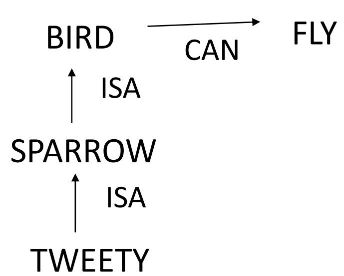
- (get 'snoopy 'isa)
- (DOG)

- (get snoopy 'has-legs)
- (nil)

- We use an inference called inheritance

Property Inheritance

A form of Default Reasoning



Therefore, Tweety can fly!!

Really Simple Isa inheritance

```
(defun get-attr (entity attribute)
  (or (get entity attribute)
    (loop for e in (get entity 'isa) thereis
        (get-attr e attribute))))
```

- Now, a feature attached higher in hierarchy can be retrieved from any descendent in ISA tree...

when inheritance fails

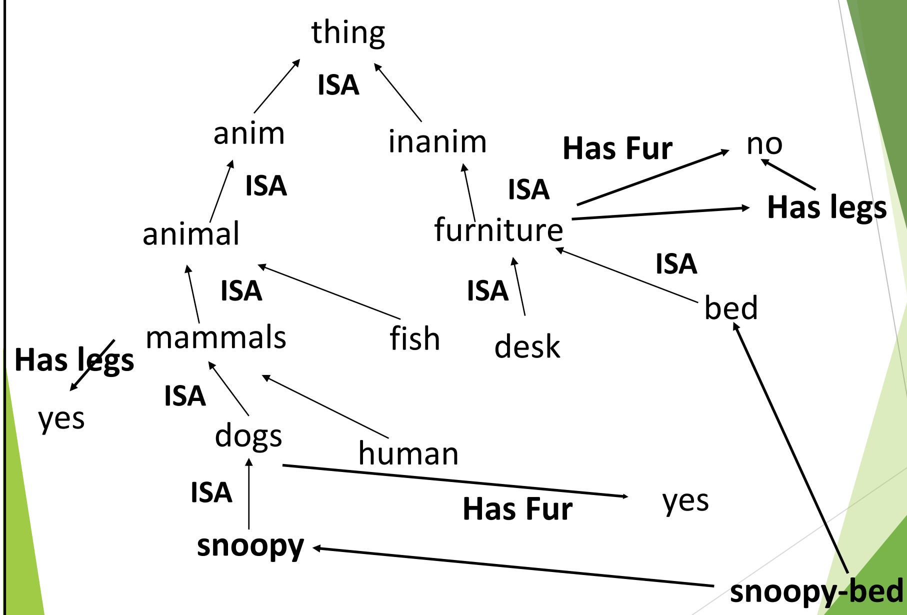
- What about Ostriches not flying?
- What about Snoopy Beds?
  - –Both animate and inanimate?
  - –Need Non-Monotonic Blocking form of multiple inheritance!
  - –Many Ph.D's theses were written about this.

"Elephants are grey",

"Clyde is an elephant"

"Clyde is pink".

Example of Semantic Network



ISA is more complicated to get right

What does it mean?

- Set inclusion?
- Generalization?

  f(x) -> f(y)
- Concept Containment?

  any feature of x is a feature of y
- Description?

  a trunk is a cylinder
- Instantiation?

  tweety is a bird