

# **Knowledge Representation & Reasoning**

---

Shyamanta M Hazarika  
CSE, SoE  
Tezpur University

# Taxonomies vs. frame hierarchies

---

The taxonomies in DL look like the **IS-A** hierarchies with frames.

There is a big difference, however:

- in frame systems, the KB designer gets to decide what the fillers of the :IS-A slot will be; the :IS-A hierarchy is constructed manually
- in DL, the taxonomy is completely determined by the meaning of the concepts and the subsumption relation over concepts

For example, a concept such as

[AND Fish [FILLS :Size large]]

must appear in the taxonomy below Fish even if it was first constructed to be given the name Whale. It cannot simply be positioned below Mammal.

To correct our mistake, we need to associate the name with a different concept:

[AND Mammal [FILLS :Size large] ...]

# Inheritance and propagation

---

As in frame hierarchies, atomic concepts in DL inherit properties from concepts higher up in the taxonomy.

For example, if a Doctor has a medical degree, and Surgeon is below Doctor, then a Surgeon must have a medical degree.

This follows from the logic of concepts:

If  $KB \models (\text{Doctor} \sqsubseteq [\text{EXISTS } 1 : \text{MedicalDegree}])$   
and  $KB \models (\text{Surgeon} \sqsubseteq \text{Doctor})$   
then  $KB \models (\text{Surgeon} \sqsubseteq [\text{EXISTS } 1 : \text{MedicalDegree}])$

This is a simple form of *strict* inheritance (*cf.* next chapter)

Also, as noted in computing satisfaction (e.g. with joe and canCorp), adding an assertion like  $(c \rightarrow e)$  to a KB can cause other assertions  $(c' \rightarrow e')$  to be entailed for other individuals.

This type of propagation is most interesting in applications where membership in classes is monitored and changes are significant.

# Extensions to the language

---

A number of extensions to the DL language have been considered in the literature:

- upper bounds on the number of fillers

[AND [EXISTS 2 :Child] [AT-MOST 3 :Child]]

opens the possibility of inconsistent concepts

- sets of individuals: [ALL :Child [ONE-OF wally theodore]]
- relating the role fillers: [SAME-AS :President :CEO]
- qualified number restriction: [EXISTS 2 :Child Female] vs.  
[AND [EXISTS 2 :Child] [ALL :Child Female]]
- complex (non-atomic) roles: [EXISTS 2 [RESTR :Child Female]]  
[ALL [RESTR :Child Female] Married] vs.  
[ALL :Child [AND Female Married]]

Each of these extensions adds extra complexity to the problem of calculating subsumption.

This topic will be explored for RESTR in Chapter 16.

# Some applications

---

Like production systems, description logics have been used in a number of sorts of applications:

- interface to a DB
  - relational DB, but DL can provide a nice higher level view of the data based on objects
- working memory for a production system
  - instead of a having rules to reason about a taxonomy and inheritance of properties, this part of the reasoning can come from a DL system
- assertion and classification for monitoring
  - incremental change to KB can be monitored with certain atomic concepts declared “critical”
- contradiction detection in configuration
  - for a DL that allows contradictory concepts, can alert the user when these are detected. This works well for incremental construction of a concept representing e.g. a configuration of a computer.

# Hierarchy and inheritance

---

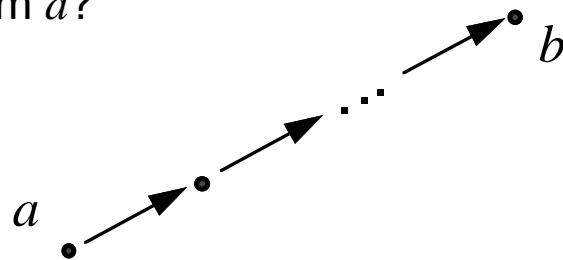
As we noticed with both frames and description logics, hierarchy or taxonomy is a natural way to view the world

importance of *abstraction* in remembering and reasoning

- groups of things share properties in the world
- do not have to repeat representations
  - e.g. sufficient to say that “elephants are mammals” to know a lot about them

Inheritance is the result of transitivity reasoning over paths in a network

- for strict networks, *modus ponens* (if-then reasoning) in graphical form
- “does  $a$  inherit from  $b$ ?” is the same as “is  $b$  in the transitive closure of **:IS-A** (or subsumption) from  $a$ ?”



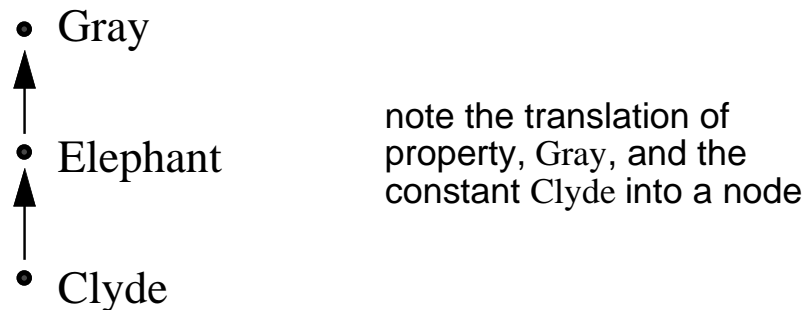
graphically, is there a path of **:IS-A** connections from  $a$  to  $b$ ?

# Path-based reasoning

---

## Focus just on inheritance and transitivity

- many interesting considerations in looking just at where information comes from in a network representation
- abstract frames/descriptions, and properties into nodes in graphs, and just look at reasoning with paths and the conclusions they lead us to

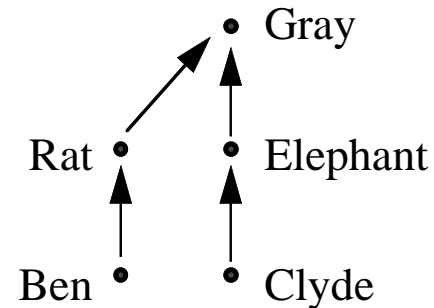


- edges in the network: Clyde·Elephant, Elephant·Gray
- paths included in this network: edges plus {Clyde·Elephant·Gray}  
in general, a path is a sequence of 1 or more edges
- conclusions supported by the paths:  
Clyde → Elephant; Elephant → Gray; Clyde → Gray

# Inheritance networks

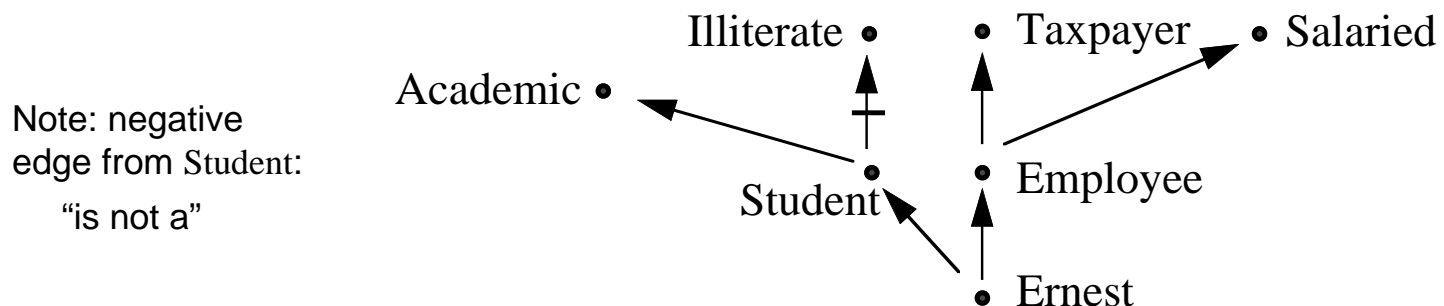
## (1) Strict inheritance in trees

- as in description logics
- conclusions produced by complete transitive closure on all paths (any traversal procedure will do); all reachable nodes are implied



## (2) Strict inheritance in DAGs

- as in DL's with multiple AND parents (= multiple inheritance)
- same as above: all conclusions you can reach by any paths are supported

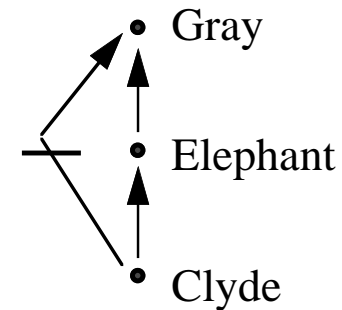


# Inheritance with defeasibility

## (3) Defeasible inheritance

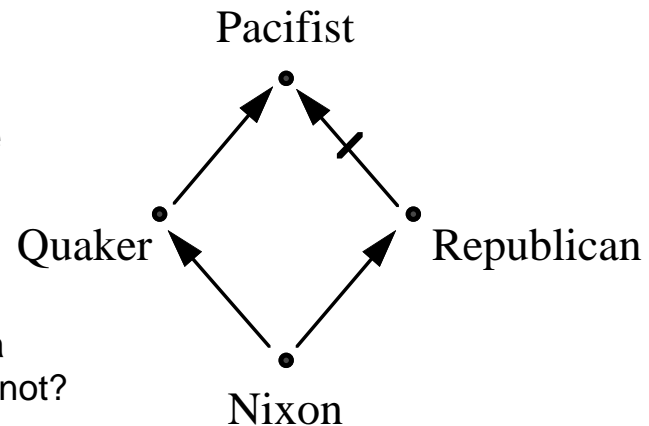
- as in frame systems
- inherited properties do not always hold, and can be *overridden* (defeated)
- conclusions determined by searching upward from “focus node” and selecting first version of property you want

while elephants in general are gray, Clyde is not



## A key problem: *ambiguity*

- *credulous* accounts choose arbitrarily
- *skeptical* accounts are more conservative



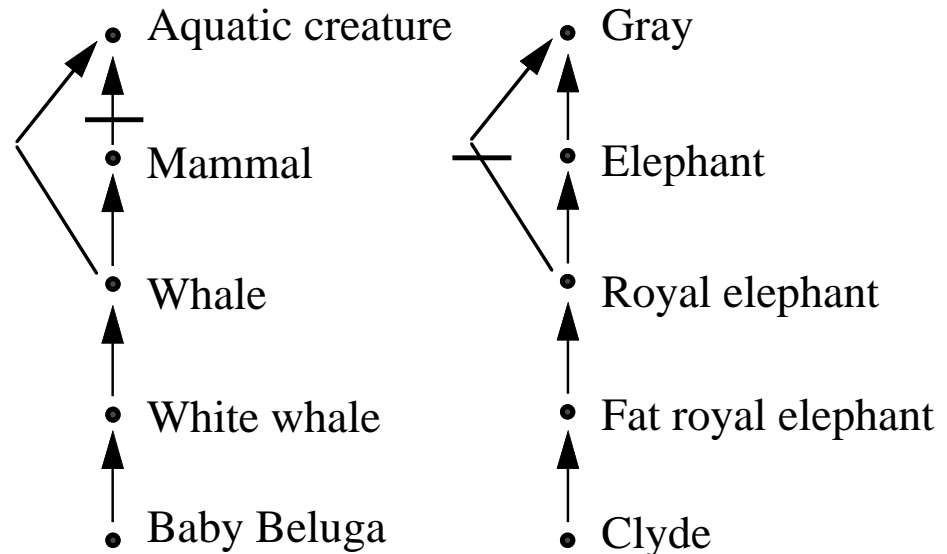
Is Nixon a pacifist or not?

# Shortest path heuristic

## Defeasible inheritance in DAGs

- links have *polarity* (positive or negative)
- use shortest path heuristic to determine which polarity counts

Intuition: inherit from the most specific subsuming class



- as a result, not all paths count in generating conclusions
  - some are “preempted”
  - but some are “admissible”

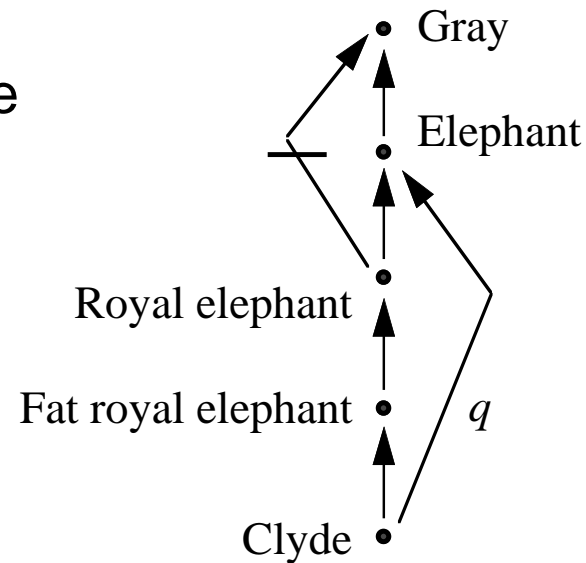
think of paths as *arguments* in support of conclusions

⇒ the inheritance problem = what are the admissible conclusions?

# Problems with shortest path

1. Shortest path heuristic produces incorrect answers in the presence of redundant edges (which are already implied!)

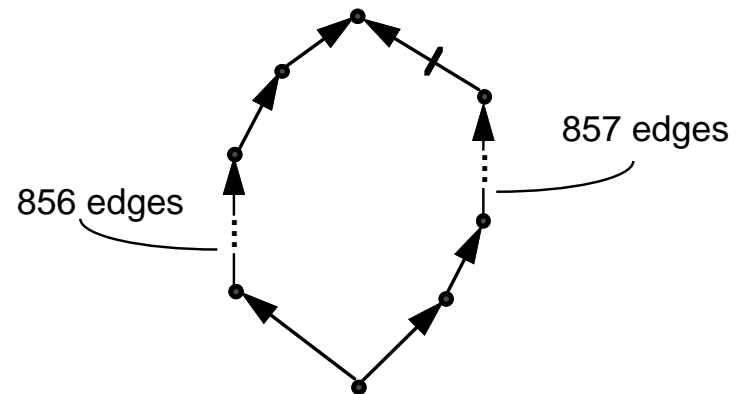
the redundant edge  $q$ ,  
expressing that Clyde is an  
Elephant changes polarity of  
conclusion about color



2. Anomalous behavior with ambiguity

adding 2 edges to the  
left side changes the  
conclusion!

Why should length be a factor?  
This network should be ambiguous...



# Specificity criteria

---

Shortest path is a specificity criterion (sometimes called a preemption strategy) which allows us to make admissibility choices among competing paths

- It's not the only possible one
- Consider “*inferential distance*”:  
not linear distance, but topologically based
  - a node  $a$  is nearer to node  $b$  than to node  $c$  if there is a path from  $a$  to  $c$  through  $b$
  - idea: conclusions from  $b$  preempt those from  $c$

This handles  $\text{Clyde} \rightarrow \neg\text{Gray}$  just fine,  
as well as redundant links

- But what if path from  $b$  to  $c$  has some of its edges preempted? what if some are redundant?

