COMP219: Artificial Intelligence

Lecture 17: Semantic Networks

Overview

- Last time
  - Rules as a KR scheme; forward vs backward chaining
- Today
  - Another approach to knowledge representation
    - *Structured objects*: *semantic nets*
      - Notation
      - Extended example
- Learning outcomes covered today:

  Distinguish the characteristics, and advantages and disadvantages, of the major knowledge representation paradigms that have been used in AI, such as production rules, semantic networks, propositional logic and first-order logic;

  Solve simple knowledge-based problems using the AI representations studied;

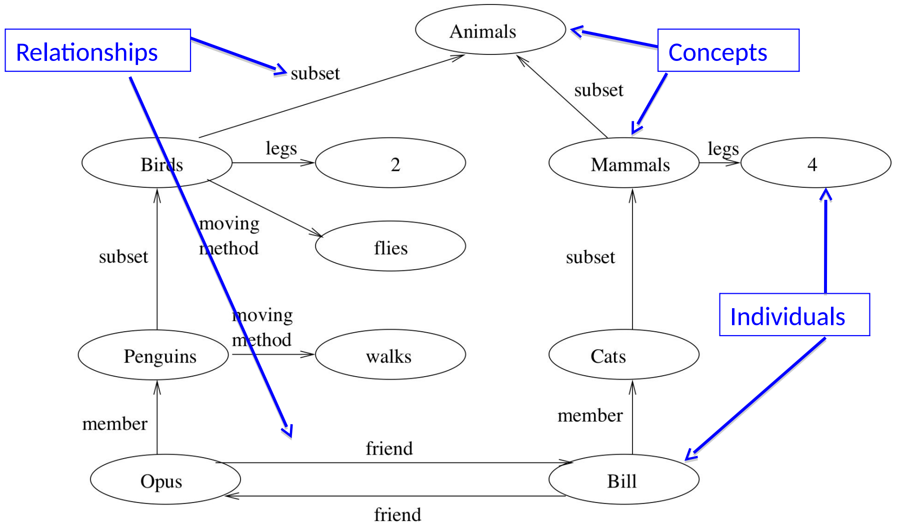
Structured Objects

- 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

Semantic Networks

- Charles S Peirce 1909 – *existential graphs*
- Quillian in 1968 – *semantic memory*
  - as a model for human memory; reasonable view of how semantic information is stored by humans
  - associative reasoning (via links)
- Semantic net is a *labelled graph*
  - nodes in graph represent *objects*, *concepts*, or *situations/events*
  - arcs in graph represent *relationships* between these things

Semantic Networks

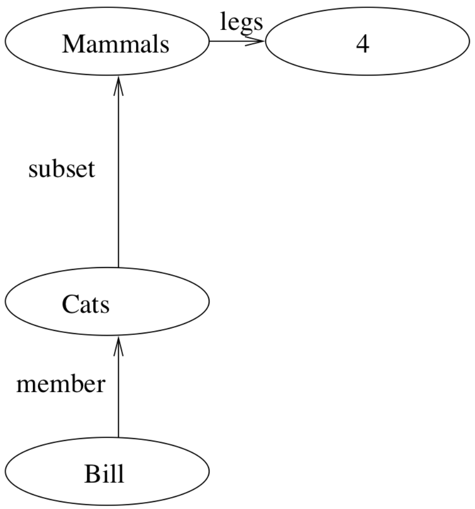


Important Arc Types

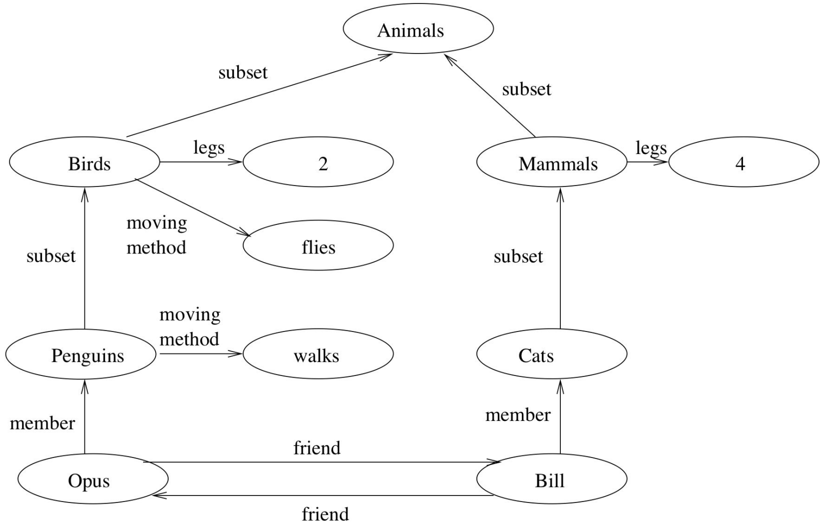
- Subset
  - X is a kind of Y
  - Penguin subset Bird: Concept to Concept
- Member
  - X is a Y: X is an instance of Y
  - Opus member Penguin: Individual to Concept
- R-relation
  - X relation-name Y
  - Opus is a friend of Bill; Lou is a parent of Ian Individual to Individual

Inheritance

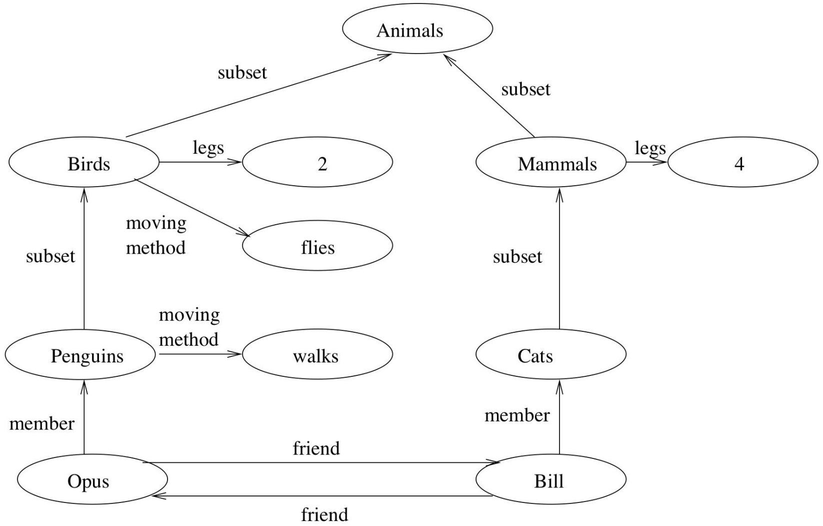
- Inheritance is one of the main kinds of reasoning done in semantic nets
- The subset relation is often used to link a class and its superclass
- Some links (e.g. legs) are inherited along subset paths
- Many variants of semantic nets
  - semantics can be relatively informal or very formal
  - “What's in a Link: Foundations for Semantic Networks” (Woods, 1975)



Example

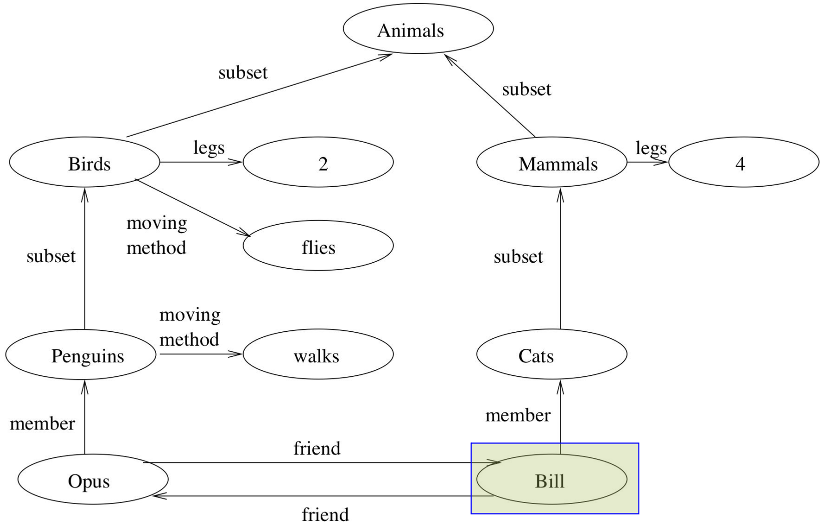


Example



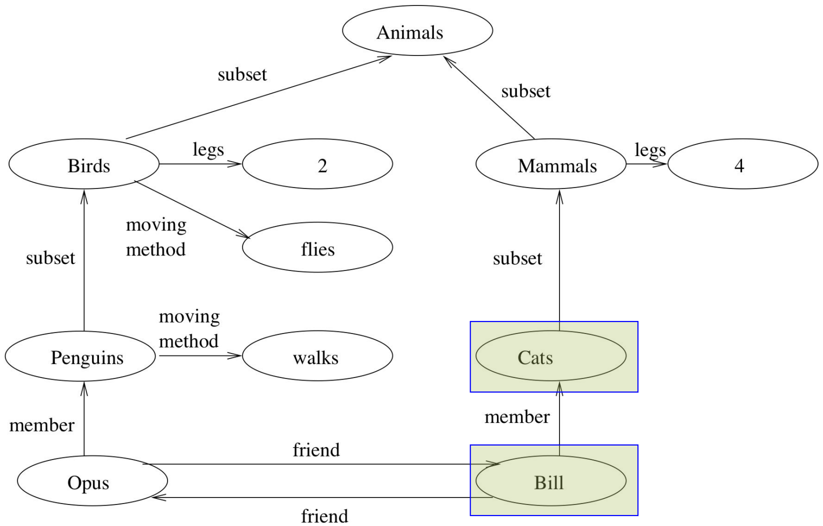
Bill has four legs

Example



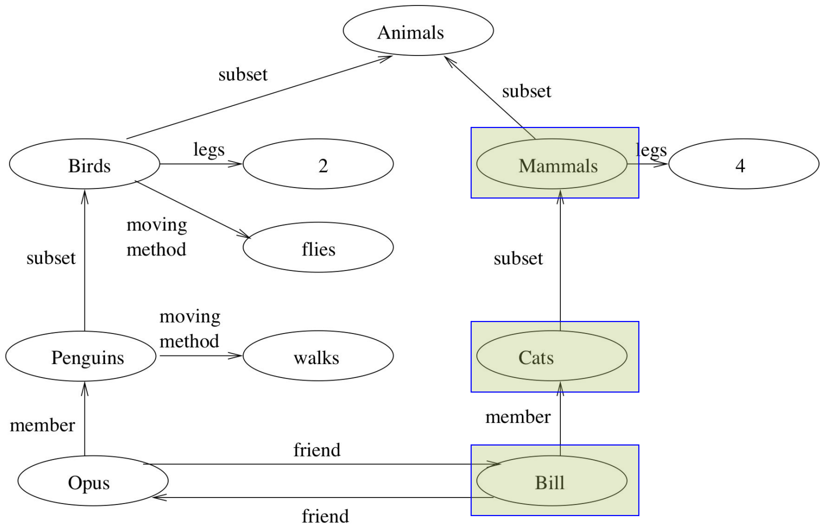
Bill has four legs

Example



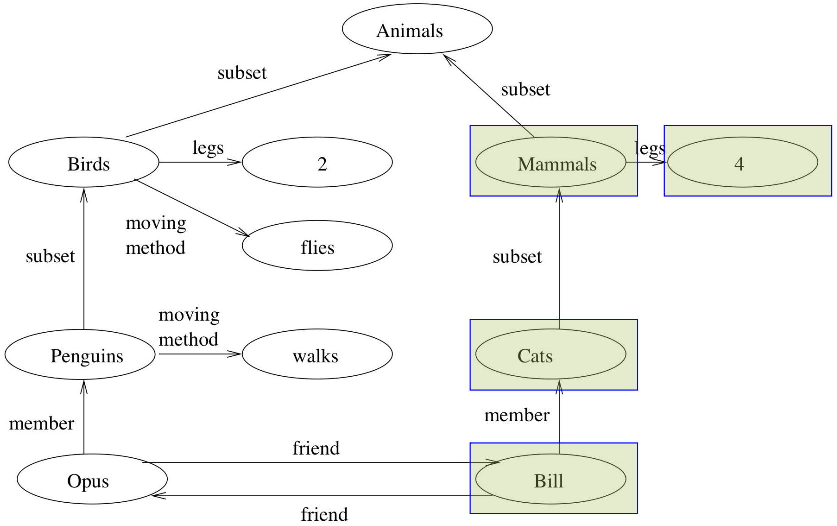
Bill has four legs

Example



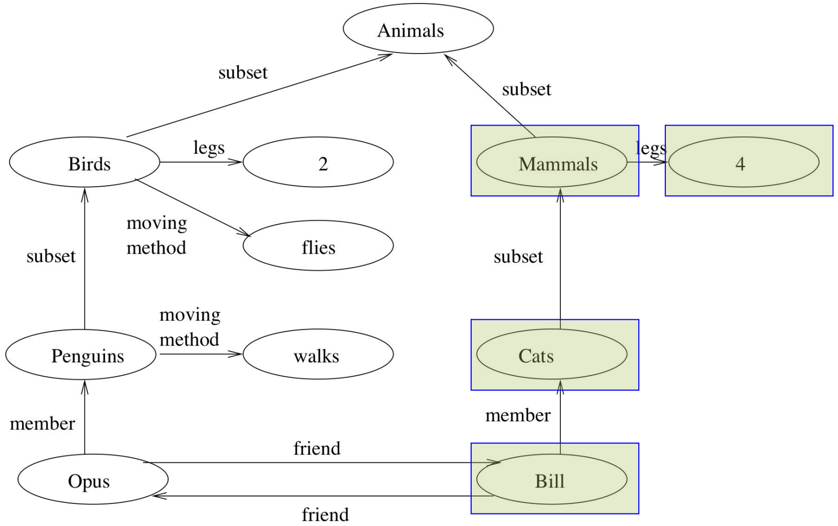
Bill has four legs

Example



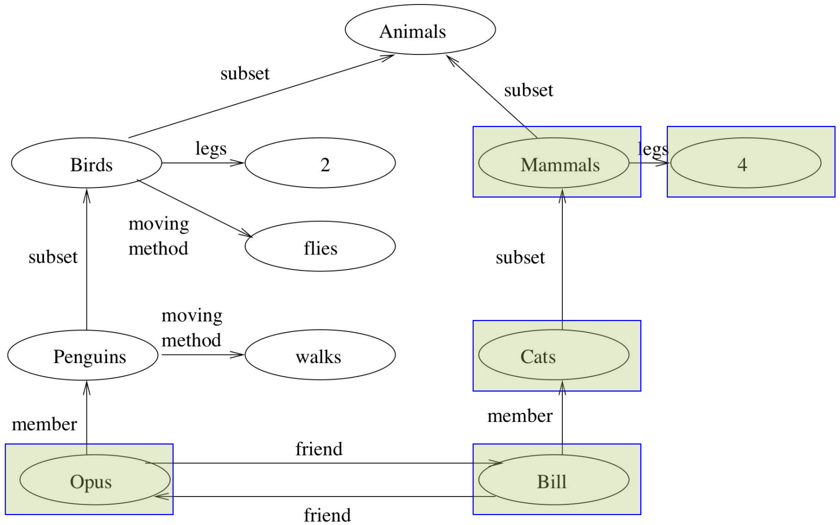
Bill has four legs

Example



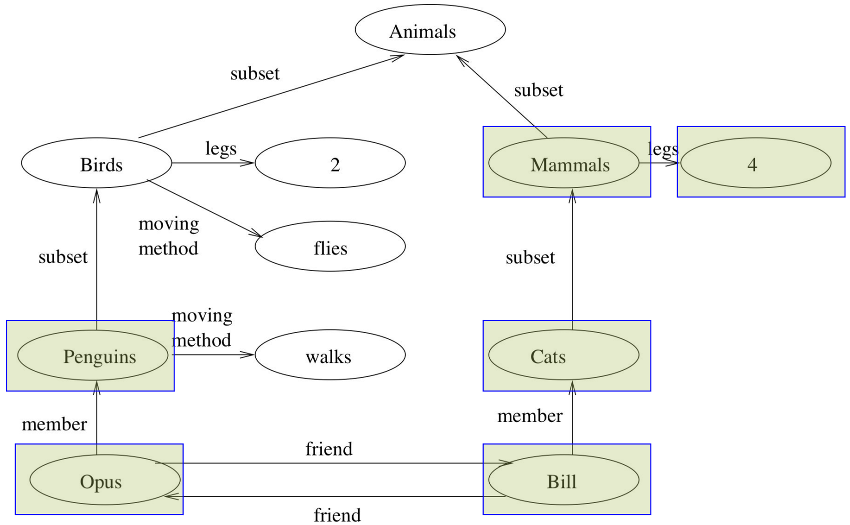
Bill has four legs *Opus is a Bird*

Example



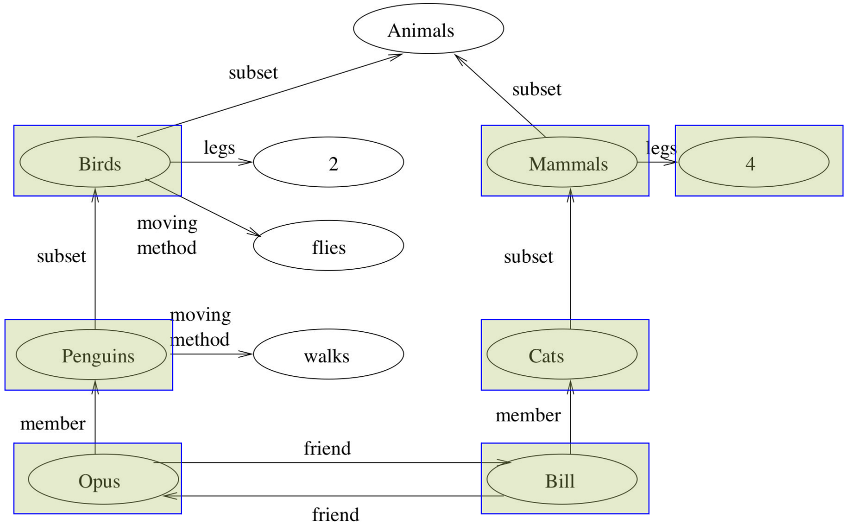
Bill has four legs *Opus is a Bird*

Example



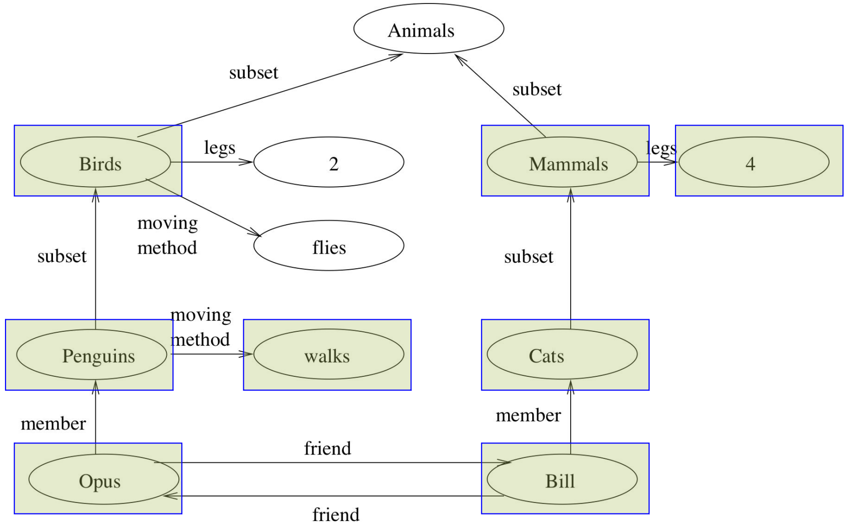
Bill has four legs *Opus is a Bird*


Example



Bill has four legs *Opus is a Bird*


Example

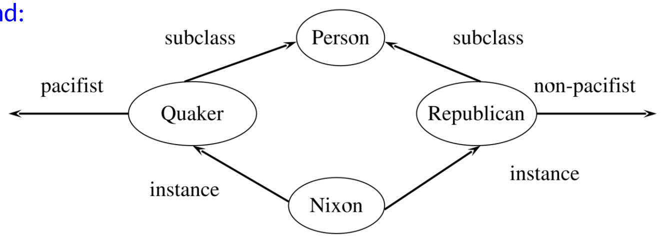


Bill has four legs *Opus is a Bird* *Opus walks*

Multiple Inheritance

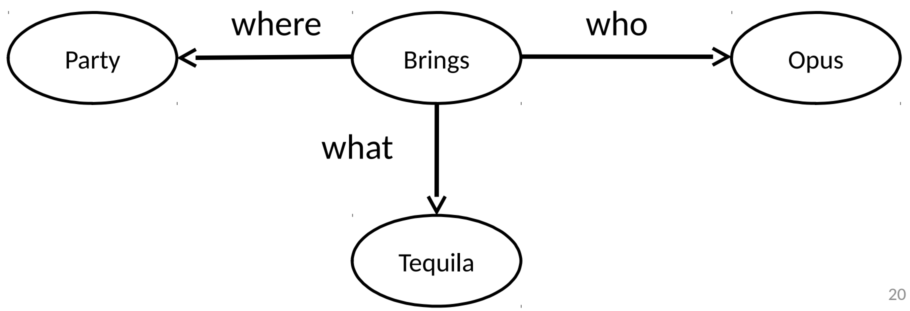
- A node can have any number of superclasses that contain it, enabling a node to inherit properties from multiple parent nodes and their ancestors in the network. It can cause conflicting inheritance

Nixon Diamond:



Problems with Semantic Nets

- Binary relations are easy to represent
- Others are harder
- Example: “Opus brings tequila to the party”



Exercise

- Suppose we have the information “Bill brings whiskey to the party”.
- How could we extend the semantic network to include this information?
- Can you see any problems with the reasoning in the example once we introduce this information?

Binary Relations

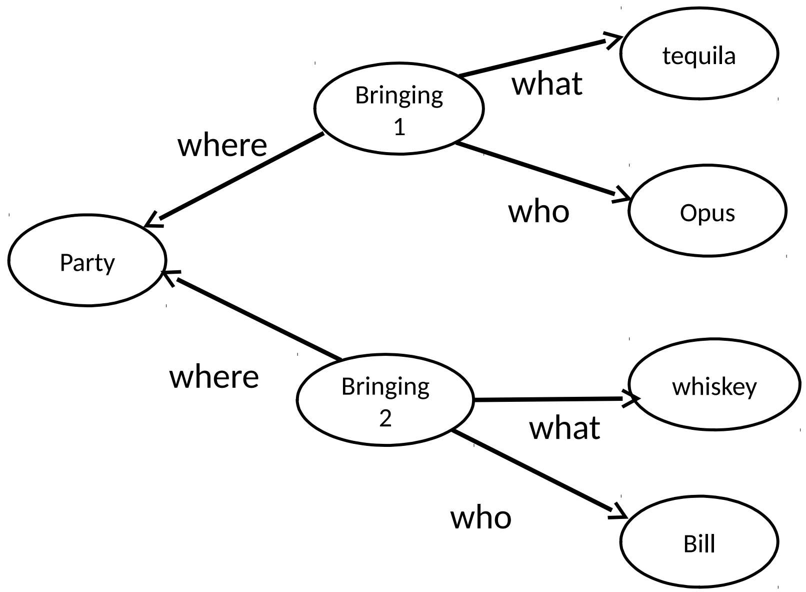
- Any relation can be rewritten as a set of binary relations
- Bringing-1(Opus,tequilla,party)
- Bringing-2(Bill,whiskey,party)
- Make the event a thing and make one binary relation ***per role***
  - who(bringing-1,Opus); who(bringing-2,Bill)
  - what(bringing-1,tequila); what(bringing-2,whiskey)
  - where(bringing-1,party); where(bringing-2,party)

Binary Relations

- Any relation can be rewrit relations
- Bringing-1(Opus,tequilla,p
- Bringing-2(Bill,whiskey,pa
- Make the event a thing and make one binary relation ***per role***
  - who(bringing-1,Opus); who(bringing-2,Bill)
  - what(bringing-1,tequila); what(bringing-2,whiskey)
  - where(bringing-1,party); where(bringing-2,party)

Turn relation into a thing:
- **'thingification'** (McCarthy)
- more common:
  - **reification**
  - from Latin (*res*)

Now we can see who brought what



Other Problems are Harder

- Negation
  - Opus and Dirk are not friends
    - Can just assume an absence of a link
- Cancellation/Exception
  - Property inherited from a distant superclass cancelled at a lower level
    - Birds fly, penguins don't
- Disjunction
  - Opus either drinks tea or coffee
- Quantification
  - "every dog has bitten a postman"
  - "every dog has bitten every postman"

Advantages of Semantic Nets

- Easy to visualise
- Flexible: relationships can be arbitrarily defined by the knowledge engineer
- Formal definitions of semantic networks have been developed
- Related knowledge is easily clustered
- Efficient in space requirements
- Objects represented only once
- Inference reduced to search

Disadvantages of Semantic Nets

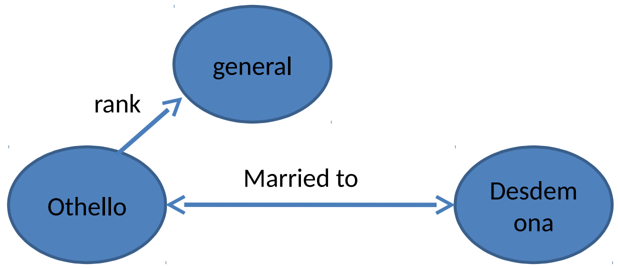
- Inheritance (particularly from multiple sources and when exceptions in inheritance are required) can cause problems
- No standards about node and arc types, and semantics might not be quite clear (what does "IS-A" mea?n)
- Limited expressiveness: may require a number of specially coded procedures ('procedural attachment')
- The above problems make it difficult to
  - verify and validate the systems
  - share knowledge
  - reuse knowledge
  - acquire knowledge methodically

wikipedia

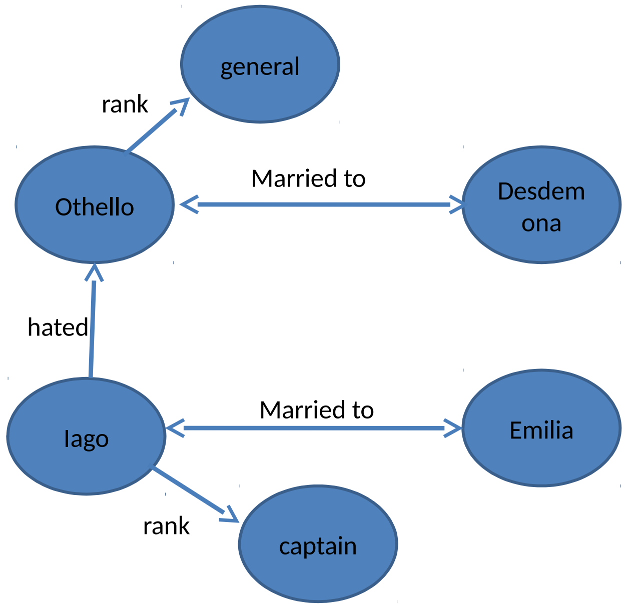
The Story of Othello

- Othello was a general who was married to Desdemona
- Iago was a captain who was married to Emilia; he hated Othello
- Iago told Othello lies about Desdemona
- Othello killed Desdemona with a pillow. He felt remorse and killed himself with a dagger

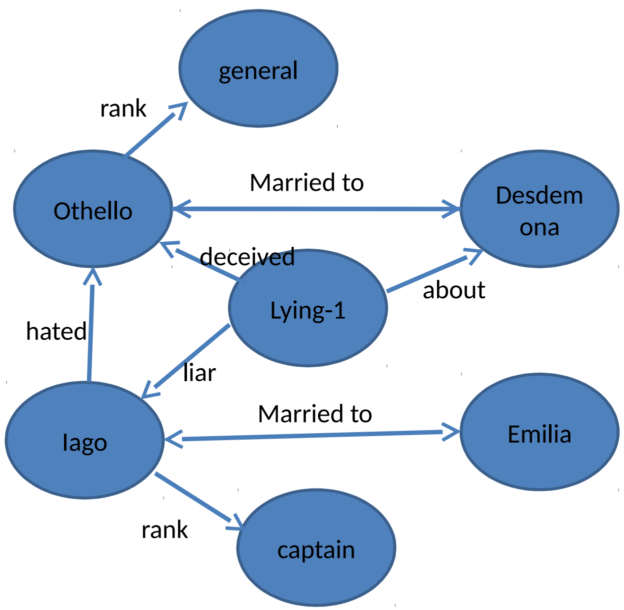
Othello was a general who was married to Desdemona

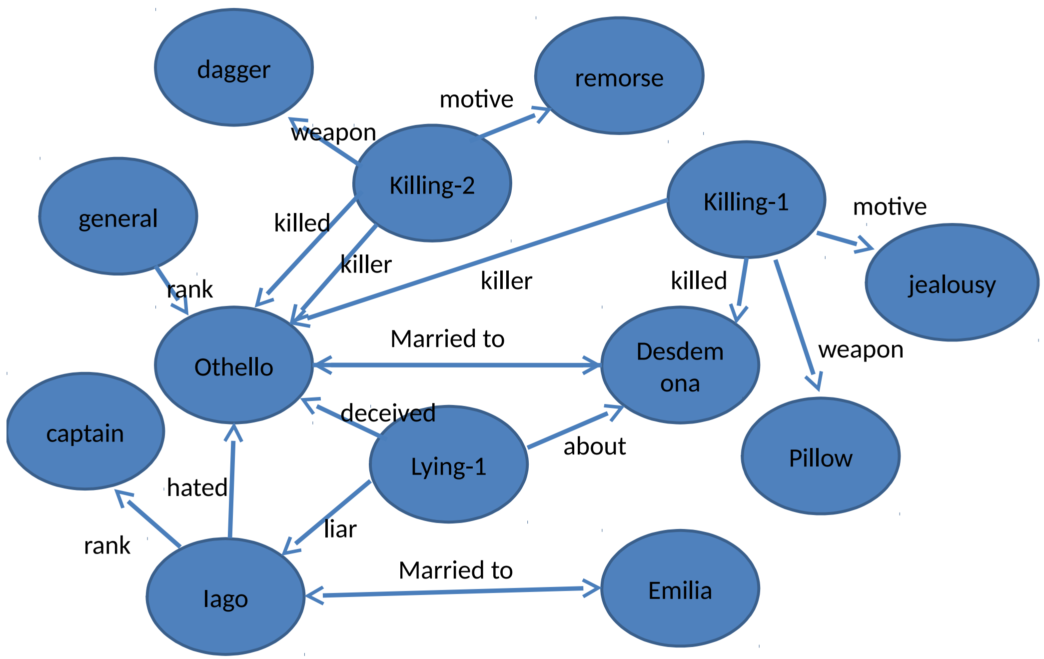


Iago was a captain who was married to Emilia; he hated Othello



Iago told Othello lies about Desdemona





Othello killed Desdemona with a pillow. He felt remorse and killed himself with a dagger

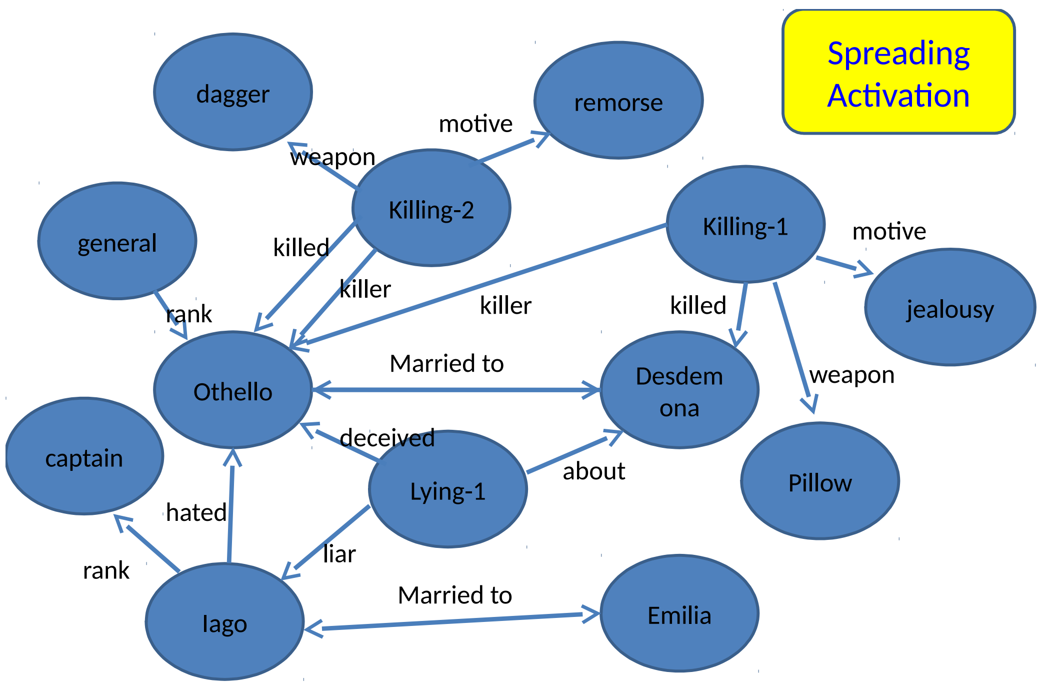
Prolog – Organised by Relations

```
marriedTo(Husband,Wife).
marriedTo(X,Y):-marriedTo(Y,X).
rank(Soldier,Rank).
male(Person).
alive(Person).
killing(Killer,Killed,Weapon,Motive).
motiveForKilling(Person,Motive):-
   killing(Person,_,_,Motive).
```

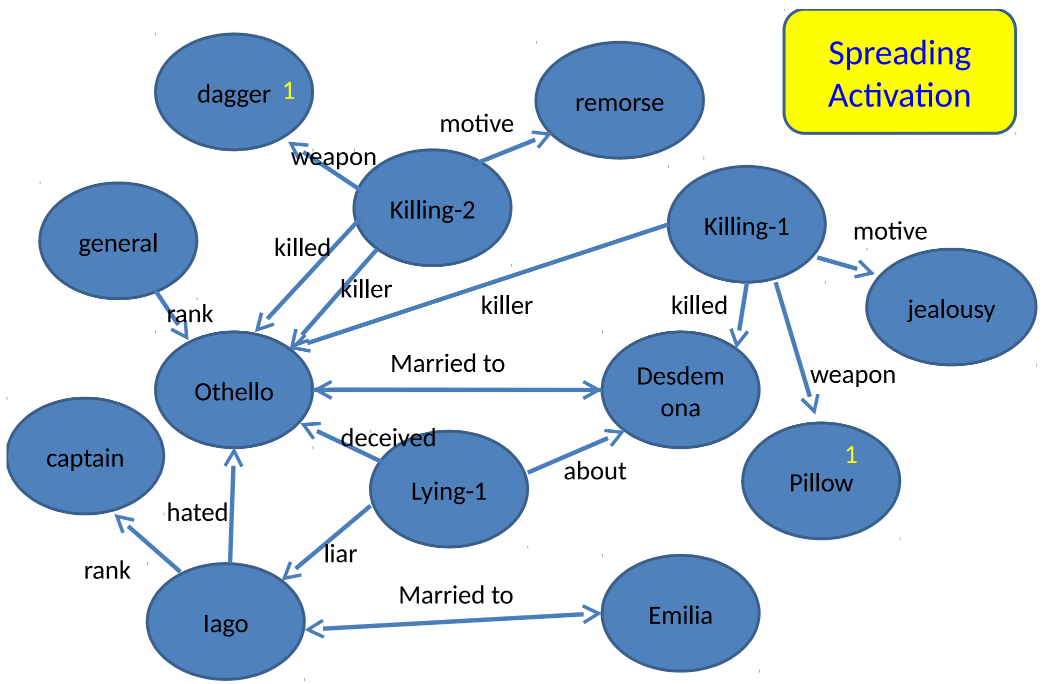
And so on…

Manipulating the Knowledge

- So far we have represented the knowledge in a variety of ways
- We also need to manipulate the knowledge
- This can be done in a variety of ways

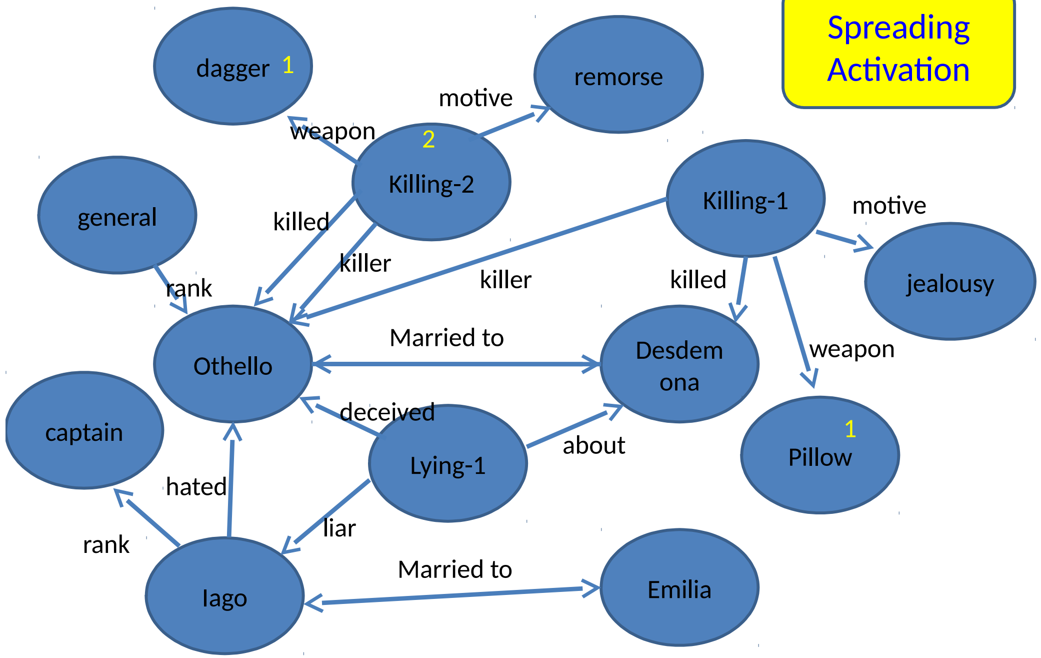


What do the pillow and the dagger have in common?



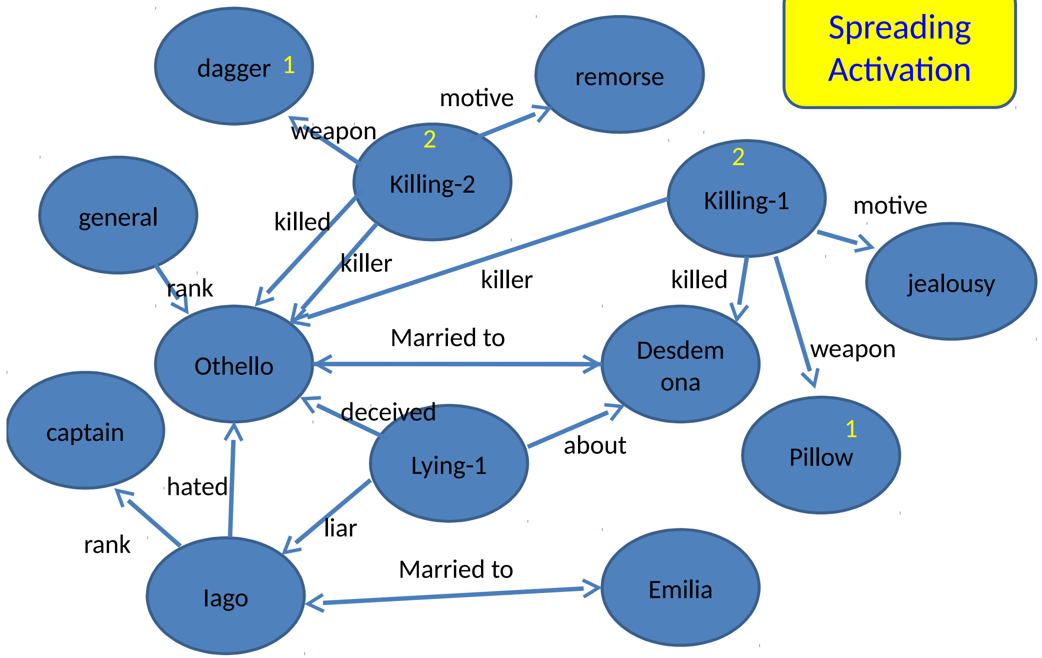
What do the pillow and the dagger have in common?

Spreading Activation



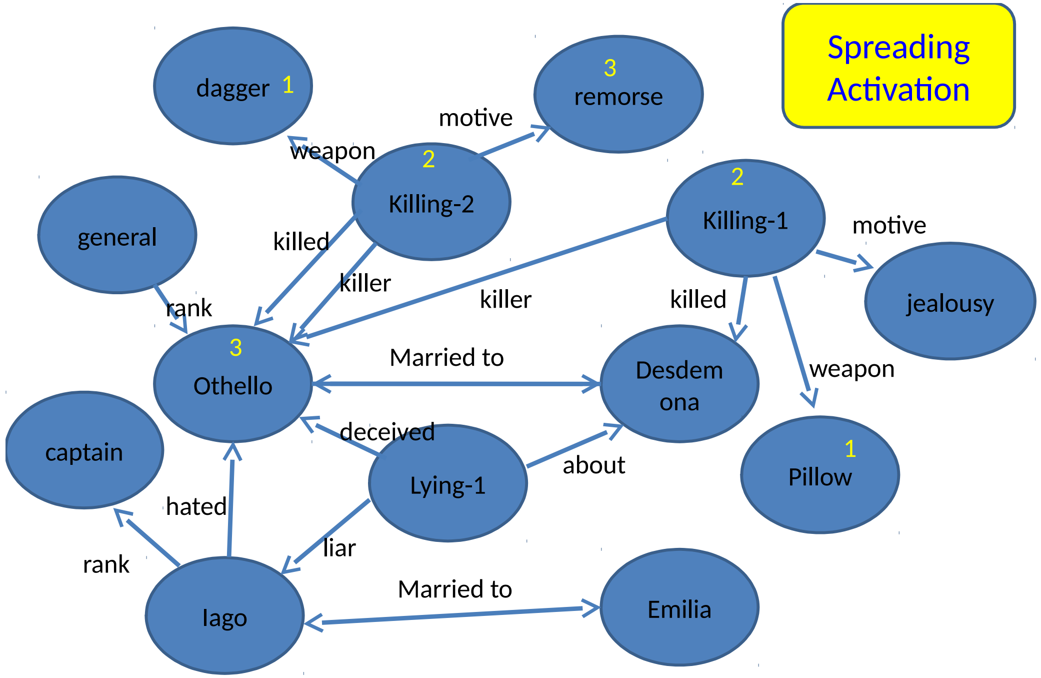
What do the pillow and the dagger have in common?

Spreading Activation



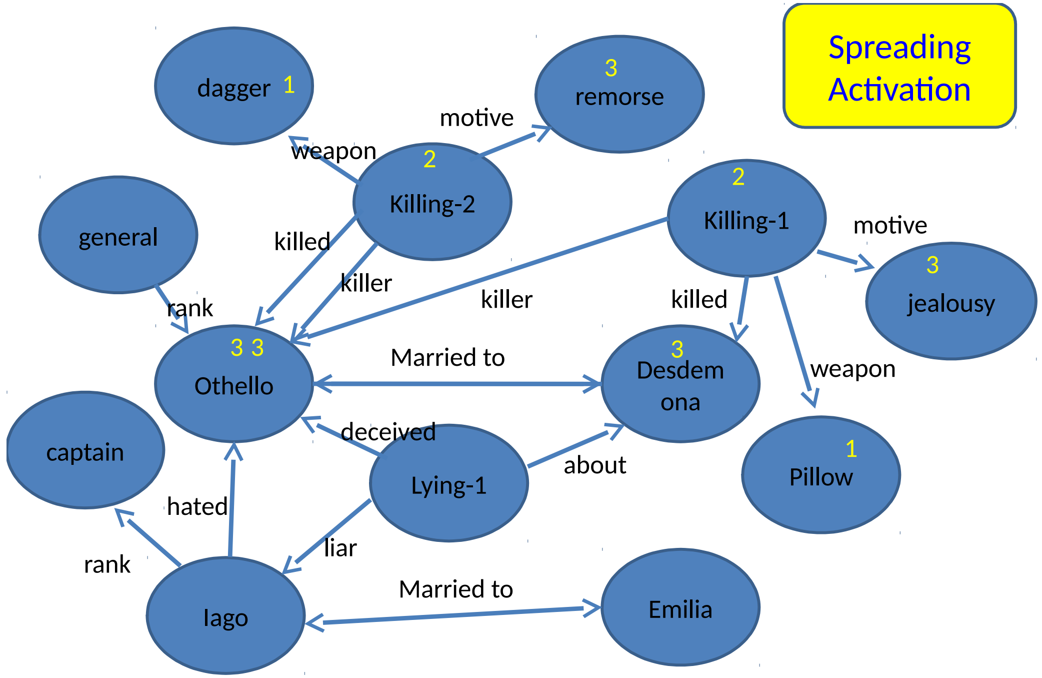
What do the pillow and the dagger have in common?

Spreading Activation



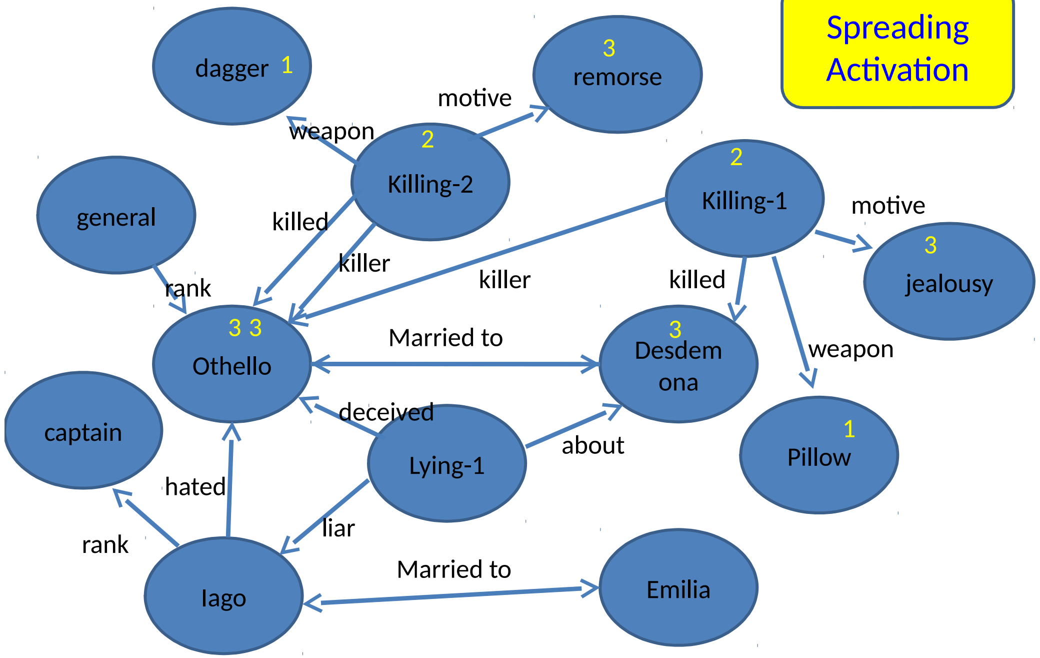
What do the pillow and the dagger have in common?

Spreading Activation



What do the pillow and the dagger have in common?

Spreading Activation



What do the pillow and the dagger have in common?

Weapons used by Othello in killings

Using Rules

```
IF (?X is-a killing) AND (?X killed ?Y) THEN
   REMOVE (?Y alive T) AND
   ADD (?Y alive F).
IF create(killing, ?X, ?Y) THEN
   execute(?X.weapon) AND
   execute(?X.motive) AND
   put(?Y.alive,F).
```

- Or we can use clauses for Prolog

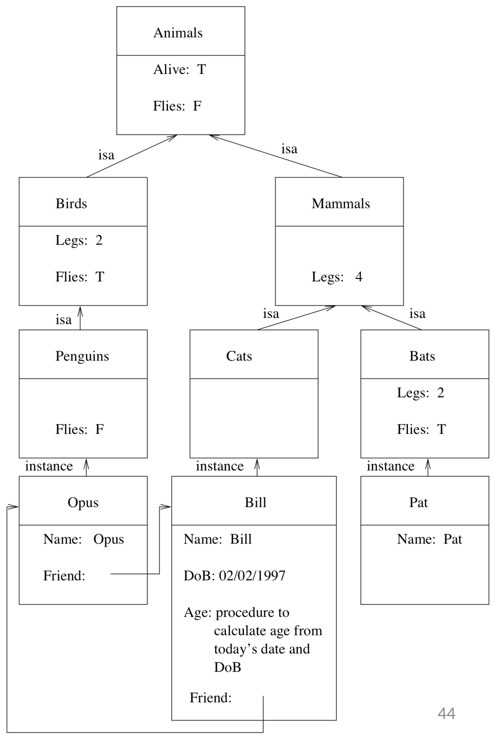
```
alive(X,false):-killing(_,X,_,_).
```


Frames

- Development of semantic nets
- Desire to exploit the powerful mechanism of inheritance
- Observation: things of a given type participate in the same set of relationships
- A lot of information is available by default – it is the exceptions that are interesting

Frames

- Frames - semantic net with *properties and methods*
  - Devised by Marvin Minsky, 1974.
- Incorporates certain valuable human thinking characteristics:
  - Expectations, assumptions, stereotypes, exceptions.
- The essence of this form of knowledge is that we represent the typical case *and exceptions*, *rather* than give *definitions*.
- Hierarchical structure, similar to class hierarchies.



Problems with Frames & Semantic Nets

- Useful for representing certain sorts of knowledge
  - e.g., inheritance
- But node and edge types can be *ad hoc*.
  - no clear meaning, or *semantics*.
- Inheritance reasoning is very easy, but more general reasoning is difficult to define
  - often special purpose.

Developments

- Many of the ideas of frames are now expressed in ontologies (see next lecture)
- Frame system + procedures for retrieving and manipulating knowledge = Object System
- AI research influenced the development of Object Oriented Programming, which has become a standard paradigm
  - good example of how AI contributes to mainstream computing

Agents

- Agents can be seen as a development from OO programming:
  - Agents don't wait for messages: they proactively poll the environment to find new information.
  - Agents decide whether to respond to messages.
  - The elements of proactivity and autonomy make them part of AI.

Summary

- Semantic networks were a popular method of structuring information
- In recent years people have attempted to be more principled and formal
  - Simply working on special cases and limited domains is no longer enough
  - Next we will consider these developments in the context of ontologies and logic-based approaches
- Structured objects developed into OO programming, now a conventional technique

- Next time
  - Expert systems and ontologies