CS4710: Artificial Intelligence
Knowledge Representation

For most AI problems, we must represent complex information. What are some techniques for doing this? What are the pros and cons of each?

Topics

- Overview of Knowledge Representation
- What Info do we Need to Represent?
- Techniques for doing so…etc.

Physical Symbol Systems and the beginning of AI

Physical Symbol Systems

- Newell and Simon (left)
- Axiom (maybe):
  - ***Computers and minds*** are both examples of ***physical symbol systems***
- ***Symbol***: A meaningful pattern that can be manipulated
- ***Symbol System***: Creates, modifies, destroys, etc. symbols
- ***Physical***: Exists directly in the world (controls physical muscles, etc.)

Physical Symbol System Hypothesis

Hypothesis:

A physical symbol system has the necessary and sufficient means for general intelligent action

Thoughts on this?

Discuss the reading with those around you. Make a list of five "thoughtful observations" regarding the reading and physical symbol systems.

Representation

- AI agents deal with knowledge (data)
  - Facts (believe & observe knowledge)
  - Procedures (how to knowledge)
  - Meaning (relate & define knowledge)

- Right representation is crucial
  - Early realisation in AI
  - Wrong choice can lead to project failure
  - Active research area

Choosing a Representation

- For certain problem solving techniques
  - ‘Best’ representation already known
  - Often a requirement of the technique
  - Or a requirement of the programming language (e.g. Prolog)

- Examples
  - First order theorem proving… first order logic
  - Inductive logic programming… logic programs
  - Neural networks learning… neural networks

Knowledge Representation

- **Goals:**
  - To represent the knowledge in the world in a way that allows for an AI to "reason"

- **What to Represent?**
  - **Objects**
    - -- Facts about objects in our world domain. *e.g.* Guitars have strings, trumpets are brass instruments.
  - **Events**
    - -- Actions that occur in our world. *e.g.* Steve Vai played the guitar in Frank Zappa's Band.

Knowledge Representation

- **Goals:**
  - To represent the knowledge in the world in a way that allows for an AI to “reason”

- **Performance**
  - -- A behavior like *playing the guitar* involves knowledge about how to do things.

- **Meta-knowledge**
  - -- knowledge about what we know.

Thus in solving problems in AI we must represent knowledge and there are two entities to deal with:

- **Facts**
  - -- truths about the real world and what we represent. This can be regarded as the *knowledge level*
  - *E.g, Floryan loves ice cream*
- **Representation of the facts**
  - which we manipulate. This can be regarded as the *symbol level* since we usually define the representation in terms of symbols that can be manipulated by programs.

Two Levels of Knowledge

- **Knowledge Level:**
  - At which facts are described
  - *All dogs have tails*
  - *Fido is a dog*
  - *Etc…*

- **Symbol Level:**
  - *hasatail(x) means 'x' has a tail*
  - *D is a dog* as *dog(D)* We could then infer that all dogs have tails with:
  - *dog(Spot)*
  - Vx : *dog(x)*→ *hasatail(x)* We can then deduce:
    - *hasatail(Spot)*

  - ** Doesn't have to be first-order logic as above. This is just an example*

Properties for Knowledge Representation Systems

- **Representational Adequacy**
  - -- the ability to represent the required knowledge;
- **Inferential Adequacy**
  - - the ability to manipulate the knowledge represented to produce new knowledge corresponding to that inferred from the original;
- **Inferential Efficiency**
  - - the ability to direct the inferential mechanisms into the most productive directions by storing appropriate guides;
- **Acquisitional Efficiency**
  - - the ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

To date no single system optimizes all of the above

- **Representational Adequacy**
  - -- the ability to represent the required knowledge;

- What it means:
  - Need a way to represent ALL types of knowledge.
  - Easy:
    - Spot is a dog, all dogs have tails
  - Medium:
    - Spot does not have a tail (apparent contradiction)
  - A Little Harder:
    - I must go grocery shopping between 8 and 2 on Tuesdays.

- **Inferential Adequacy**
  - - the ability to manipulate the knowledge represented to produce new knowledge corresponding to that inferred from the original;
- What it means:
  - In formal logic, this means applying logical reasoning to infer new things.
    - Spot is a dog
    - All dogs have tails
    - Thus, Spot has a tail
- If we can represent knowledge, but can't use it to infer anything interesting, then our representation is quite poor.

- **Inferential Efficiency**
  - - the ability to direct the inferential mechanisms into the most productive directions by storing appropriate guides;
- What it means:
  - When making complex inferences, system should do so in the most efficient manner possible.
    - Basically an algorithms problem
  - System should be able to choose, if many inferential paths possible, the most important or productive path!

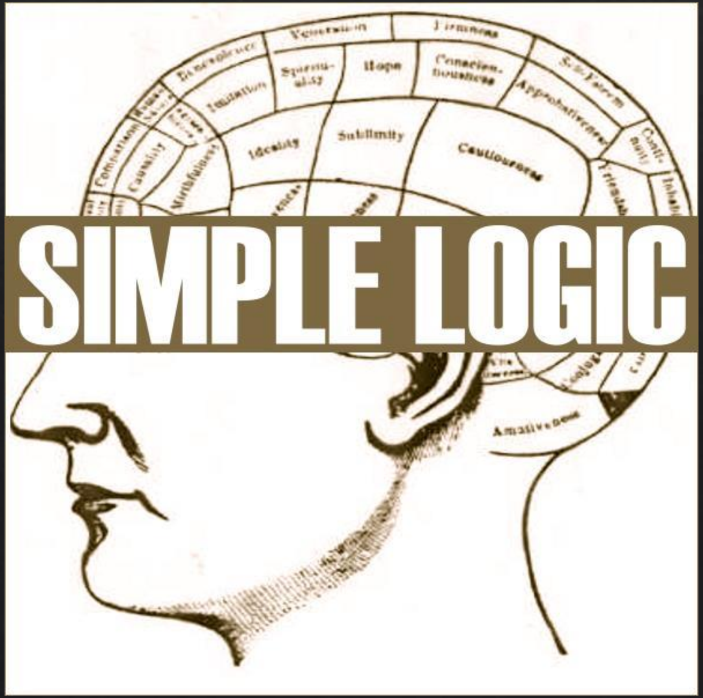
- **Acquisitional Efficiency**
  - - the ability to acquire new knowledge using automatic methods wherever possible rather than reliance on human intervention.

- What it means:
  - Humans should not have to manually teach the system everything. It needs to learn on it’s own as much as possible.

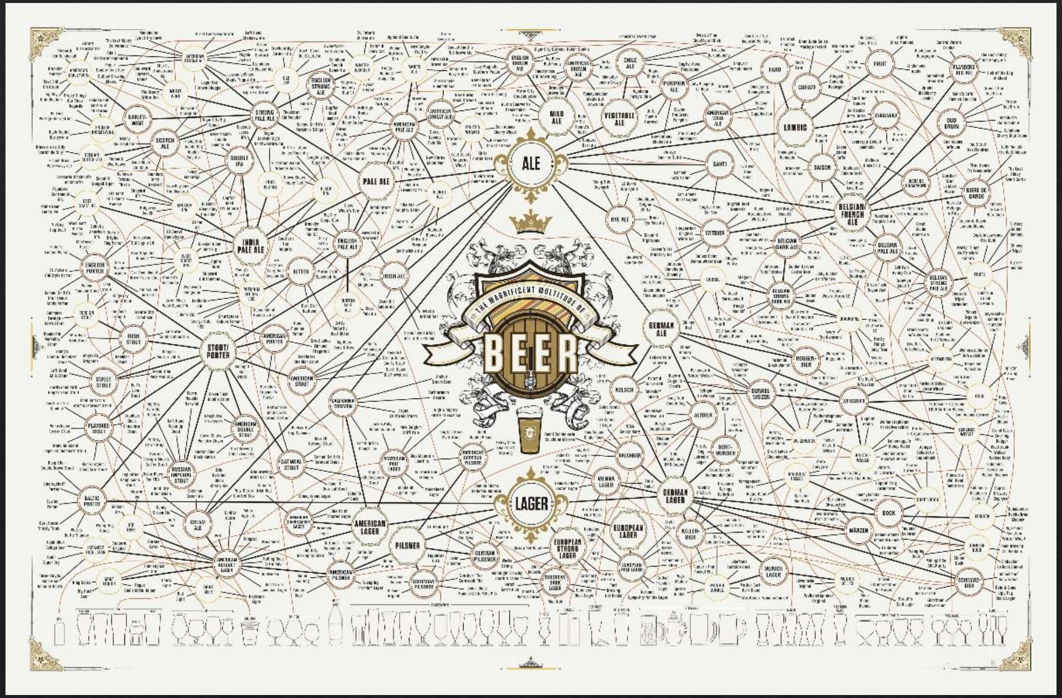
Basic Approaches to Knowledge Representation

Two Schools of Thought

- **Knowledge as Logic:**
  - E.g., Propositional Logic:
  - *"If x is a cardinal, then x is red"*
  - $\forall x(cardinal(x) \rightarrow red(x))$



- **Associationist:**
  - *Idea that knowledge is built via observations and their associations with other observations*
  - *First cited by Plato and Aristotle*
  - *Believe that humans don't actually think logically only*



What is a Logic?

- A **<u>language</u>** with concrete rules
  - ***No ambiguity*** in representation (may be other errors!)
  - Allows unambiguous communication and processing
  - Very unlike natural languages e.g. English
- Many ways to translate between languages
  - A statement can be represented in different logics
  - And perhaps differently in same logic
- **Expressiveness** of a logic
  - How much can we say in this language?
- Not to be confused with logical reasoning
  - Logics are languages, reasoning is a process (may **use** logic)

Syntax and Semantics

- Syntax
  - Rules for constructing legal sentences in the logic
  - Which symbols we can use (English: letters, punctuation)
  - How we are allowed to combine symbols
- Semantics
  - How we interpret (read) sentences in the logic
  - Assigns a meaning to each sentence
- Example: “All lecturers are seven foot tall”
  - A valid sentence (syntax)
  - And we can understand the meaning (semantics)
  - This sentence happens to be false (there is a counterexample)

Propositional Logic

- Syntax
  - Propositions, e.g. “it is wet”
  - Connectives: and, or, not, implies, iff (equivalent)

    $\wedge \quad \vee \quad \neg \quad \rightarrow \quad \leftrightarrow$

  - Brackets, T (true) and F (false)

- Semantics (Classical AKA Boolean)
  - Define how connectives affect truth
    - “P and Q” is true if and only if P is true and Q is true
  - Use **truth tables** to work out the truth of statements

Predicate Logic

- Propositional logic combines atoms
  - An atom contains no propositional connectives
  - Have no structure (today_is_wet, john_likes_apples)

- **Predicates** allow us to talk about objects
  - Properties: is_wet(today)
  - Relations: likes(john, apples)
  - True or false

- In predicate logic each atom is a predicate
  - e.g. first order logic, higher-order logic

First Order Logic

- More expressive logic than propositional
- **Constants** are objects: john, apples
- **Predicates** are properties and relations:
  - likes(john, apples)
- **Functions** transform objects:
  - likes(john, fruit_of(apple_tree))
- **Variables** represent any object: likes(X, apples)
- **Quantifiers** qualify values of variables
  - True for all objects (Universal): $\forall X.$ likes(X, apples)
  - Exists at least one object (Existential): $\exists X.$ likes(X, apples)

Example: FOL Sentence

- What does this mean?

$$\forall X.\big(rose(X) \rightarrow \exists Y.(has(X,Y) \land thorn(Y))\big)$$

Example: FOL Sentence

- “Every rose has a thorn”

$$\forall X.(rose(X) \rightarrow \exists Y.(has(X,Y) \wedge thorn(Y)))$$

- For all X
  - if (X is a rose)
  - then there exists Y
    - (X has Y) and (Y is a thorn)

Example: FOL Sentence

$$\forall X.((is\_mon(X) \vee is\_wed(X)) \rightarrow eat\_meal(me, houseOf(john), X))$$

Example: FOL Sentence

- "On Mondays and Wednesdays I go to John's house for dinner"

$$\forall X.((is\_mon(X) \vee is\_wed(X)) \rightarrow eat\_meal(me, houseOf(john), X))$$

- Note the change from "and" to "or"
  - Translating is problematic

Higher Order Logic

- More expressive than first order
- Functions and predicates are also objects
  - Described by predicates: binary(addition)
  - Transformed by functions: differentiate(square)
  - Can quantify over both
- E.g. define red functions as having zero at 17

$$\forall F.(red(F) \leftrightarrow F(0) = 17)$$

- Much harder to reason with

Production Rules

- Rule set of <condition,action> pairs
  - "if condition then action"

- Match-resolve-act cycle
  - **Match**: Agent checks if each rule's condition holds
  - **Resolve**:
    - Multiple production rules may fire at once (**conflict set**)
    - Agent must choose rule from set (**conflict resolution**)
  - **Act**: If so, rule "fires" and the action is carried out

- Working memory:
  - rule can write knowledge to working memory
  - knowledge may match and fire other rules

Production Rules Example

- IF (at bus stop AND bus arrives) THEN action(get on the bus)
- IF (on bus AND not paid AND have oyster card) THEN action(pay with oyster) AND add(paid)
- IF (on bus AND paid AND empty seat) THEN sit down

- conditions and actions must be clearly defined
  - can easily be expressed in first order logic!

Beyond True and False

- Multi-valued logics
  - More than two truth values
  - e.g., true, false & unknown
  - **Fuzzy logic** uses probabilities, truth value in [0,1]

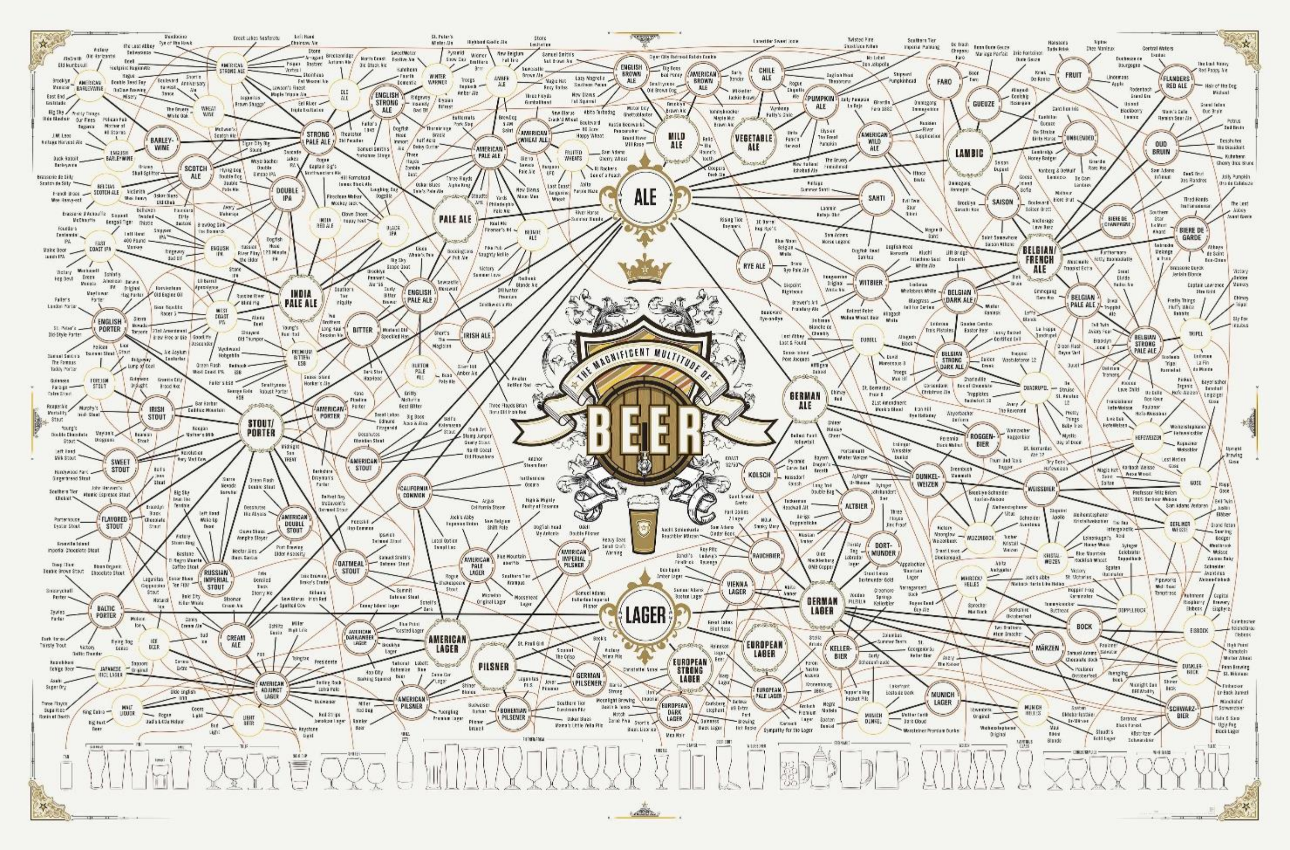
Logic is a Good Representation

- Fairly easy to do the translation when possible
- Branches of mathematics devoted to it
- It enables us to do logical reasoning
  - Tools and techniques come for free
- Basis for programming languages
  - Prolog uses logic programs (a subset of FOL)
  - λProlog based on HOL

Associationism

- **Facts Perceived Directly By Senses:**
  - *Cold / Hot is a feeling I sense*
  - *Hard / Soft*
  - *Etc.*

- **Connections Built Up From This:**
  - *Snow is white (a color I perceive directly)*
  - *Snow is cold*
  - *Snow is soft*
  - *Etc…*

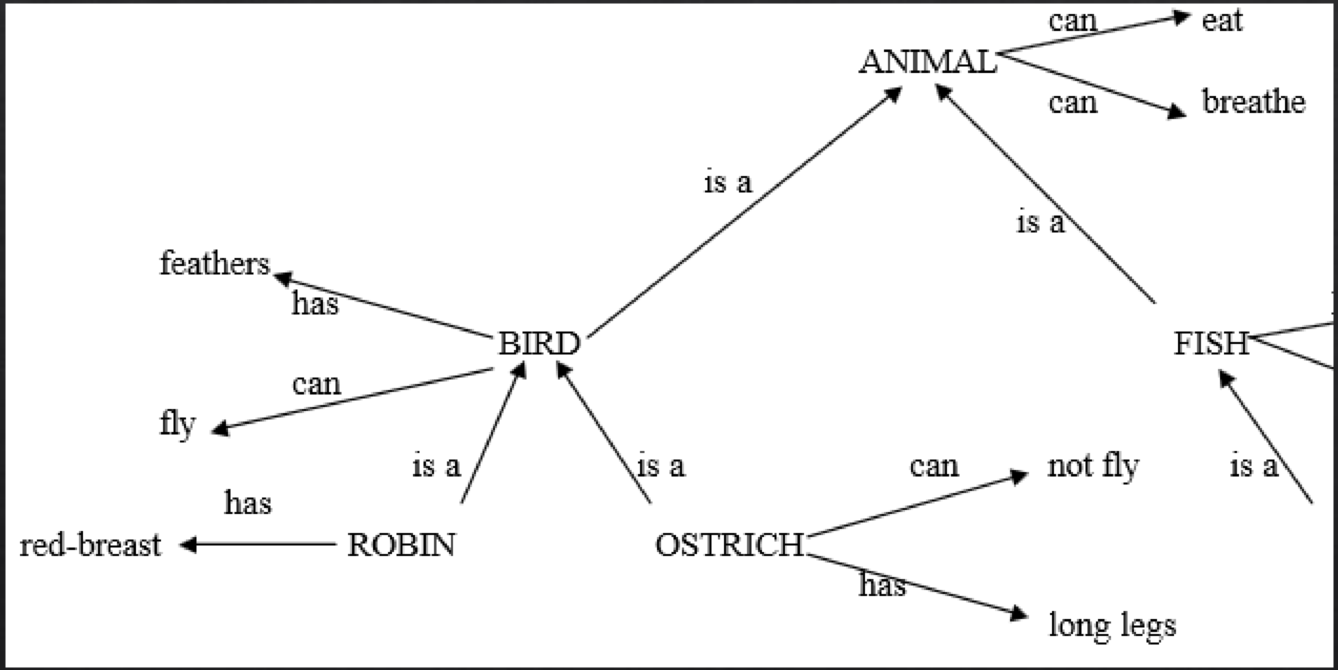


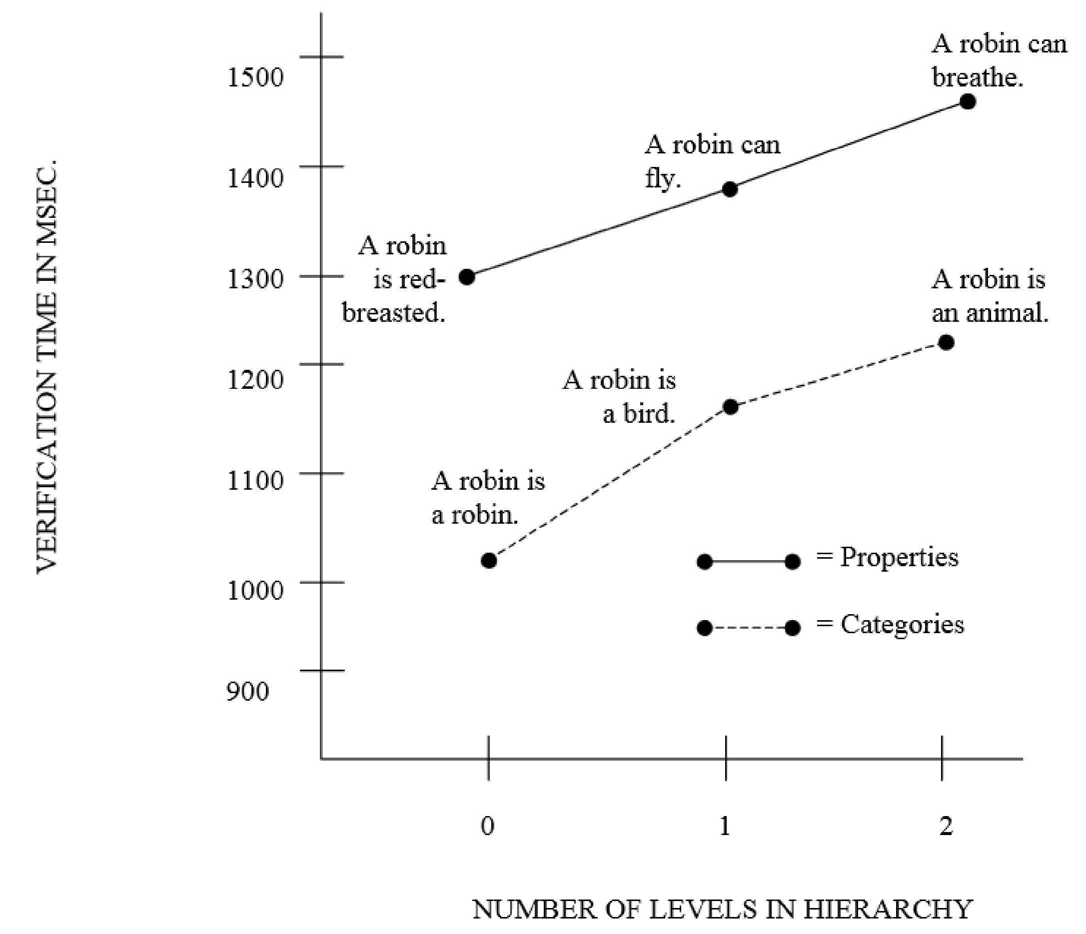
Associationism: Inheritance

- **Consider the two questions:**
  - ***Can a Canary Sing?***
  - ***Can a Canary Fly?***

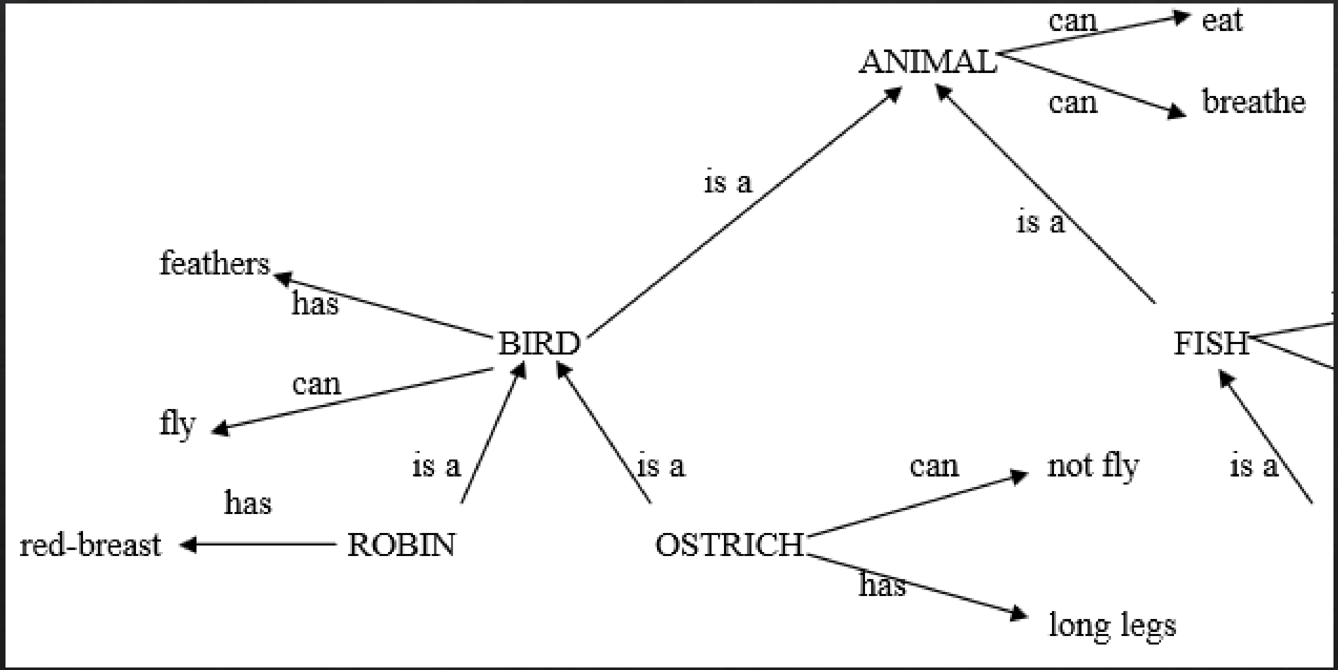
- *Collins and Quillian (1969) showed that human knowledge is hierarchical.*
  - *Can answer first question quickly…why?*
  - *Second question takes longer…why?*

Associationism: Inheritance





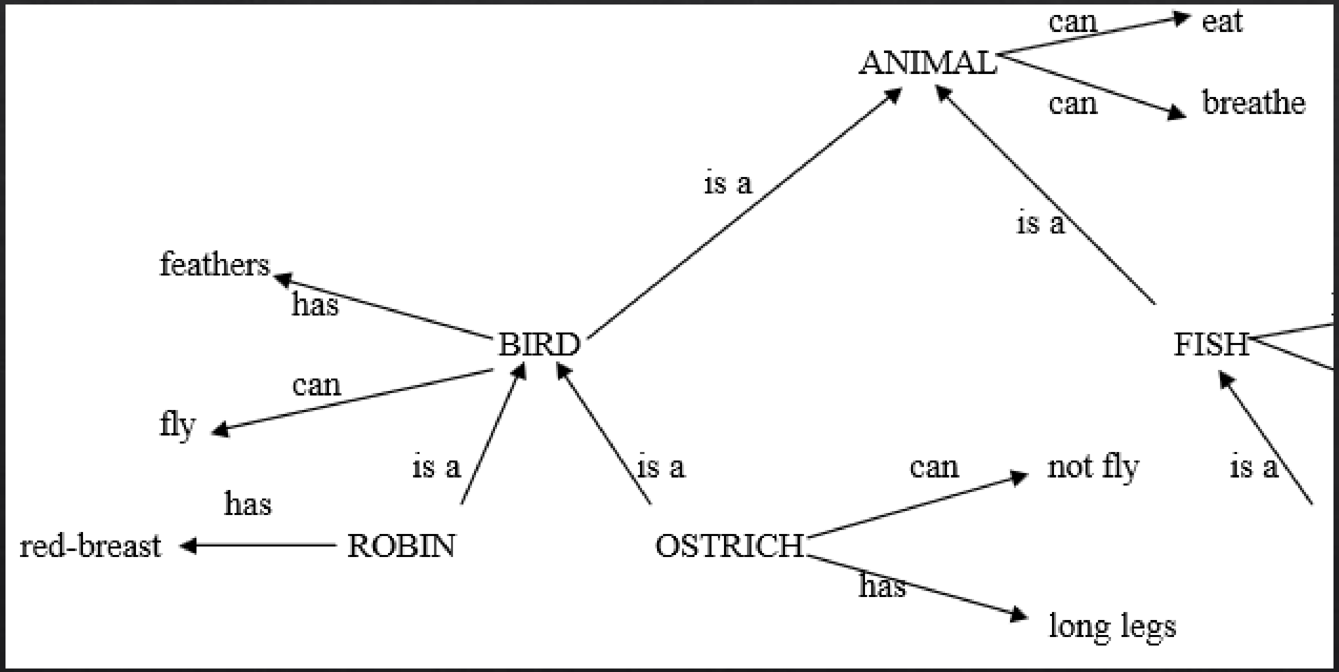
Semantic Net



- **This is called a Semantic Net:**
  - *Objects – Things we observe in the world, can be abstract like 'Animal'*
  - *Relationships – between objects. Note the semantic information on the edges in the graph*

Semantic Net

- How might we acquire new knowledge though?



- *Think about our direct senses with the environment?*

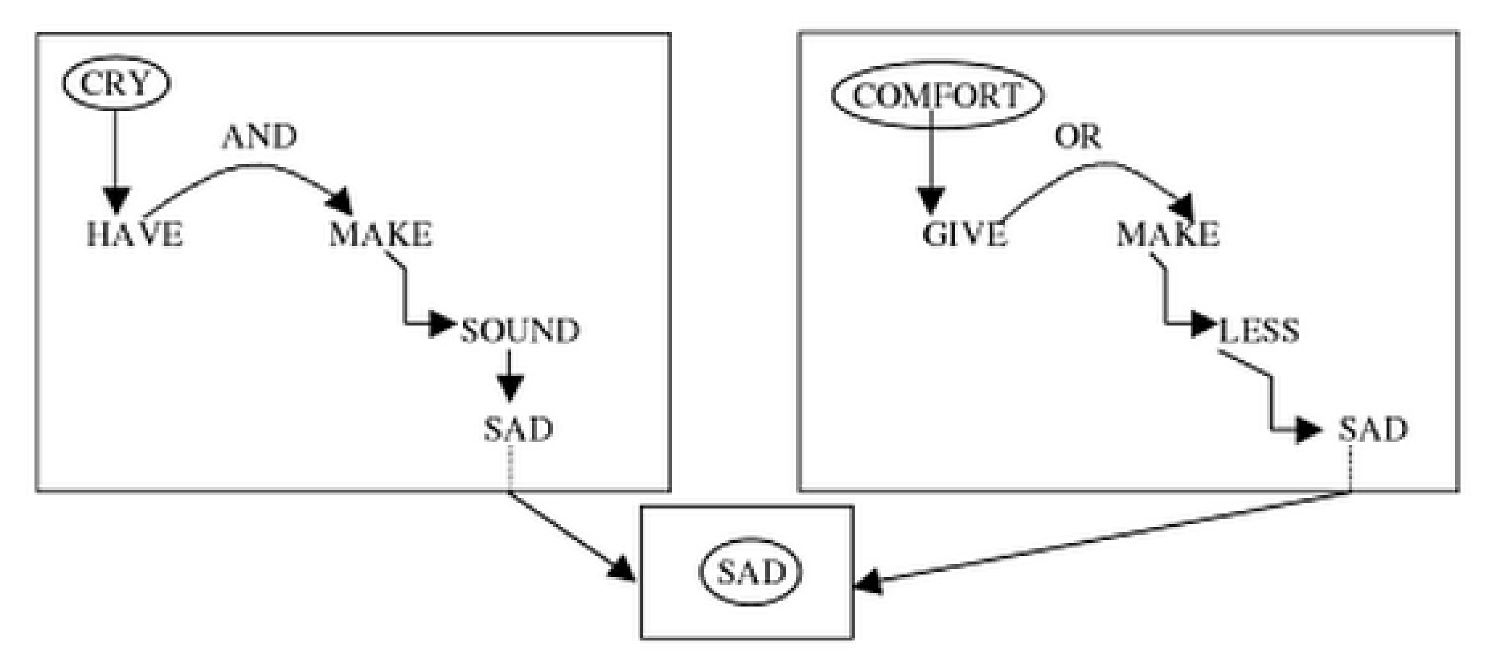
Semantic Networks

- Graphical representation (a graph)
  - Links indicate subset, member, relation, …

- Equivalent to logical statements (usually FOL)
  - Easier to understand than FOL?
  - Specialised SN reasoning algorithms can be faster

- Example: natural language understanding
  - Sentences with same meaning have same graphs
  - e.g. Conceptual Dependency Theory (Schank)

Semantic Net



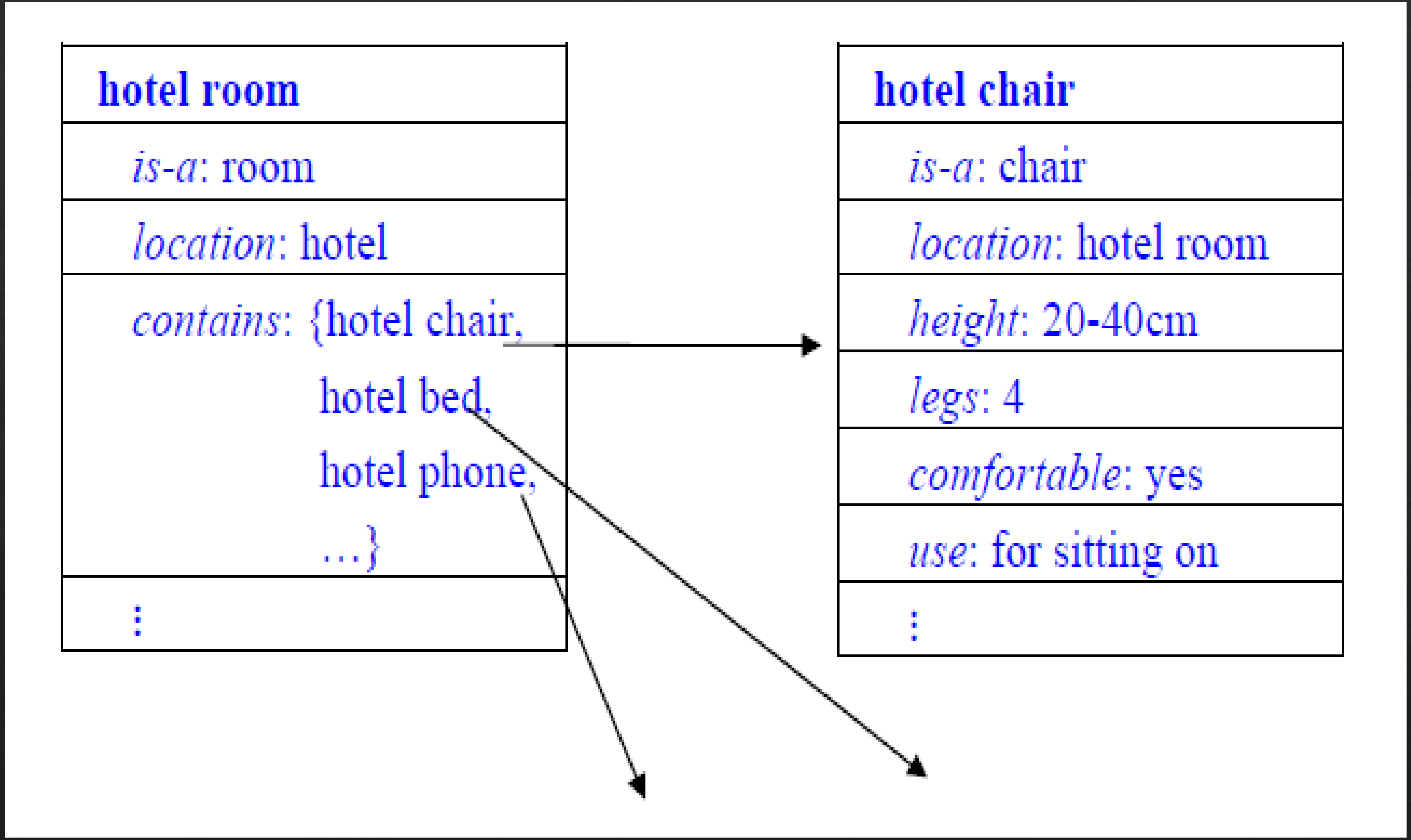
- By Quillian (same). Able to write a program that was asked about similarity between cry and comfort.
- Program outputted:
  - "Cry (2) is among other things to make a sad sound. To comfort (3) can be to make (2) something less sad."

Frames

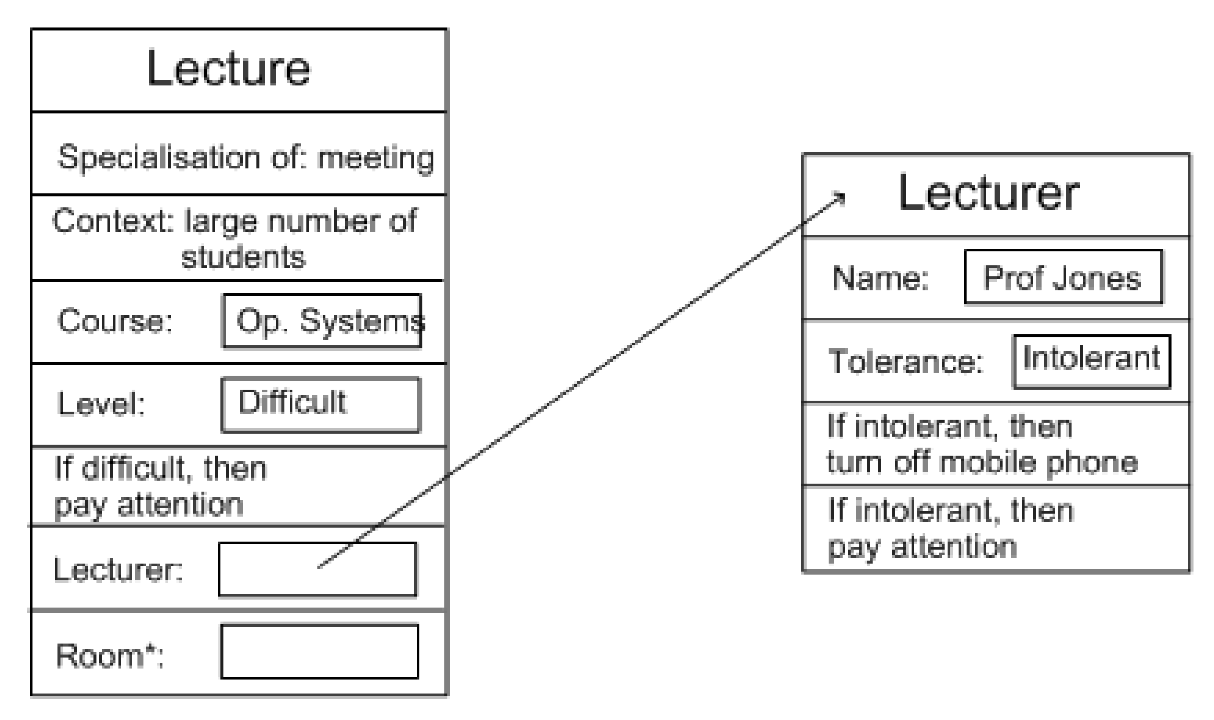
Frame Representations

- Semantic networks where nodes have structure
  - Frame with a number of slots (age, height, ...)
  - Each slot stores specific item of information
- When agent faces a **<u>new situation</u>**
  - Slots can be filled in (value may be another frame)
  - Filling in may **<u>trigger actions</u>**
  - May trigger retrieval of other frames
- ***Inheritance of properties*** between frames
  - **<u>Very similar</u>** to objects in **<u>OOP</u>**

Frame Knowledge Representation



Example: Frame Representation



Flexibility in Frames

- Slots in a frame can contain
  - Information for **choosing a frame** in a situation
  - **Relationships** between this and other frames
  - **Procedures** to carry out after various slots filled
  - Default **information** to use where **input is missing**
  - Blank slots: left blank unless required for a task
  - **Other frames**, which gives a hierarchy

- Can also be expressed in first order logic
  - So essentially equivalent

Representation & Logic

- AI wanted “non-logical representations”
  - Semantic networks
    - Conceptual graphs, frames
- But all can be expressed in first order logic!
- Best of both worlds
  - Logical reading ensures representation well-defined
  - Representations specialised for applications
  - Can make reasoning easier, more intuitive