

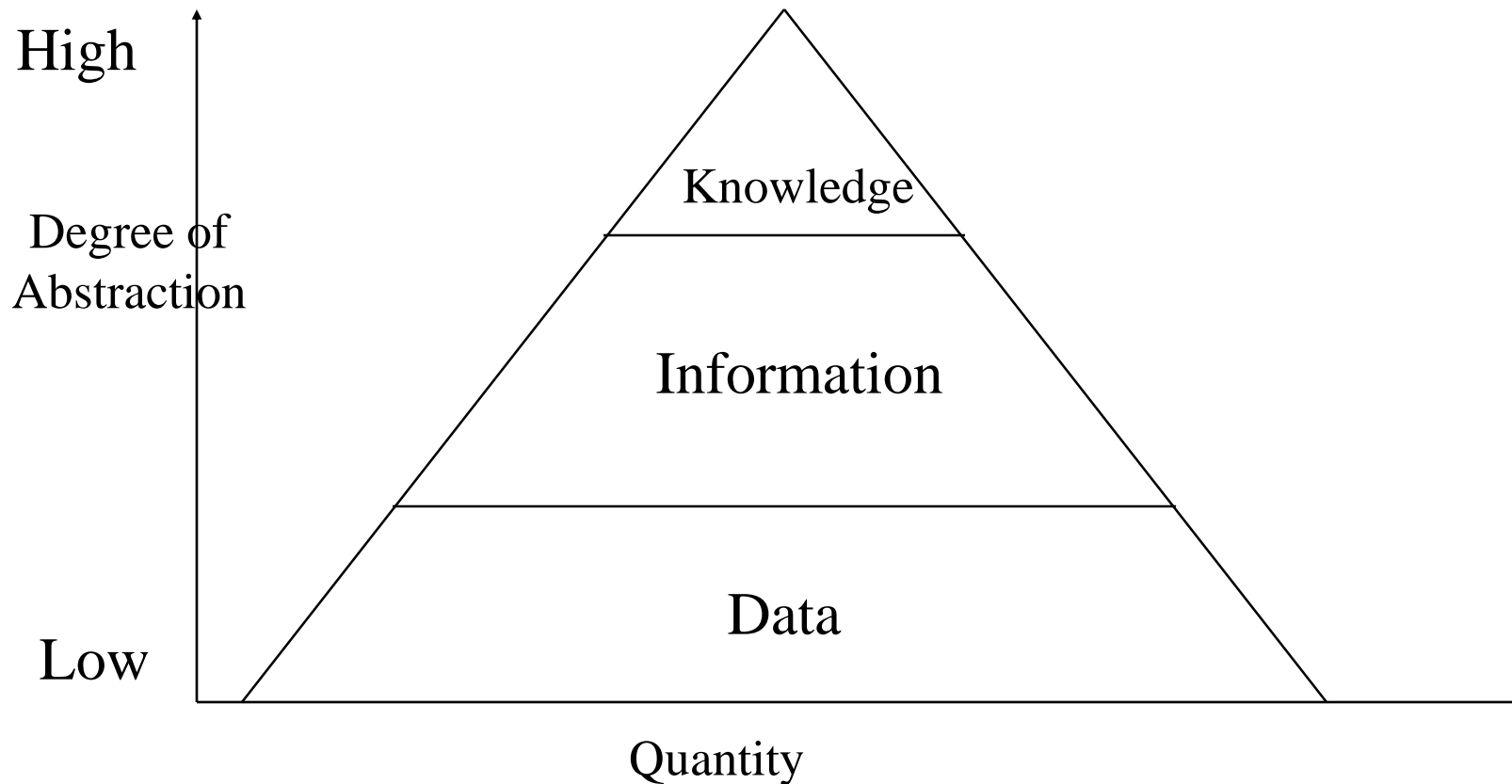


Knowledge Representation

Ass. Prof. Dr. Asaad Sabah Hadi

Knowledge

- *Knowledge* is a collection of specialized facts, procedures and judgment rules



Knowledge Sources

- Documented (books, manuals, etc.)
- Undocumented (in people's minds)
 - From people, from machines
- Knowledge Acquisition from Databases
- Knowledge Acquisition Via the Internet

Major Categories of Knowledge

- Declarative Knowledge
- Procedural Knowledge
- Metaknowledge

Declarative Knowledge

Descriptive Representation of Knowledge

- Expressed in a factual statement
- Shallow
- Important in the initial stage of knowledge acquisition

Procedural Knowledge

- **Considers the manner in which things work under different sets of circumstances**
 - **Includes step-by-step sequences and how-to types of instructions**
 - **May also include explanations**
 - **Involves automatic response to stimuli**
 - **May tell how to use declarative knowledge and how to make inferences**

- **Descriptive knowledge** relates to a specific object. Includes information about the meaning, roles, environment, resources, activities, associations and outcomes of the object
- **Procedural knowledge** relates to the procedures employed in the problem-solving process

Metaknowledge

Knowledge about Knowledge

In ES(Expert System),
Metaknowledge refers to
knowledge about the operation of
knowledge-based systems
Its reasoning capabilities

Knowledge Modeling

- The *knowledge model* views knowledge acquisition as the construction of a model of problem-solving behavior— a model in terms of knowledge instead of representations
- Can *reuse models* across applications

Knowledge Representation

- **Logical representation** – first order predicate calculus, Prolog, declarative knowledge
- **Procedural representation** – a set of instructions for solving a problem, such as a production system
- **Network representation** – knowledge is in a graph structure, such as conceptual dependency and conceptual graphs..
- **Structured representation** – an extension of networks, such as scripts or frames .

Knowledge Representation

**Once acquired, knowledge
must be organized for use**

Introduction

- A good knowledge representation *naturally* represents the problem domain
- An unintelligible knowledge representation is wrong
- Most artificial intelligence systems consist of:
 - Knowledge Base
 - Inference Mechanism (Engine)

- **Knowledge Base**

- **Forms the system's intelligence source**
 - **Inference mechanism uses to reason and draw conclusions**

- **Inference mechanism: Examines the knowledge base to answer questions, solve problems or make decisions within the domain**

- **Many knowledge representation schemes**
 - **Can be programmed and stored in memory**
 - **Are designed for use in reasoning**

- **Major knowledge representation schemas:**
 - **Production rules**
 - **Frames**

Representation in Logic and Other Schemas

- **General form of any logical process**
- **Inputs (Premises)**
- **Premises used by the logical process to create the output, consisting of conclusions (inferences)**
- **Facts known true can be used to derive new facts that are true**

- ***Symbolic logic:*** System of rules and procedures that permits the drawing of inferences from various premises

- **Basic Forms of Computational Logic**
 - Propositional logic (or propositional calculus)
 - Predicate logic (or predicate calculus)

Propositional Logic

- A proposition is a statement that is either true or false
- Once known, it becomes a premise that can be used to derive new propositions or inferences
- Rules are used to determine the truth (T) or falsity (F) of the new proposition

- Symbols represent propositions, premises or conclusions

Statement-1:

A = The mail carrier comes Monday through Friday.

Statement-2:

B = Today is Sunday.

Conclusion:

C = The mail carrier will not come today.

- Propositional logic: limited in representing real-world knowledge

Predicate Calculus

- ❑ Predicate logic breaks a statement down into component parts, an object, object characteristic or some object assertion
- ❑ Predicate calculus uses variables and functions of variables in a symbolic logic statement
- ❑ Predicate calculus is the basis for Prolog (PROgramming in LOGic)
- ❑ Prolog Statement Examples
 - comes_on(mail_carrier, monday).
 - likes(jay, chocolate).

(Note - the period “.” is part of the statement)

Lists

Written Series of Related Items

- Normally used to represent hierarchical knowledge where objects are grouped, categorized or graded according to
 - Rank or
 - Relationship

Decision Tables (Induction Table)

**Knowledge Organized in a Spreadsheet
Format**

- **Attribute List**
- **Conclusion List**
- **Different attribute configurations are matched against the conclusion**

Decision Trees

- Related to tables
- Similar to decision trees in decision theory
- Can simplify the knowledge acquisition process
- Knowledge diagramming - very natural

O-A-V Triplet

- **Objects, Attributes and Values**
- **O-A-V Triplet**
- **Objects may be physical or conceptual**
- **Attributes are the characteristics of the objects**
- **Values are specific measures of the attributes**

Representative O-A-V Items

Object	Attributes	Values
House	Bedrooms	2, 3, 4, etc.
House	Color	Green, white, brown, etc.
Admission to a university	Grade-point average	3.0, 3.5, 3.7, etc.
Inventory control	Level of inventory	14, 20, 30, etc.
Bedroom	Size	9 X 10, 10 X 12, etc.

Default Logic

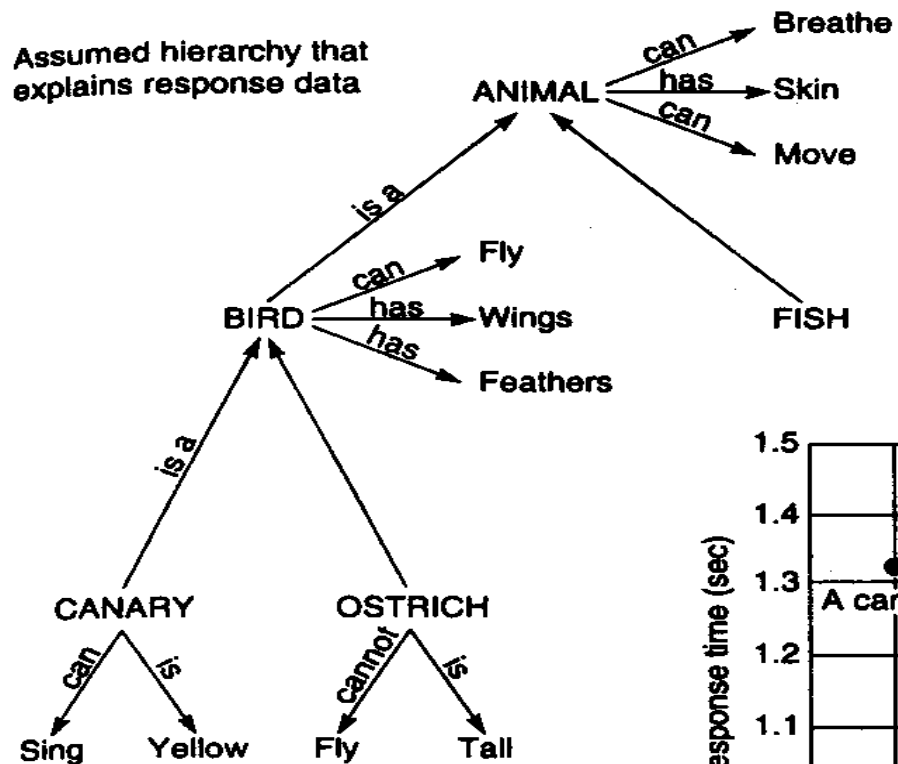
- Deals with uncertainties
- Incomplete information

Knowledge Maps

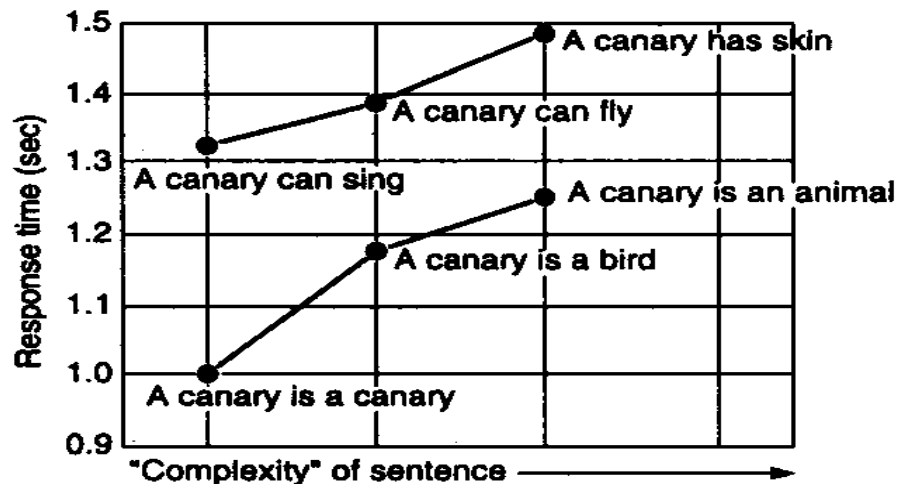
- **Visual representation**
- **Cognitive maps**

Semantic Networks

- Semantics nets were introduced by Quillian in the late 1960s for representing knowledge as a network of associations



- By following links, simple questions can be answered
- Studies with human recall supported this model



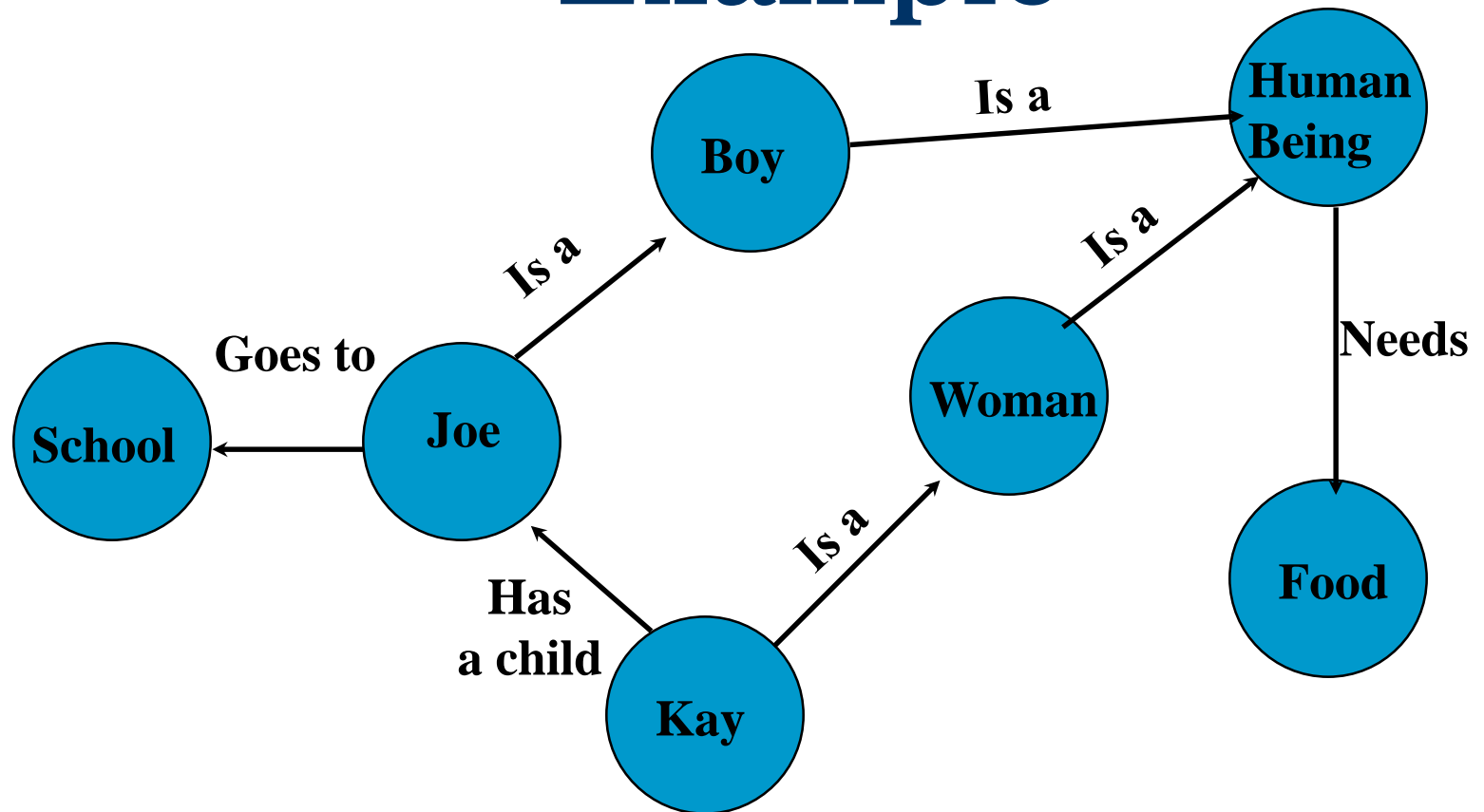
Semantic Networks

- **Graphic Depiction of Knowledge**
- **Nodes and Links Showing Hierarchical Relationships Between Objects**
- **Nodes: Objects**
- **Arcs: Relationships**
 - is-a
 - has-a

- Semantic networks can show *inheritance*
- *Semantic Nets* - visual representation of relationships
- Can be combined with other representation methods

Semantic Network

Example



Conceptual Graphs

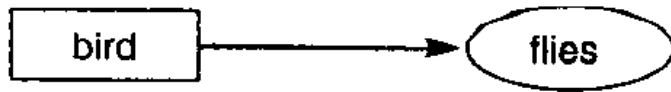
□ Graph Structure

- Finite, connected, bipartite
- Arcs are not labeled
- Conceptual relation nodes are introduced between concepts
- The bipartite nature of the graph means concepts can only link to conceptual relations and vice versa
- In drawings, concepts are shown in boxes and conceptual relations in ellipses

- Concepts may be concrete (dog, child, etc.) or abstract (love, beauty, etc.)

Arity of Relations

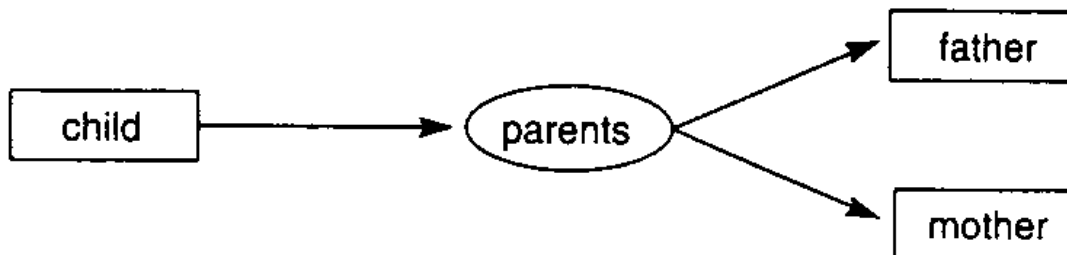
- Examples of 1-ary, 2-ary, and 3-ary relations



Flies is a 1-ary relation.

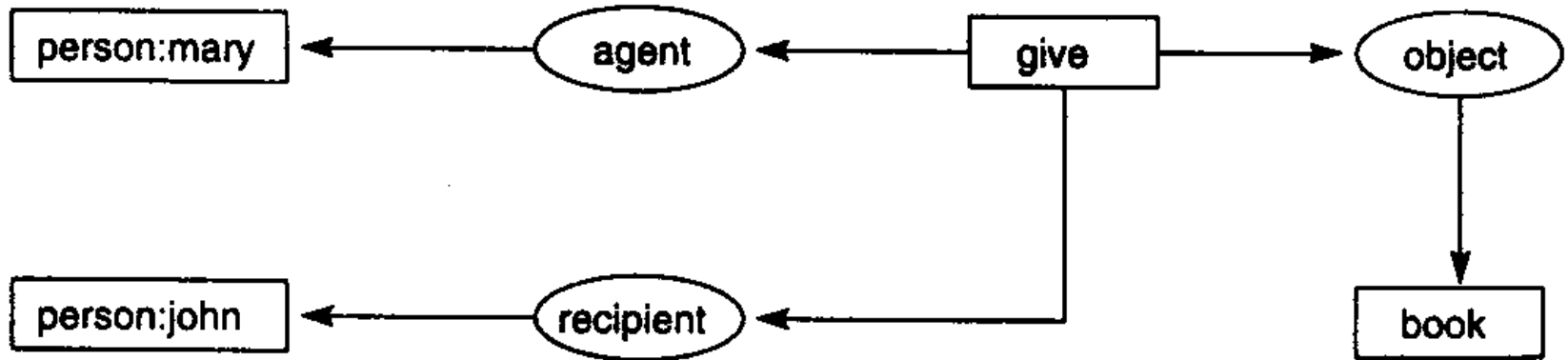


Color is a 2-ary relation.



Parents is a 3-ary relation.

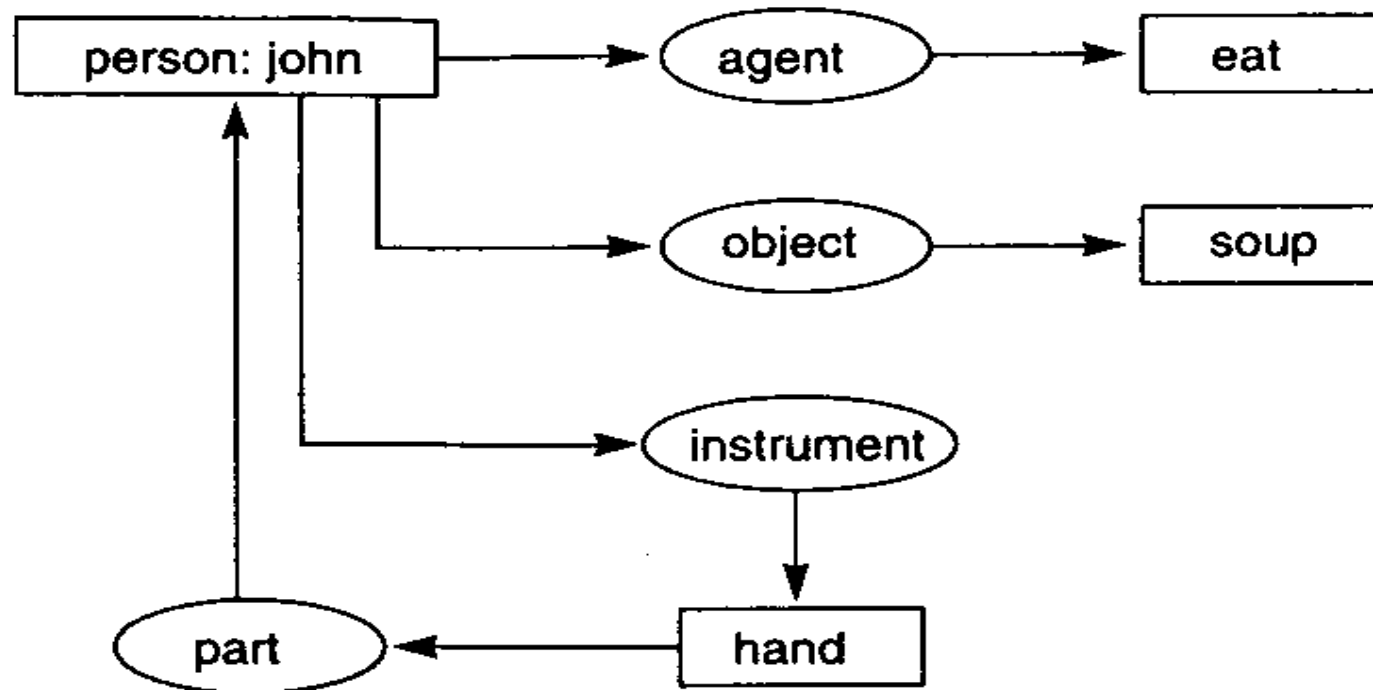
Graph of a Sentence



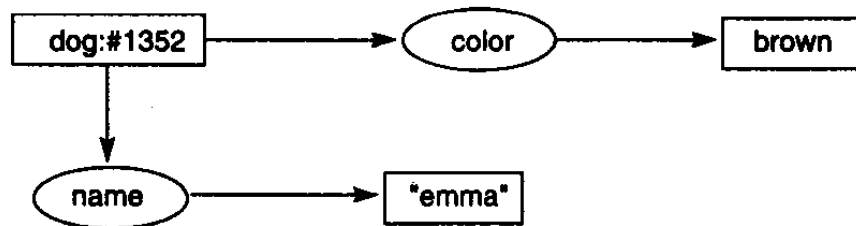
- As in conceptual dependency, the verb plays a central role in the structure
- The verb “give” in this sentence has an agent, an object, and a recipient

Group Work

- What does the following conceptual graph represent



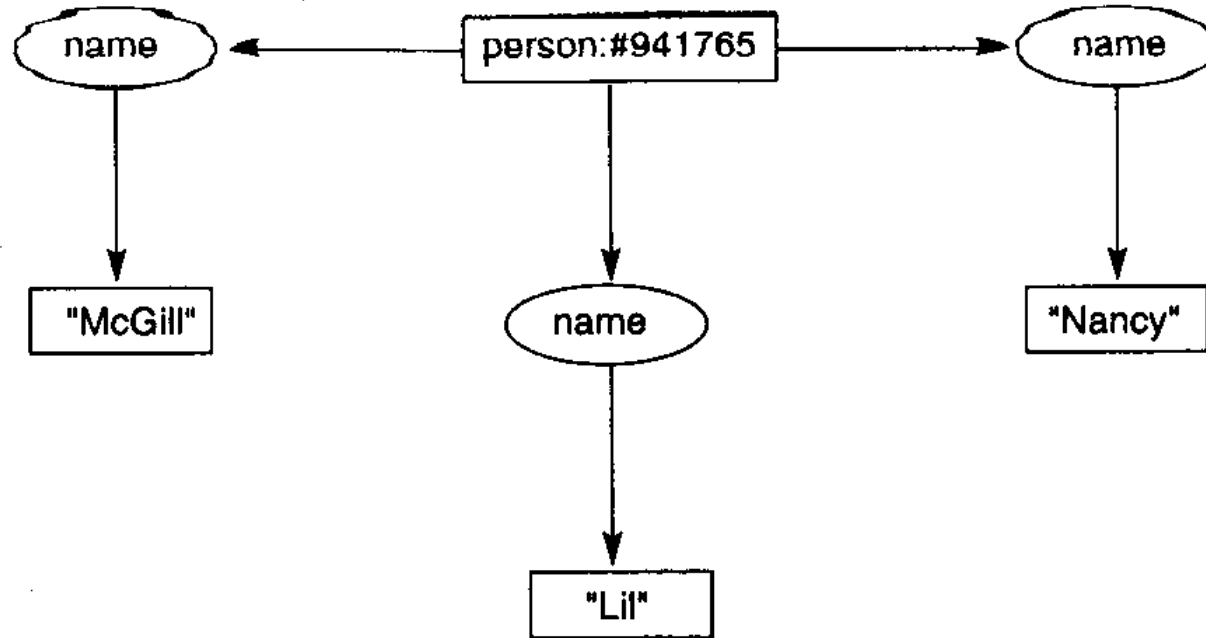
Types and Individuals



- In the first case, the type is dog and the individual is “emma”
- A specific but unnamed dog is given a unique number (#)
- An alternative representation is to use a dog specified by a # and add a conceptual relation for a name

Three Names

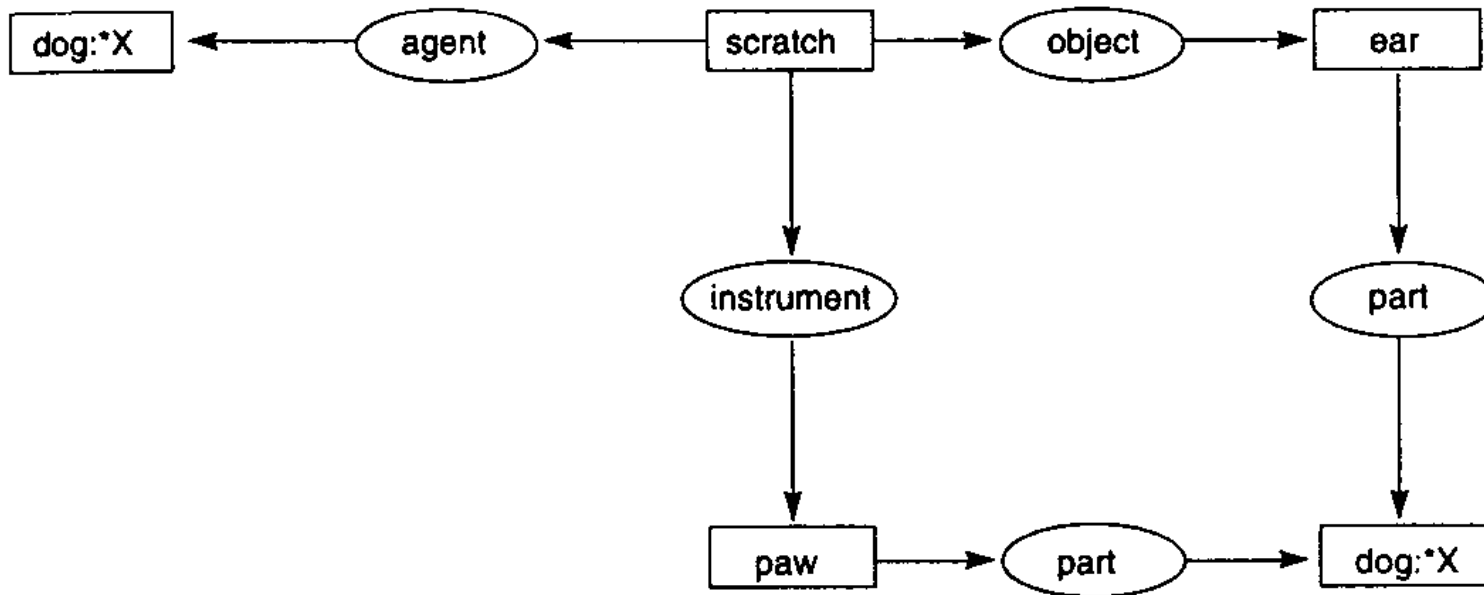
- “Her name was McGill and she called herself Lil, but everyone knew her as Nancy” (song lyric)



- Who was the artist? What was the name of the song?

Itchy Dog

- What is the English sentence for this structure?



- If the same, unspecified individual is present in two or more nodes, a variable can be introduced that may eventually be bound to the same value

Inheritance

- Inheritance is a form of generalization
- Generalization does not guarantee that the resultant graph is true even if the original graphs are true



A conceptual graph

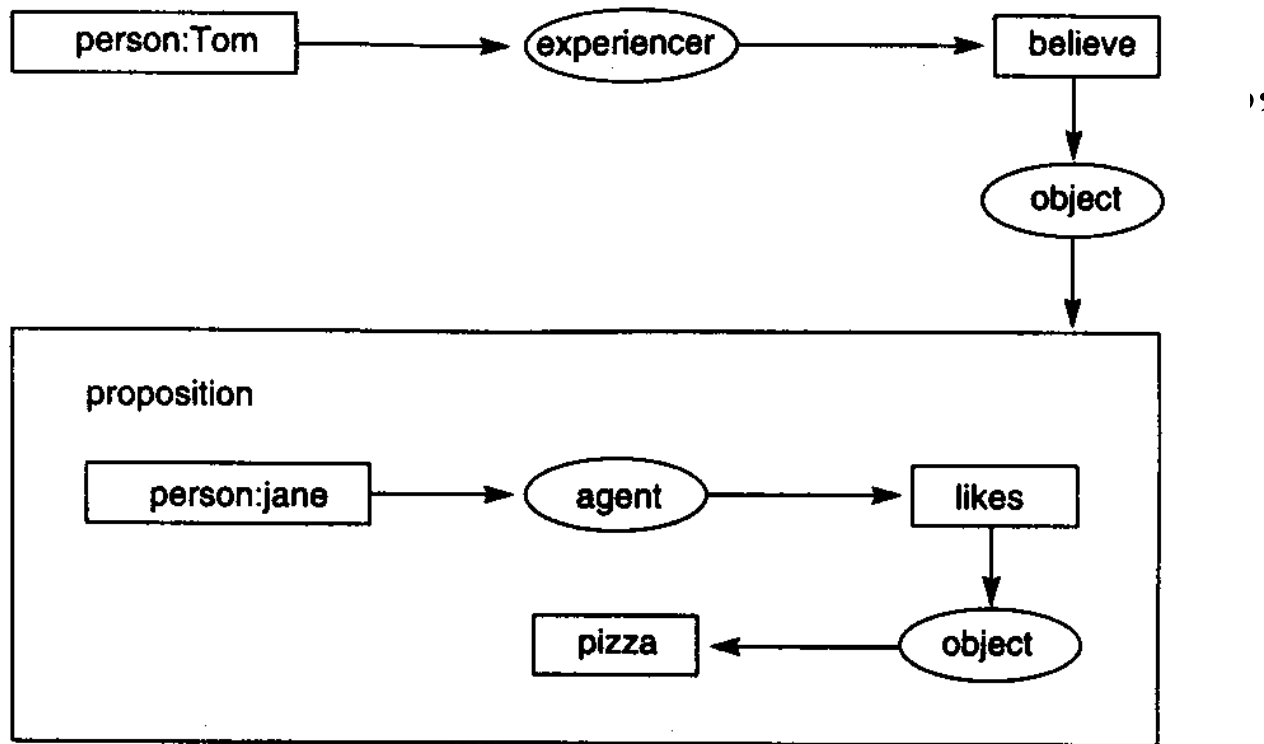


Inheritance of a property by a subclass



Inheritance of a property by an individual

Propositional Nodes



- The verb believes takes a propositional node as its object

Production Rules

- **Condition-Action Pairs**
 - **IF this condition (or premise or antecedent) occurs,**
 - **THEN some action (or result, or conclusion, or consequence) will (or should) occur**

- **IF the stop light is red AND you have stopped, THEN a right turn is OK**

- Each production rule in a knowledge base represents an autonomous chunk of expertise
- When combined and fed to the inference engine, the set of rules behaves synergistically
- Rules can be viewed as a simulation of the cognitive behavior of human experts
- Rules represent a model of actual human behavior

Forms of Rules

- IF premise, THEN conclusion
 - IF your income is high, THEN your chance of being audited by the IRS is high

- Conclusion, IF premise
 - Your chance of being audited is high, IF your income is high

□ Inclusion of ELSE

- IF your income is high, OR your deductions are unusual, THEN your chance of being audited by the IRS is high, OR ELSE your chance of being audited is low

□ More Complex Rules

- IF credit rating is high AND salary is more than \$30,000, OR assets are more than \$75,000, AND pay history is not "poor," THEN approve a loan up to \$10,000, and list the loan in category "B."
- Action part may have more information: THEN "approve the loan" and "refer to an agent"

Knowledge and Inference Rules

Common Types of Rules

- Knowledge rules, or declarative rules, state all the facts and relationships about a problem
- Inference rules, or procedural rules, advise on how to solve a problem, given that certain facts are known
- Inference rules contain rules about rules (metarules)
- Knowledge rules are stored in the knowledge base
- Inference rules become part of the inference engine

Advantages of Rules

- Easy to understand (natural form of knowledge)
- Easy to derive inference and explanations
- Easy to modify and maintain
- Easy to combine with uncertainty
- Rules are frequently independent

Limitations of Rules

- ❑ **Complex knowledge requires many rules**
- ❑ **Builders like rules (hammer syndrome)**
- ❑ **Search limitations in systems with many rules**

Characteristics of Rule Representation

	First Part	Second Part
Names	Premise → Antecedent → Situation → IF →	Conclusion Consequence Action THEN
Nature	Conditions, similar to declarative knowledge	Resolutions, similar to procedural knowledge
Size	Can have many IFs	Usually one conclusion
Statements	AND statements	All conditions must be true for a conclusion to be true
	OR statements	If any of the OR statement is true, the conclusion is true

Frames

Definitions and Overview

- **Frame:** Data structure that includes all the knowledge about a particular object
 - Knowledge organized in a hierarchy for diagnosis of knowledge independence
 - Form of *object-oriented programming* for AI and ES.
-
- Each Frame Describes One Object
 - Special Terminology

Frames (2)

Here is the essence of the frame theory: When one encounters a new situation (or makes a substantial change in one's view of a problem) one selects from memory a structure called a "frame." This is a remembered framework to be adapted to fit reality by changing details as necessary (Minsky 1975).

- Frames, like scripts, are used in stereotypical situations
 - When a new situation is encountered, a frame may be recalled from memory
 - The frame provides a complete framework
 - Details may vary from situation to situation
 - Frames can provide default values
 - Frames can be arranged in a hierarchy

Frames and NLP

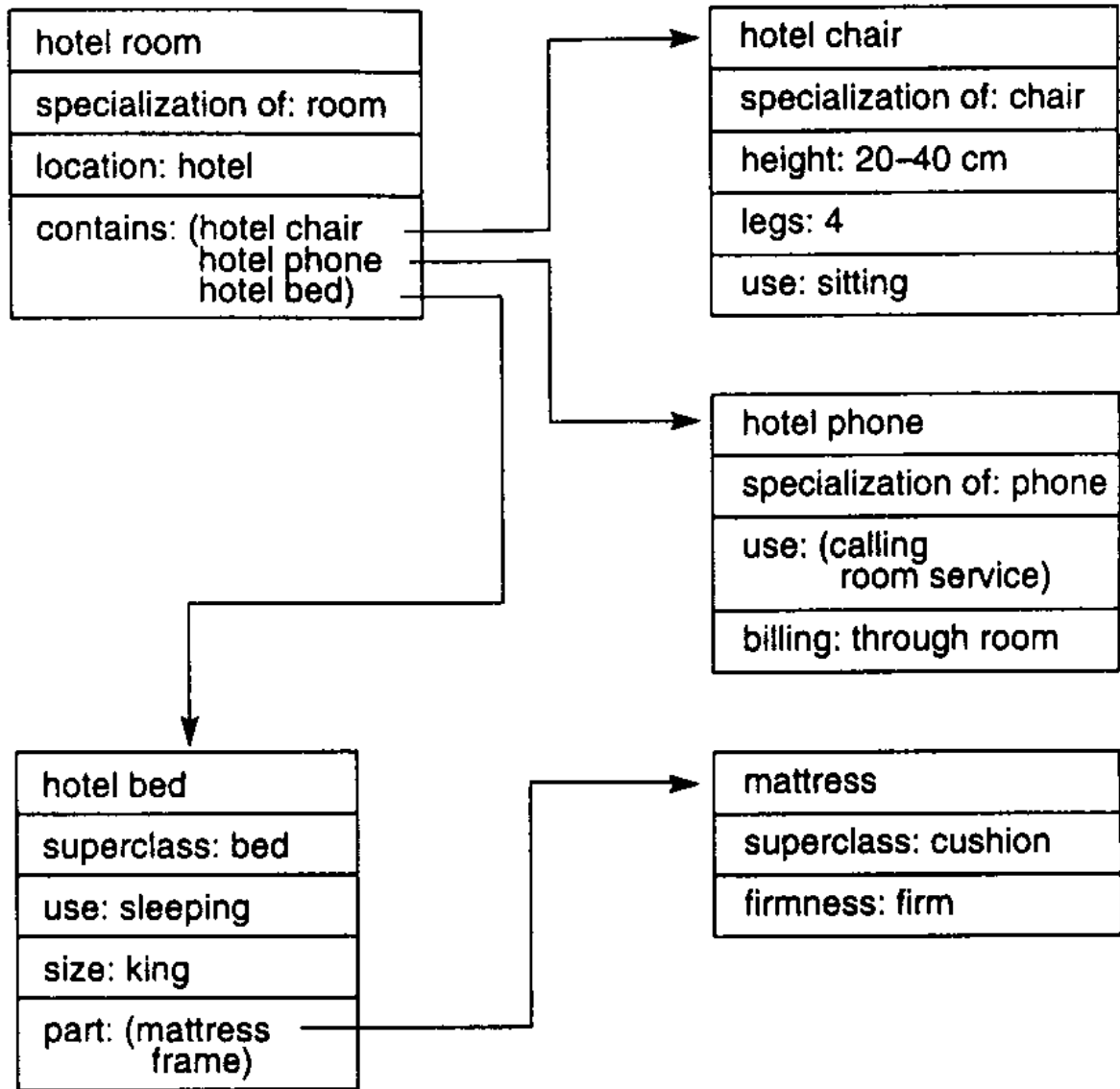
- Much of the inference required for NLP involves making assumptions about what is typically true about a situation
- Encode this stereotypical information in a **frame**
- Looks like themes, but on a higher level of abstraction

Frame Terminology

Default	Instantiation
Demon	Master frame
Facet	Object
Hierarchy of frames	Range
If added	Slot
If needed	Value (entry)
Instance of	

Components of a Frame

1. *Frame identification information.*
2. *Relationship of this frame to other frames.* The “hotel phone” might be a special instance of “phone,” which in turn might be an instance of a “communication device.”
3. *Descriptors of requirements for frame match.* A chair, for instance, has its seat between 20 and 40 cm from the floor, its back higher than 60 cm, etc. These requirements may be used to determine when new objects fit the stereotyp defined by the frame.
4. *Procedural information on use of the structure described.* An important feature of frames is the ability to attach procedural code to a slot.
5. *Frame default information.* These are slot values that are taken to be true when no evidence to the contrary has been found. For instance, chairs have four legs, telephones are pushbutton, or hotel beds are made by the staff.
6. *New instance information.* Many frame slots may be left unspecified until given a value for a particular instance or when they are needed for some aspect of problem solving. For example, the color of the bedspread may be left unspecified in the definition of bed.



Frame Capabilities

Ability to clearly document information about a domain model; for example, a plant's machines and their associated attributes

Related ability to constrain the allowable values that an attribute can take on

Modularity of information, permitting ease of system expansion and maintenance

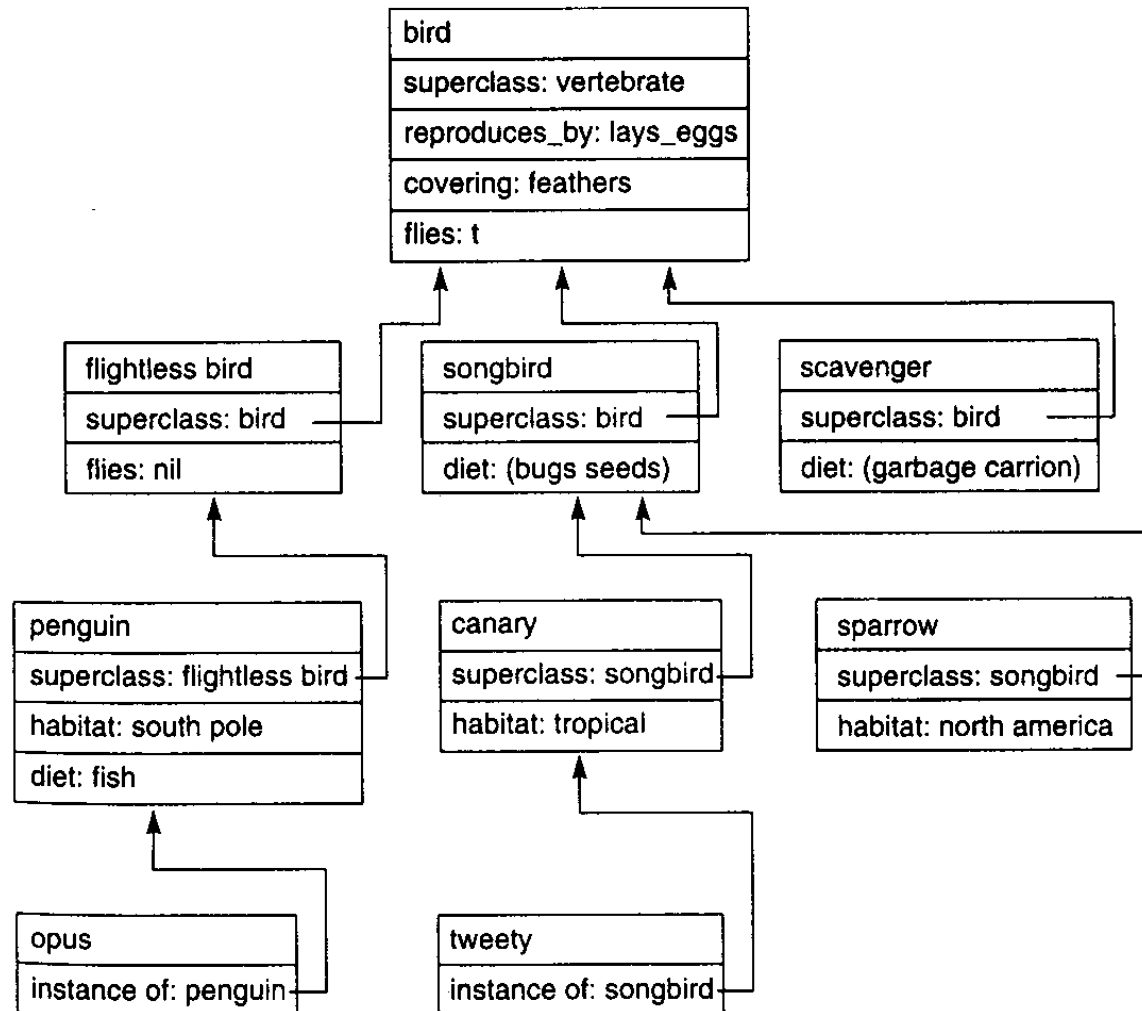
More readable and consistent syntax for referencing domain objects in the rules

Platform for building a graphic interface with object graphics

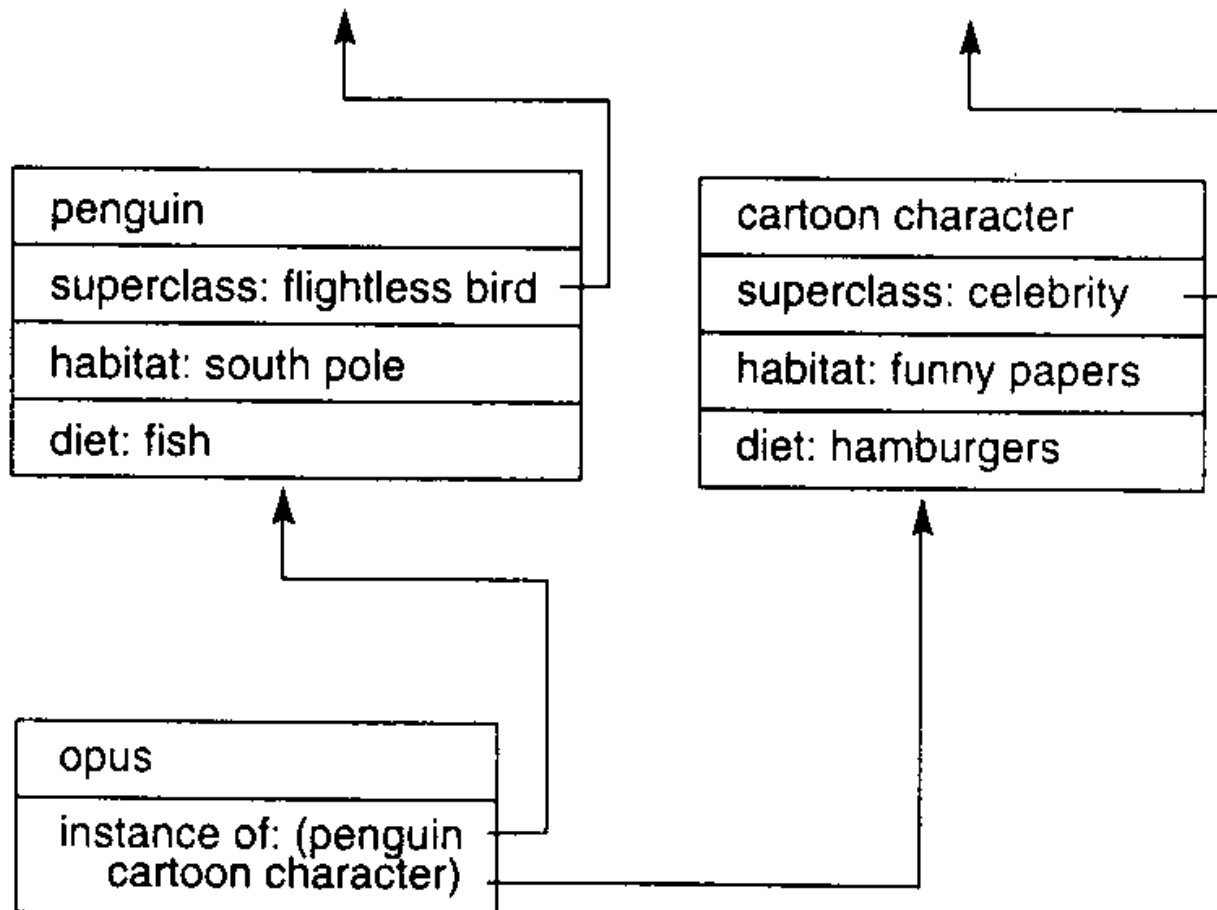
Mechanism that will allow us to restrict the scope of facts considered during forward or backward chaining

Access to a mechanism that supports the inheritance of information down a class hierarchy

Inheritance - 1

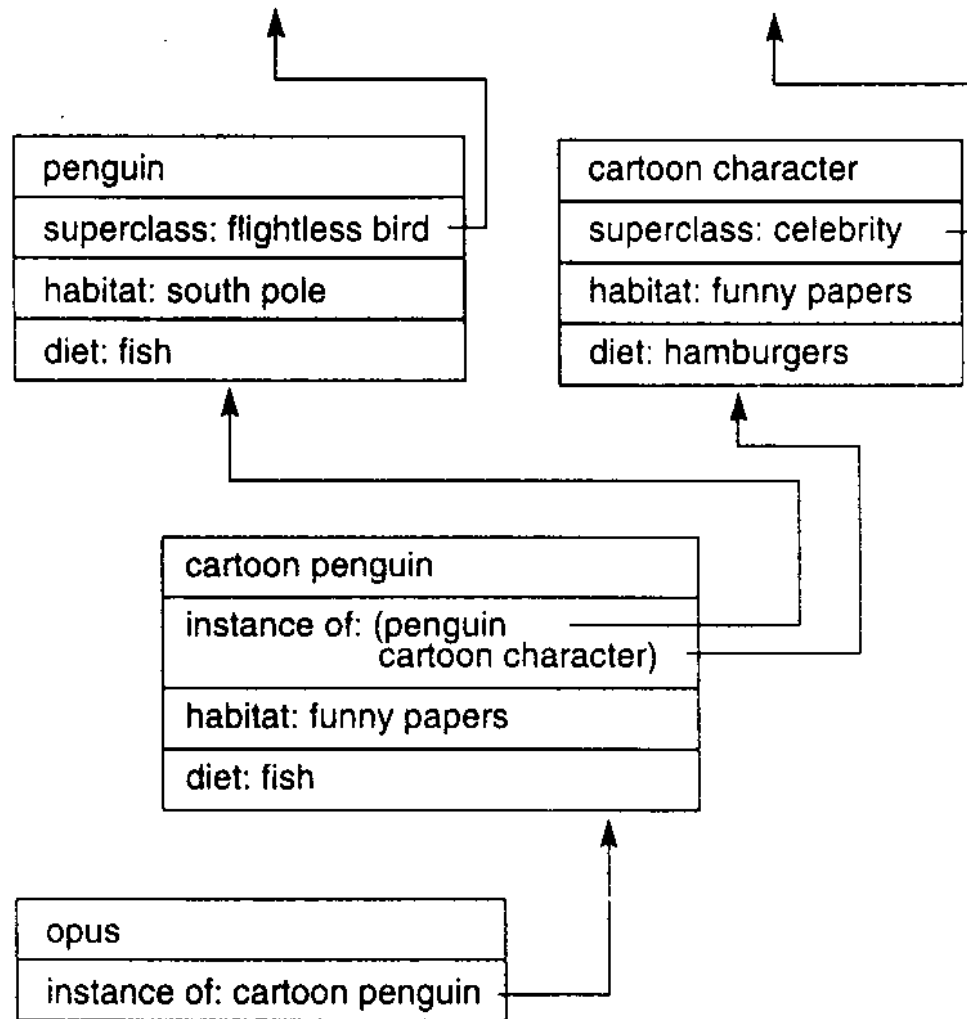


Inheritance - 2



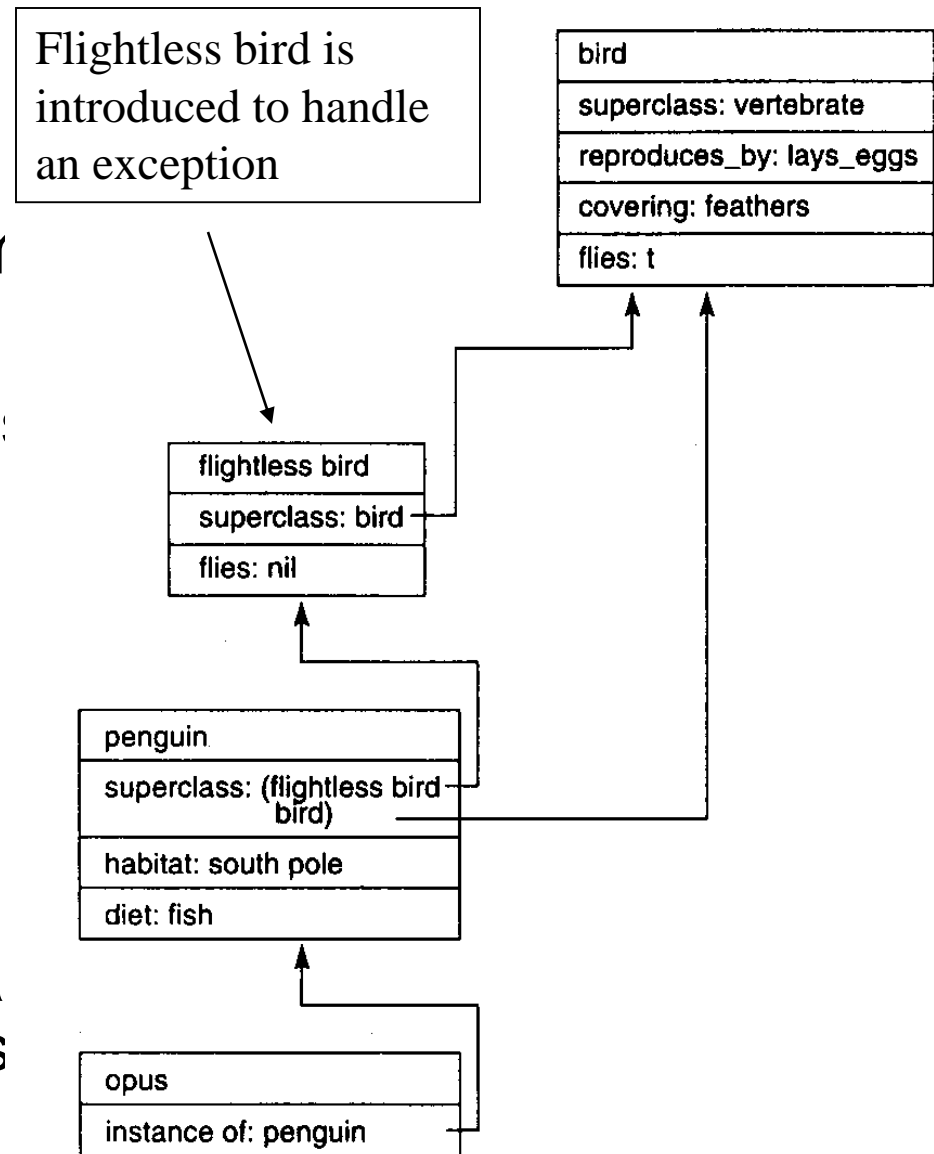
Inheritance - 3

- A new class to resolve ambiguity



Transitivity of Subclasses

- Fixing one problem
 - Penguins don't fly
 - Introduce a flightless bird class
- Results in other problems
 - If subclasses are transitive, we infer penguin is a bird
 - This adds an extra link that introduces problems with multiple inheritance



A Summary of Frames

- Frames organize knowledge into structures
- Frames are recalled on an as needed basis
- Procedures can be attached to frames where the procedure may process one of the slots in the frame in some way, such as detecting changes
- Frames support class inheritance
- Frames can supply default knowledge
- In essence, frames extended semantic networks by providing organization and structure

Scripts

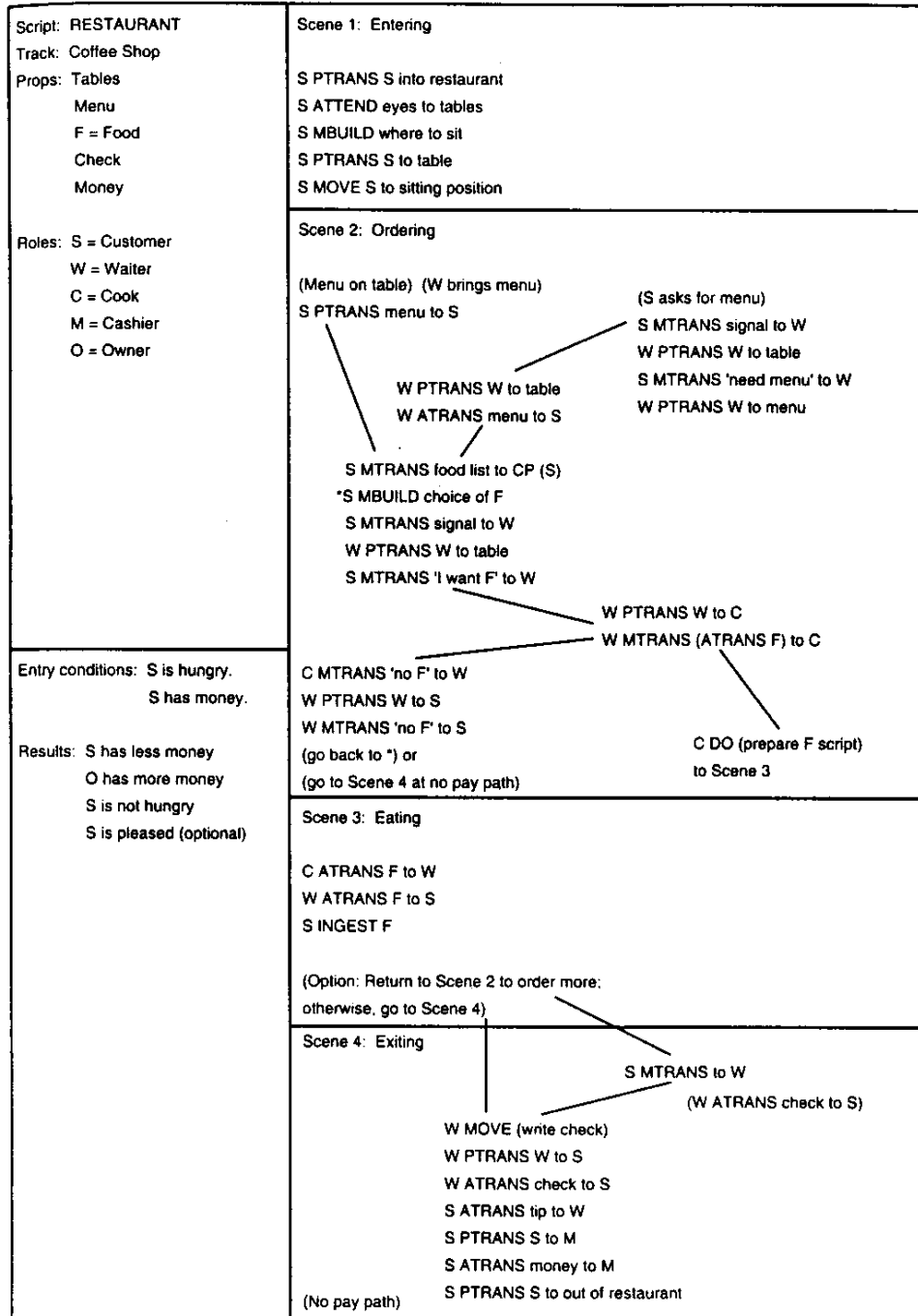
- A means of identifying common situations in a particular domain
- A means of generating expectations
- We precompile information, rather than recomputing from first principles.
- **Elements include**
 - **Entry Conditions:** These must be satisfied before events in the script can occur
 - **Props:** Slots representing objects involved in events
 - **Roles:** Persons involved in the events
 - **Tracks:** Variations on the script. Different tracks may share components of the same script.
 - **Scenes:** The sequence of *events* that occur. *Events* are represented in *conceptual dependency* form.

Scripts

Travel by plane:

- Roles: Actor, Clerk, Source, Destination, Airport, Ticket, Money, Airplane
- Constraints: Person(Actor), Value(Money, Price(Ticket)), . . .
- Preconditions: Owns(Actor, Money), At(Actor, Source)
- Effects: not(Owns(Actor, Money)), not(At(Actor, Source)), At(Actor, Destination)
- Decomposition:
 - GoTo(Actor, Airport)
 - BuyTicket(Actor, Clerk, Money, Ticket), . . .

The Restaurant Script



The Robbery Script

Script: ROBBERY		<i>Track: Successful Snatch</i>	
<i>Props:</i> G = Gun, L = Loot, B= Bag, C = Get away car.		<i>Roles:</i> R = Robber, M = Cashier, O = Bank Manager, P = Policeman.	
<i>Entry Conditions:</i> R is poor. R is destitute.		<i>Results:</i> R has more money. O is angry. M is in a state of shock. P is shot.	
<i>Scene 1: Getting a gun</i> R PTRANS R into Gun Shop R MBUILD R choice of G R MTRANS choice. R ATRANS buys G (go to scene 2)			
<i>Scene 2 Holding up the bank</i> R PTRANS R into bank R ATTEND eyes M, O and P R MOVE R to M position R GRASP G R MOVE G to point to M R MTRANS "Give me the money or ELSE" to M P MTRANS "Hold it Hands Up" to R R PROPEL shoots G P INGEST bullet from G M ATRANS L to M M ATRANS L puts in bag B M PTRANS exit O ATRANS raises the alarm (go to scene 3)			
<i>Scene 3: The getaway</i> M PTRANS C			

Advantages and Disadvantages

Advantages of Scripts:

- Ability to predict events.
- A single coherent interpretation may be build up from a collection of observations.

Disadvantages:

- Less general than frames.
- May not be suitable to represent all kinds of knowledge.

Issues in Knowledge Representation

- We have examined several ways to represent knowledge
 - Predicate calculus
 - Procedural, as in an expert system
 - Network, as in semantic nets, conceptual dependency and conceptual graphs
 - Structured, as in frames and scripts
- Particular problems arise with each type, we examine problems with more recent types
 - Hierarchies and inheritance
 - Exceptions

Considerations for Evaluating a Knowledge Representation

- Naturalness, uniformity and understandability
- Degree to which knowledge is explicit (declarative) or embedded in procedural code
- Modularity and flexibility of the knowledge base
- Efficiency of knowledge retrieval and the heuristic power of the inference procedure

- ❑ **No single knowledge representation method is ideally suited by itself for all tasks**
- ❑ **Multiple knowledge representations: each tailored to a different subtask**
- ❑ **Production Rules and Frames works well in practice**
- ❑ **Object-Oriented Knowledge Representations**
 - **Hypermedia**

Multiple Knowledge Representations

- Rules + Frames
- Others

Knowledge Representation *Must* Support

- Acquiring knowledge
- Retrieving knowledge
- Reasoning



Thanks for Listening