Al-Mustaqbal University
College of Sciences
Intelligent Medical System Department

جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قسم الانظمة الطبية الذكية

Lecture: (7)

Knowledge Representation

Subject: Artificial Intelligence

Class: Third

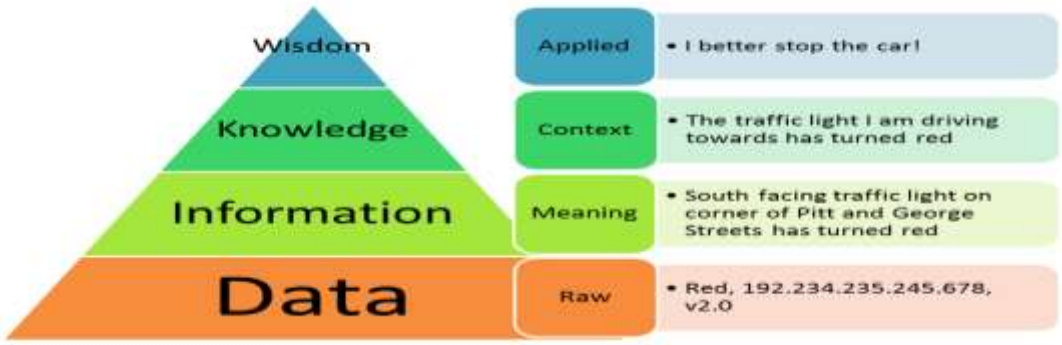
Lecturer: Dr. Maytham N. Meqdad

Knowledge Representation

What is knowledge? Is knowledge the same thing as facts? *some* define knowledge as "the fact or condition of knowing something with familiarity gained through experience or association."

People gain knowledge through experience—they see, hear, touch, feel, and taste the world around them. We can associate something we see with something we hear, thereby gaining new knowledge about the world.

An alternate definition for knowledge is "the fact or condition of being aware of something." How do we make a computer aware of something? Suppose we know that the sun is hot, balls are round, and the sky is blue. These facts are knowledge about the world. How *do* we store this knowledge in our brain? How *could* we store this knowledge in a computer? This problem, called **knowledge representation**, is one of the first, most fundamental issues that researchers in artificial intelligence had to face. And the answer they found was ***symbols.*** Figure below illustrates the hierarchic of the knowledge



A symbol is a number or character string that represents an object or idea. Strings and numbers are used because computers are very good at processing them. This is called the internal representation of the knowledge. However, people are most comfortable using a natural language like English to represent knowledge. Thus, for practical reasons, we need mappings from facts to an internal computer representation and also to a form that people can understand.

The question here? Is the natural language is the best for computer and why?

There are many different kinds of knowledge we may want to represent: simple facts or complex relationships, mathematical formulas or rules for natural language syntax, associations between related concepts, inheritance hierarchies between classes of objects.

Consequently, choosing a knowledge representation for any particular application involves tradeoffs between the needs of people and computers. In addition to being easy to use, a good

knowledge representation also must be easily modified and extended, either by changing the knowledge manually or through automatic machine learning techniques.

We explore here some common kinds of knowledge and the most popular approaches for storing that knowledge in computers.

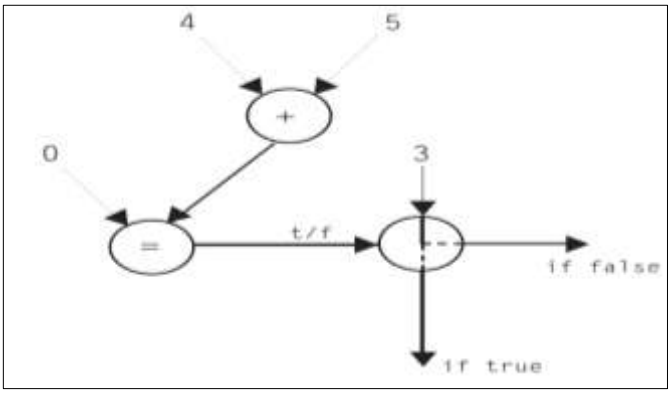
a) Procedural Representation

Perhaps the most common technique for representing knowledge in computers is *procedural* knowledge. Procedural code not only encodes facts (constants or bound variables) but also defines the sequence of operations for using and manipulating those facts. Thus, program code is a perfectly natural way of encoding procedural knowledge.

Programs written in scripting languages such as Visual Basic, JavaScript, and LotusScript are examples of a procedural knowledge representation

Basic idea:

- Knowledge encoded in some procedures
  - small programs that know how to do specific things, how to proceed.
  - *e.g* a parser in a natural language understander has the knowledge that a *noun phrase* may contain articles, adjectives and nouns. It is represented by calls to routines that know how to process articles, adjectives and nouns.



What are the disadvantages of procedural knowledge representation?

Advantages:

- *Heuristic* or domain specific knowledge can be represented.
- *Extended logical inferences*, such as default reasoning facilitated.
- *Side effects* of actions may be modelled. Some rules may become false in time. Keeping track of this in large systems may be tricky.

Disadvantages:

- Completeness -- not all cases may be represented.
- Consistency -- not all deductions may be correct.

  e.g If we know that *Fred is a bird* we might deduce that *Fred can fly*. Later we might discover that *Fred is an emu*.

b) Relational Representation

Another way to represent information is in relational form, such as that used in relational database systems. Relational databases provide a powerful and flexible mechanism for storing knowledge, which is why they have almost completely taken over the business of storing information in commercial business systems.

Knowledge is represented by tuples or records of information about an item, with each tuple containing a set of fields or columns defining specific attributes and values of that item. By storing a collection of information in a table, we can use relational calculus to manipulate the data, based on the relations defined, and query the information stored in the table. Structured Query Language (SQL) is the most popular language for manipulating relational data.

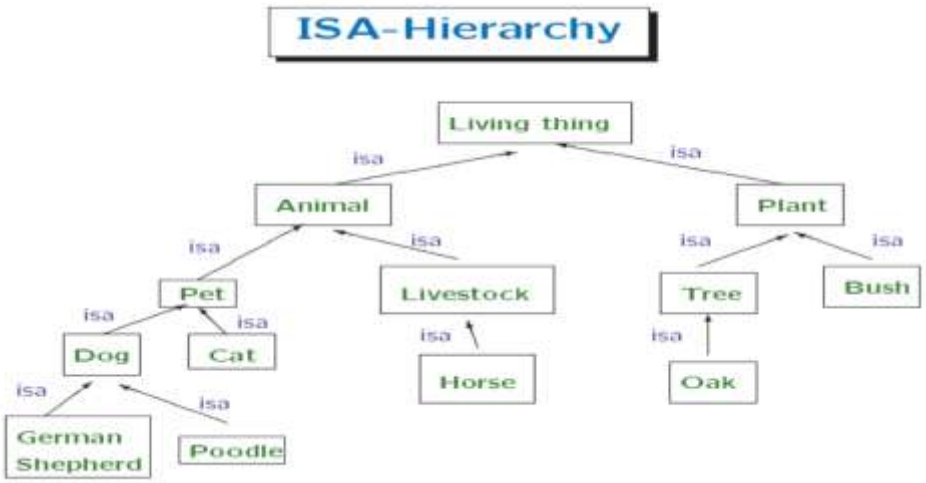
Country	Capital	Currency	Continent
United States	Washington DC	Dollar	North America
Brazil	Brasilia	Real	South America
Australia	Canberra	Dollar	Australasia
India	New Delhi	Rupee	Asia
Germany	Berlin	Euro	Europe
Egypt	Cairo	Pound	Africa

What are the disadvantages of Relational knowledge representation?

c) Hierarchical Representation

Another type of knowledge is inheritable knowledge, which centers on relationships and shared attributes between kinds or classes of objects. Hierarchical knowledge is best used to represent "isa" relationships, where a general or abstract type (for example, ball) is linked to more specific types (rubber, golf, baseball, football) which inherit the basic properties of the general type.

Object-oriented programming languages such as Smalltalk, C++, and Java provide a natural framework for representing knowledge as objects, and for reasoning about and manipulating those objects.



What are the disadvantages Hierarchical knowledge representations?

Formal logic is a language with its own *syntax*, which defines how to make sentences, and corresponding *semantics*, which describe the meaning of the sentences. The most basic form of logic representation is called boolean or

- propositional logic, where each proposition or fact is represented by a symbol that evaluates to either **true** or **false**.

  Sentences can be constructed using proposition symbols (P, Q, R, ...) and boolean connectives, such as (AND), (OR), (not ~) , (P → Q), and (A ≡ B).

  Using this simple syntax, we can write implications or rules, such as

  (P∧Q) → R.

  (P∧Q) is called the premise, R is consequent.

- predicate logic

  It allows *predicates* on objects to define attributes and relations between objects, has become the preferred logic for knowledge representation in artificial intelligence systems. Using objects, attributes, and relations, we can represent almost any type of knowledge.

  Example :

  "*London is cold in the winter*." represented in several ways:

  1- place(London) ∧ temperature(cold) ∧ season(winter)
  2- cold(London, winter).
  3- winter(London, cold).

  Using ***Resolution*** and ***Unification*** of predicate logic provide a way to derive new information.

d) Sematic Nets

The major idea is that:

- The meaning of a concept comes from its relationship to other concepts, and that,
- The information is stored by interconnecting nodes with labelled arcs.

- The physical attributes of a person can be represented as in Fig. which Nodes represent Objects, Links or Arcs represent Relationships, "instance of" - set membership, "is a" – inheritance, " has a" - attribute descriptors, "part of" - aggregation

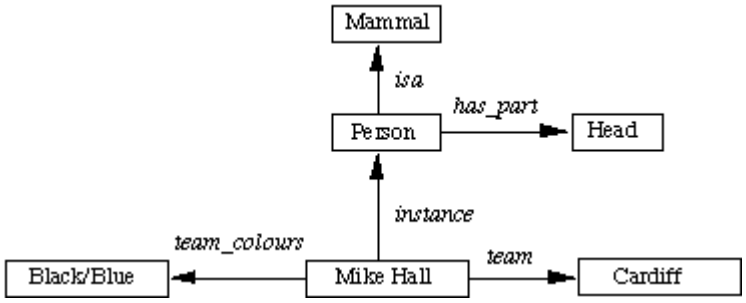


Fig. A Semantic Network

These values can also be represented in logic as: *isa(person, mammal), instance(Mike-Hall, person) team(Mike-Hall, Cardiff).*As a more complex example consider the sentence: *John gave Mary the book*. Here we have several aspects of an event.

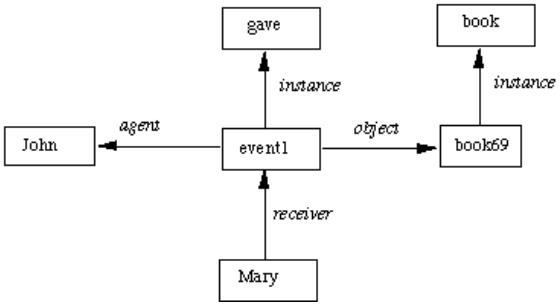


Fig. A Semantic Network for a Sentence

Advantages of Sematic Nets

- Flexible
- easy to understand
- support inheritance
- "natural" way to represent knowledge

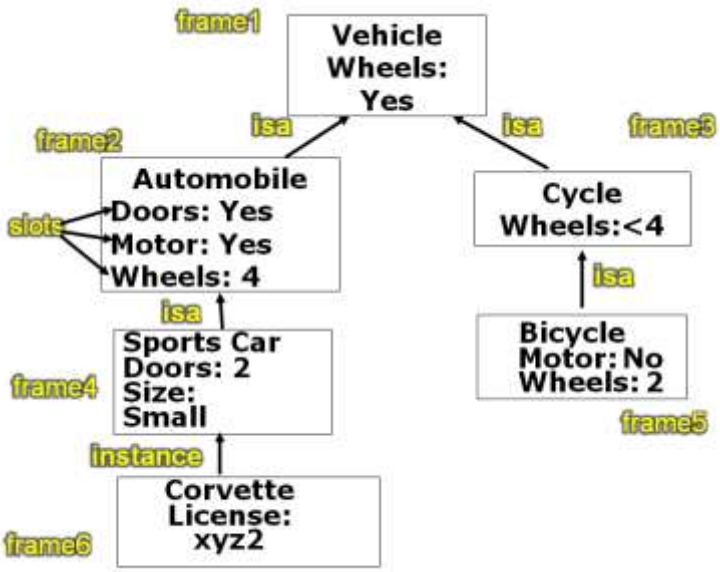
Disadvantages of Sematic Net

- Hard to deal with exceptions
- procedural knowledge difficult to represent

e) Frames

A frame is a collection of attributes which defines the state of an object and its relationship to other frames (objects). But a frame is much more than just a record or data structure containing data. In AI, frames are called slot-and-filler data representations. The slots are the data values, and the fillers are attached procedures which are called *before*, *during* (to compute the value of), or *after* the slot's value is changed. Frames are often linked into a hierarchy to represent has-part and **is-a** relationships.

Example



A frame system interpreter must be capable of the following in order to exploit the frame slot representation:

- Consistency checking -- when a slot value is added to the frame relying on the domain attribute and that the value is legal using range and range constraints.
- Propagation of *definition* values along *isa* and *instance* links.
- Inheritance of default. values along *isa* and *instance* links.
- Computation of value of slot as needed.
- Checking that only correct number of values computed.

Exercises

1. Construct Semantic Net representations of the following:
    1. Dave is Welsh, Dave is a Lecturer.
    2. Paul leant his new Frank Zappa CD to his best friend.
2. Represent the following in a Semantic Net
    1. *Mike and Mary's telephone number is the same*.
    2. *John believes that Mike and Mary's telephone number is the same*.
3. Represent the following in partitioned semantic networks:
    1. Every player kicked a ball.
    2. All players like the referee.
    3. Andrew believes that there is a fish with lungs.
4. What programming languages would be suited to implement a semantic network and frames?