Knowledge Representation using structured objects

Introduction

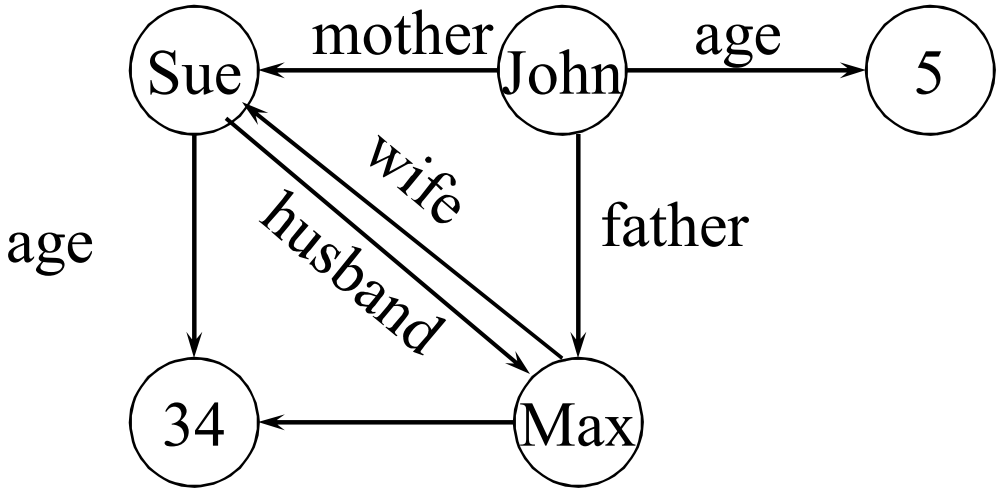
- "Traditional" knowledge representation is formal logic
- Structured objects are:
  - knowledge representation formalisms whose components are essentially similar to the nodes and arcs found in graphs.
  - in contrast to production rules and formal logic.
  - an attempt to incorporate certain desirable features of human memory organisation into knowledge representations.

Semantic Networks

- Semantic means ‘meaning’, hence, semantic network indicates a meaningful network of concepts. It constitutes two basic elements: node and arc.
- A semantic network is a structure for representing knowledge as a pattern of interconnected nodes and arcs
- Nodes in the net represent concepts of entities, attributes, events, values
- Arcs in the network represent relationships that hold between the concepts.
- A particular relation is specified by the label of arc.
- The basic structure of knowledge organization is provided by relationships.
- Semantic networks are typically used with a special set of accessing procedures which perform “reasoning”

Nodes and Arcs

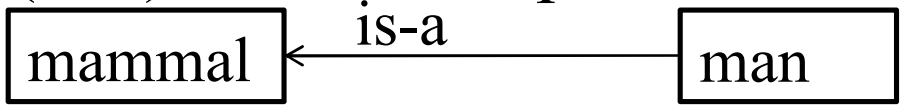
- Arcs define binary relations which hold between objects denoted by the nodes.



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

Types of relationships

- There are many types of relationships that can be used in semantic networks. The following are four of them.
- The “isa (is-a)” relationship between class and superclass.



- Here is superclass and man is class.

- The “is instance of“ relationship between instance and class.

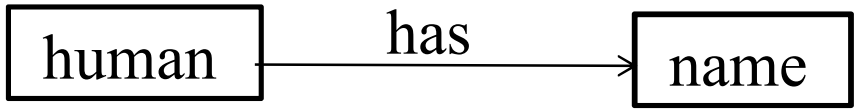
- Emu is an instance of the class bird.

Types of relationships

- The “ part of (is part)“ relationship between part and whole

- Tail is the part of the object dog

- The “has” relationship between object and attribute.



- The object human has the attribute name.

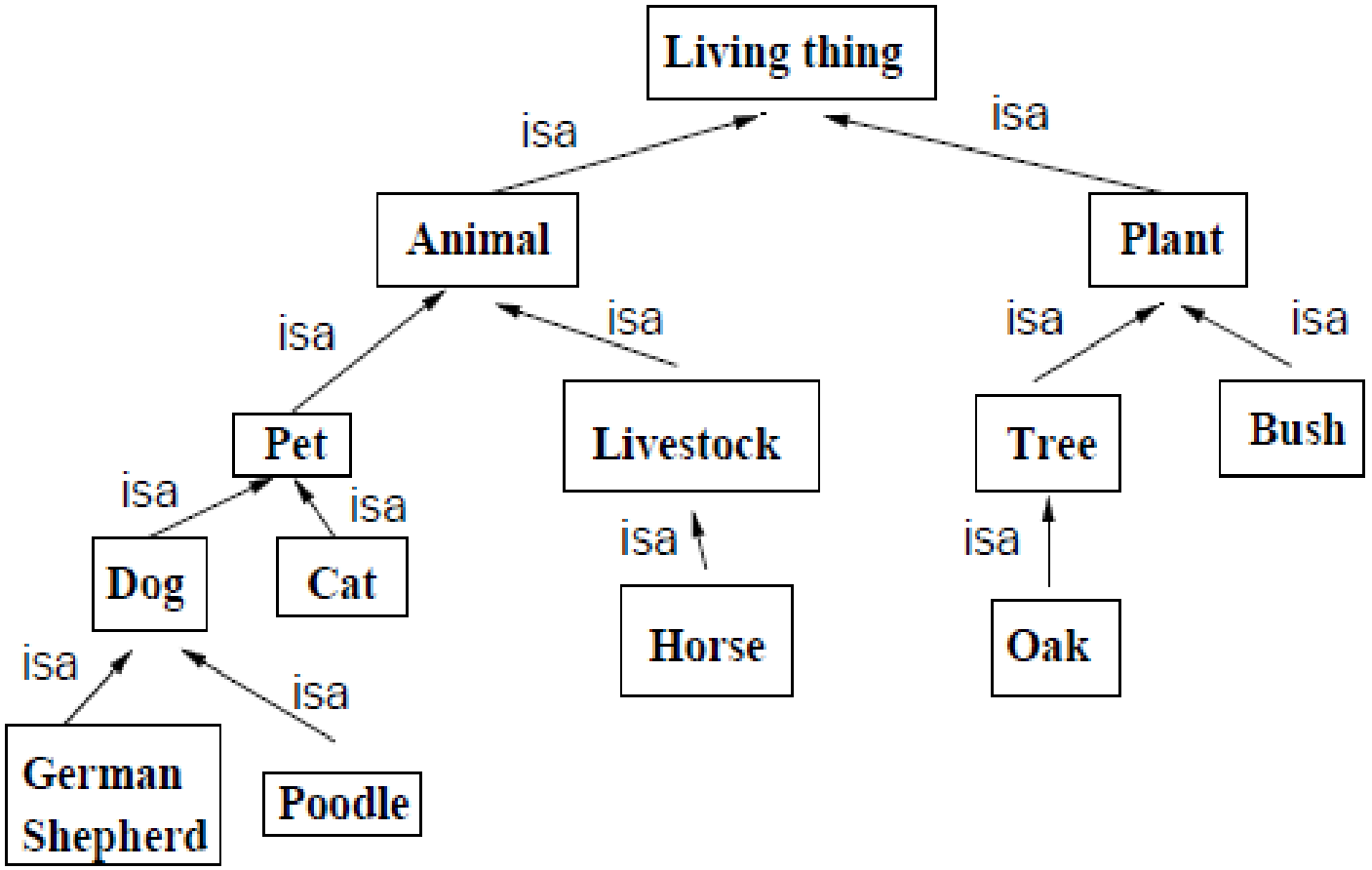
- Some function types of arcs are there depending on relationship.

ISA and INSTANCE arcs

- The link ***isa*** is used to relate one class to another
- ***isa*** relates generic nodes to generic nodes while the ***instance*** relates an instance or individual to a generic class
- The more general class that an ***isa*** arrow points to is called a Superclass and also ***isa*** points from a subclass to a class
- The objects in a class have one or more attributes in common
- The ***instance*** means "is an instance of' and refers to a specific member of a class
- A class is related to the mathematical concept of a set in that it refers to a group of objects
- In set-theory terms, ***isa corresponds to the sub-set relation*** $\subseteq$, ***and instance corresponds to*** the membership relation $\in$.

An *IS-A Hierarchy*

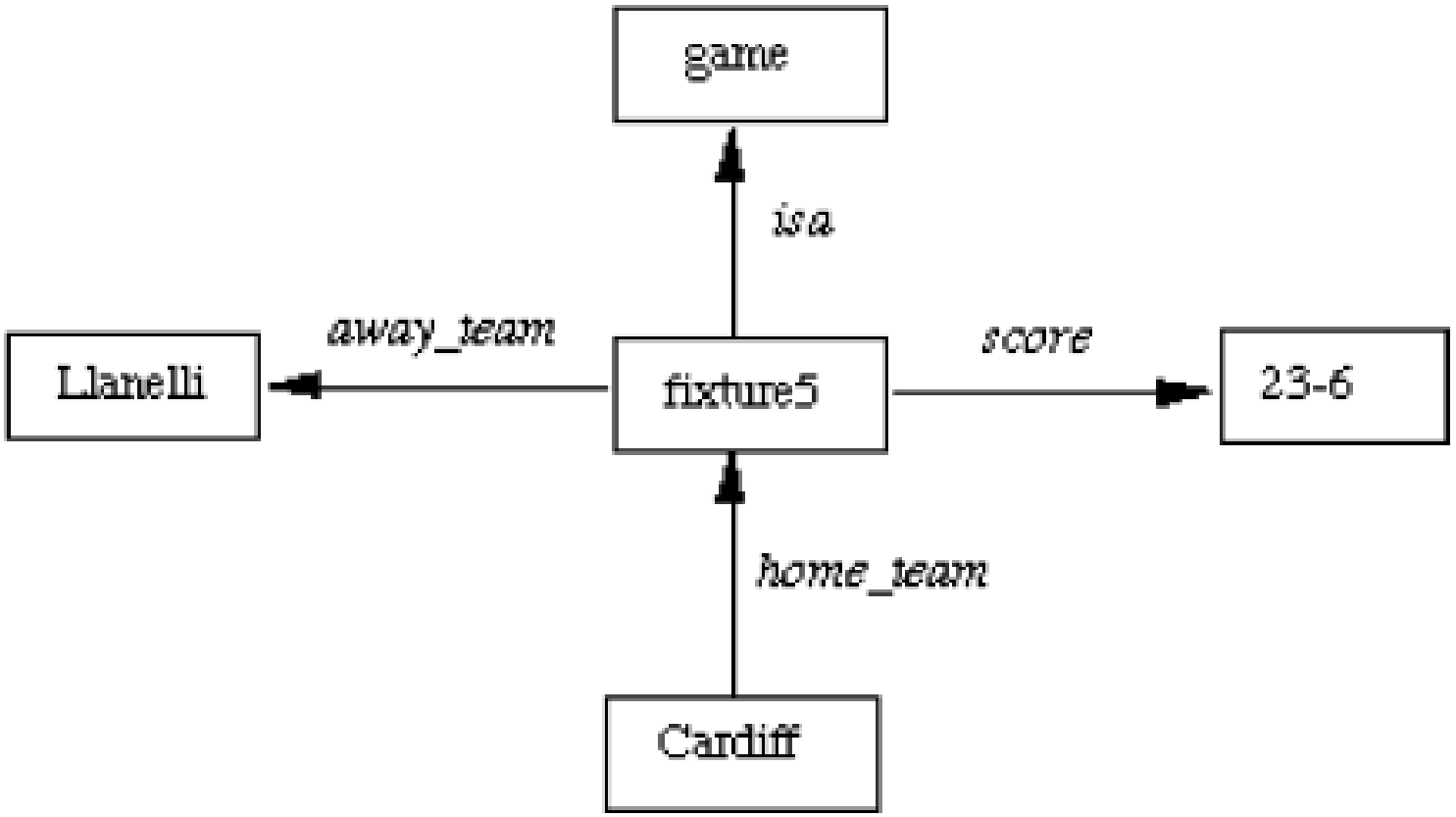
- A simple form of semantic network is an ***is-a hierarchy.***



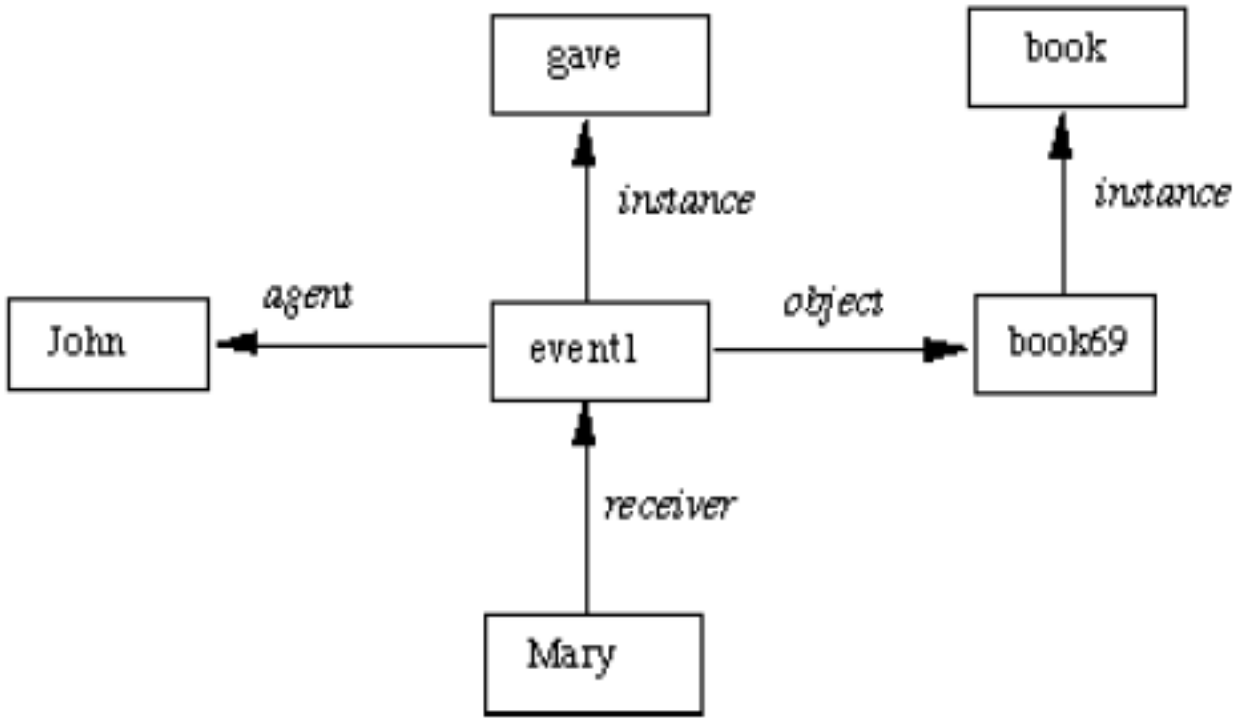
Non-binary relations

- But we have a problem: *How we can have more than 2 place predicates in semantic nets? E.g. score(Cardiff, Llanelli, 23-6)*
- Non-binary relationships can be represented by "turning the relationship into an object"
- Create new nodes to represent new objects either contained or alluded to in the knowledge, *game and fixture in the current example.*
- Relate information to nodes and fill up slots.

- We can represent non-binary relation for the given example *score(Cardiff, Llanelli, 23-6) as*



- Consider the sentence: *John gave Mary the book. Here we have several aspects of an event.*

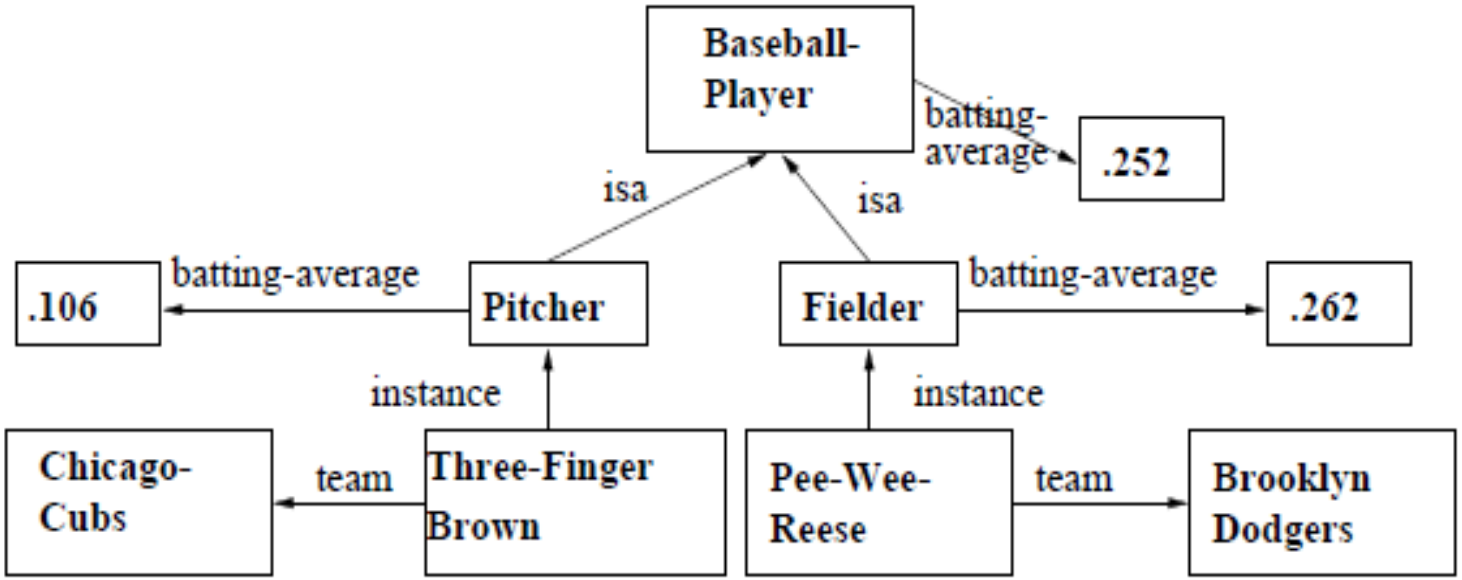


Inference in a Semantic Net

- Basic inference mechanism considers the *links between nodes*.
- Two methods are considered:
  - **Intersection search**
  - **Inheritance**

Intersection search

- *Spreading activation* from each of two nodes and seeing where the activations met .
- *Intersection of nodes finds relationships among objects. This is achieved by assigning a special tag to each visited node.*
- Question: "What is the relation between Chicago cubs and Brooklyn Dodgers?"structured networks.
- Answer: "They are both teams of baseball players."

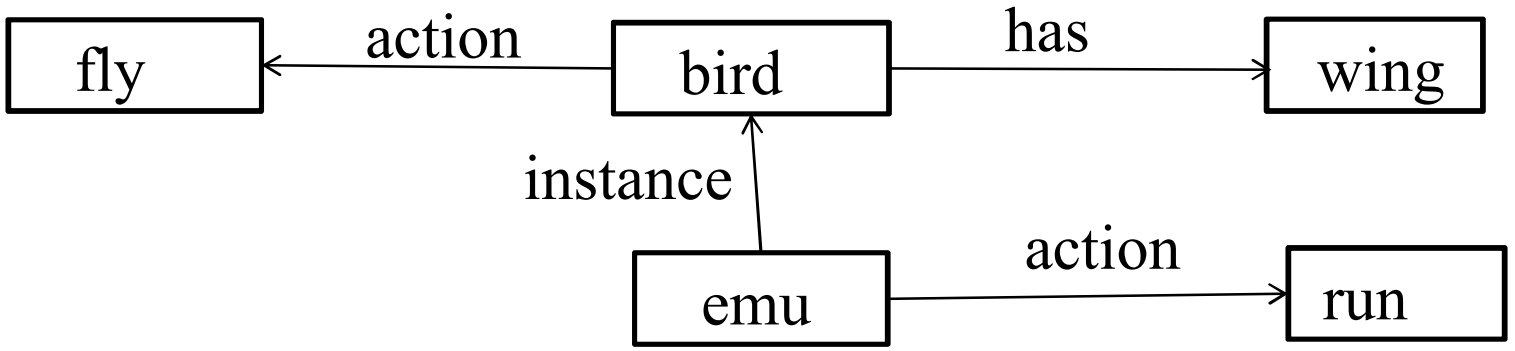


Inheritance

- One of the main types of reasoning done in a semantic net is the inheritance of values (properties) along the subclass and instance links.
- The *isa* and *instance* representation provide a mechanism to implement this.
- The local knowledge of a superclass node is referred by class node, instance node.
- Inheritance also provides a means of dealing with *default reasoning.*

Contd.

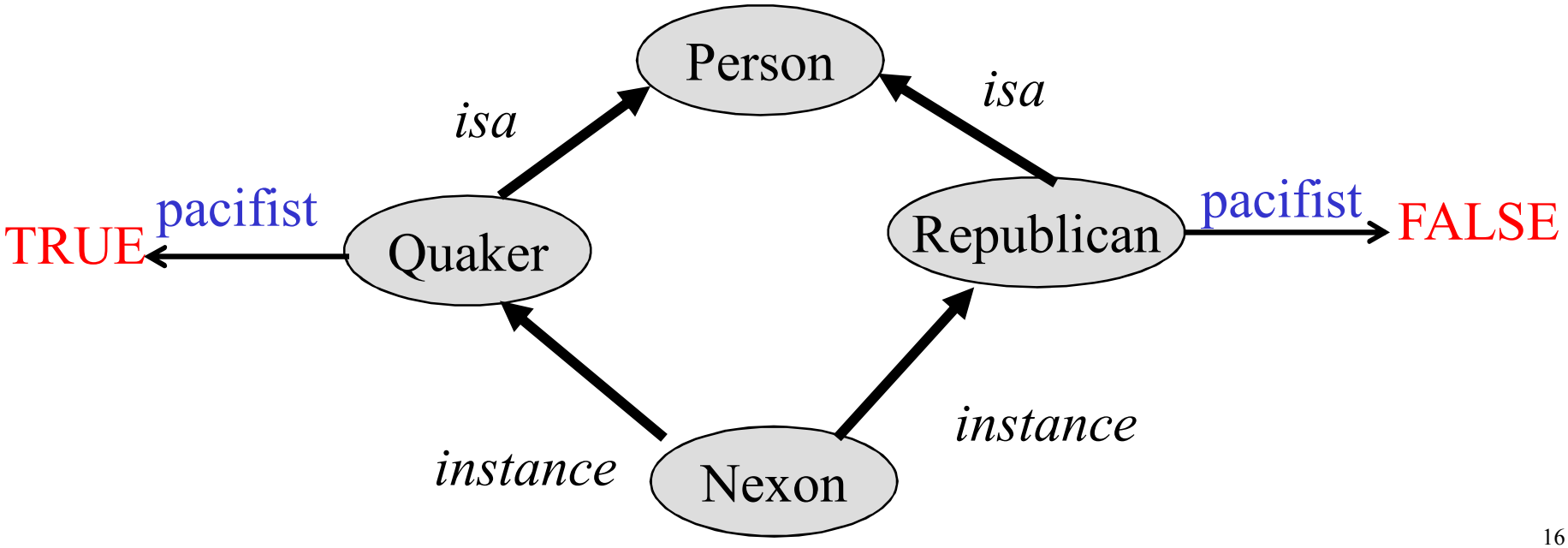
- Attribute values are inherited from higher up the hierarchy. This is more efficient than listing all the details at each level.



- Here emu can fly and has wings, i.e. inherits the attributes of bird.

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.
- Conflict or inconsistent properties can be inherited from different ancestors
- All Quakers are pacifists. All Republicans are not pacifists. Nixon is a Republican. Nixon is a Quaker. Quakers and Republicans are Persons.



Multiple inheritance

- Rules are used to determine inheritance in such "tangled" networks where multiple inheritance is allowed:
  - if $X \subseteq A \subseteq B$ and both A and B have property P (possibly with different variable instantiations), then X inherits A's property P instance (closer ancestors override far away ones).
  - If $X \subseteq A$ and $X \subseteq B$ but neither $A \subseteq B$ nor $B \subseteq A$ and both A and B have property P with different and inconsistent values, then X will not inherit property P at all; or X will present both instances of P (from A and B) to the user

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

Frame

- Frame was proposed by Minsky in 1975 as an extension of the semantic network.
- There is not clear the distinction between semantic net and frame.
- Semantic nets initially used to represent labeled connections between objects.
- As tasks became more complex the representation needs to be more structured.
- The more structured the system it becomes more beneficial to use frames.

Frame

- Frames are descriptions of conceptual individuals. Frames can exist for ``real'' objects or more ``abstract'' objects .
- A Frame system is a collection of objects. Each object contains a number of *slots*.
- A slot represents an attribute.
- Each slot has a value that describe some real world entity .
- The value of a slot is known as filler. The value of a slot can be another frame.

3 components of a frame

- frame name
- attributes (slots)
- values (fillers: list of values, range, string, etc.)

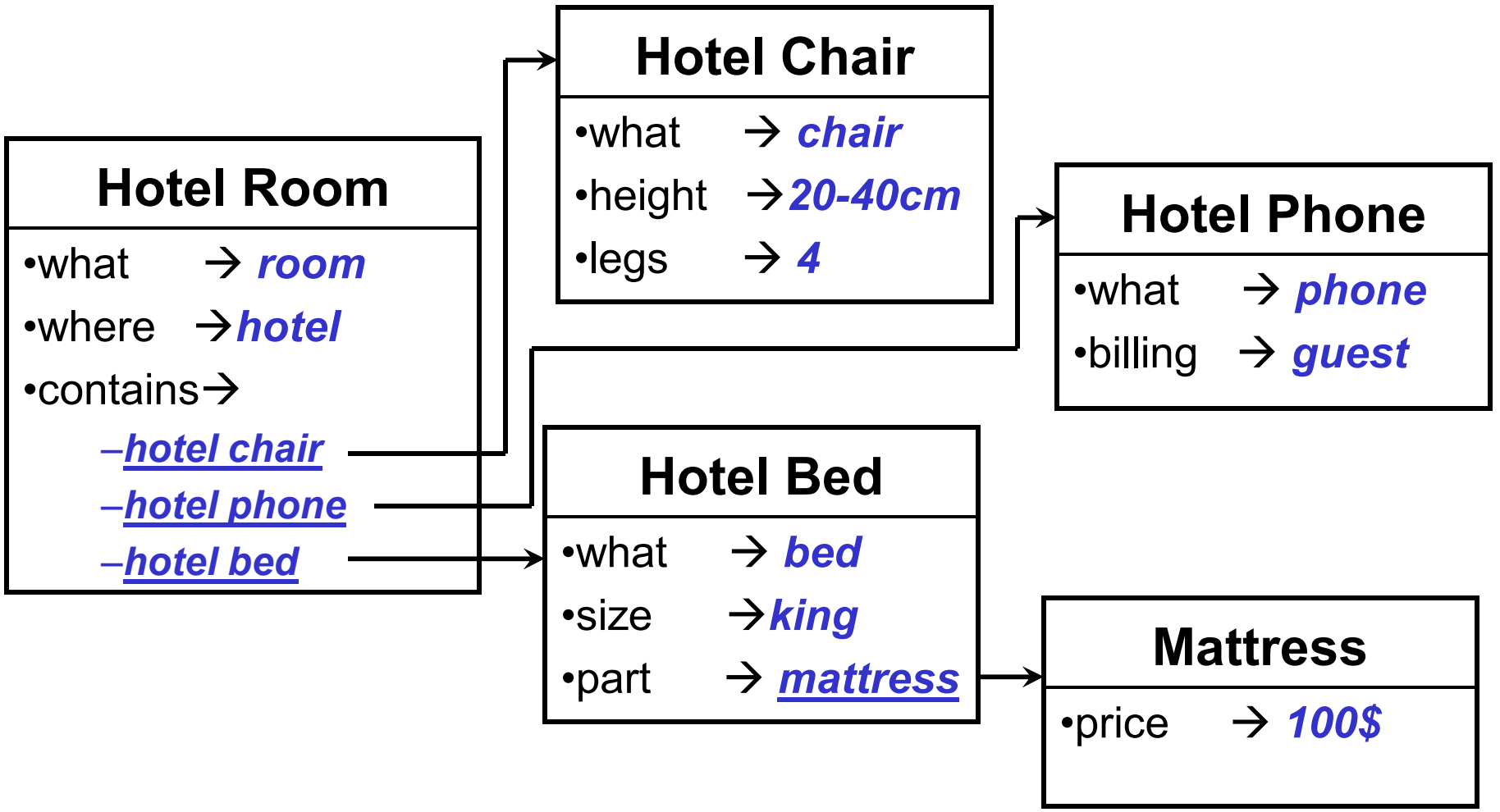
Book Frame
Slot → ***Filler***
•Title → ***AI. A modern Approach***
•Author → ***Russell & Norvig***
•Year → ***2003***

Frame

- A frame carries with it a set of *slots* that can represent objects that are normally associated with a subject of the frame.
- Frames are also represented by their relationships with other frames. Relationships between frames are represented using *slots*.
- If a frame f is in a relationship r to a frame g, then we put the value g in the r slot of f.
- The slots can then point to other slots or frames. That gives frame systems the ability to carry out inheritance.

Inheritance

- Similar to Object-Oriented programming paradigm

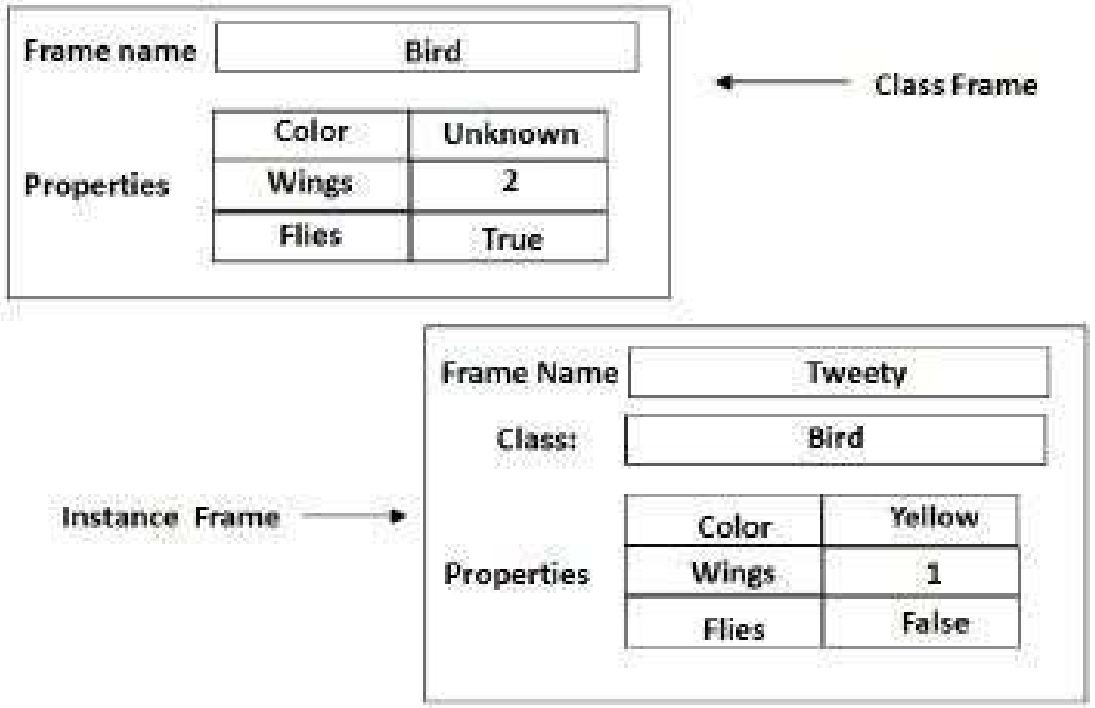


Frame as Set

- Set theory provides a good basis for understanding frame systems.
- Each frame represents:
  - a class (set), or
  - an instance (an element of a class).
- A frame carries with it a set of *slots* that can represent objects that are normally associated with a subject of the frame.
- The *isa* relation is in fact the subset relation.
- The *instance* relation is in fact element of.
- The *isa* attribute possesses a transitivity property.

Contd.

- As a class represents a set, there are 2 kinds of attributes that can be associated with it.
- 1. Its own attributes &
- 2. Attributes that are to be inherited by each element of the set.



Contd.

- Sometimes, the difference between a set and an individual instance may not be clear.

  Example: Team India is an instance of the class of Cricket Teams and can also think of as the set of players. Now the problem is if we present Team India as a subclass of Cricket teams, then Indian players automatically become part of all the teams, which is not true.

- Consider Team India a subclass of class called Cricket Players
- We need to differentiate between regular classes and meta-classes.
- The elements of Regular Classes are individual entities.
- Meta-class is a special class whose elements are classes. It represents the set of all classes. All classes are instances of it, either directly or through one of its subclasses.

Person
isa: Mammal
cardinality: 6,000,000,000
*handed: Right
Adult-Male
isa: Person
cardinality: 2,000,000,000
*height; 5-10
ML-Baseball-Player
isa: Adult-Male
cardinality: 624
*height: 6-1
*bats: equal to handed
*batting-average: . 252
*team:
*uniform-color:
Fielder
Isa: ML-Baseball-Player
cardinality: 36
*batting-average .262
Johan
insance: Fielder
height: 5-10
bats: Right
batting-average: . 309
team: Brooklyn-Dodgers
uniform-color: Blue
ML-Baseball-Team
isa: Team
cardinality: 26
*team-size: 24
*manager: 24
Brooklyn-Dodgers
instance: ML-Baseball-Team
team-size: 24
manager: Leo-Durocher
players: (Johan,Pee-Wee-Reese,…)

Example

- Frames Person, Adult-Male, ML-Baseball-Player, Fielder and ML-Baseball-Team are all classes.
- The frames Johan and Brooklyn-Dodgers are instances.
- The ***isa*** relation is the subset relation like the set of *adult males* is a subset of the set of *people*.
- The instance relation of Johan is an element of the set of fielders, so it is also an element of all of the supersets of fielders.
- Attributes that are to be inherited by each element of the set are marked as *.
- Five properties that all ML-Baseball players have (height, bats, batting average, team, and uniform-colour), and first three are specified by default values.

Demon

- One of the main advantages of frames is the ability to include demons to compute slot values.
- **Demon** - a function that computes the value of a slot on demand.
- Demons are attached to slots and which run whenever the slot is read from or written to.

HUMAN
isa: MAMMAL
cardinality: 6 million
*mortal: yes
*age: demon compute_age)

MARY
instance: HUMAN
gender: FEMALE
birthday: 11/04/90

intCompute_Age
return(today- (query birthday slot));

Benefits of Frames

- Makes programming easier by grouping related knowledge
- Easily understood by non-developers
- Expressive power
- Easy to set up slots for new properties and relations
- Easy to include default information and detect missing values

Drawbacks of Frames

- No standards (slot-filler values)
- More of a general methodology than a specific representation:
  - Frame for a class-room will be different for a professor and for a maintenance worker
- No associated reasoning/inference mechanisms