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.

![Diagram](image)

What are the disadvantages of procedural knowledge representation?

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.
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?

d) **Predicate Logic**

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

  \[(P \land Q) \rightarrow R.\]

  \((P \land Q)\) 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- \( \text{place(London)} \land \text{temperature(cold)} \land \text{season(winter)} \)
  2- \( \text{cold(London, winter)}. \)
  3- \( \text{winter(London, cold)}. \)

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