**EXPERT SYSTEMS:**

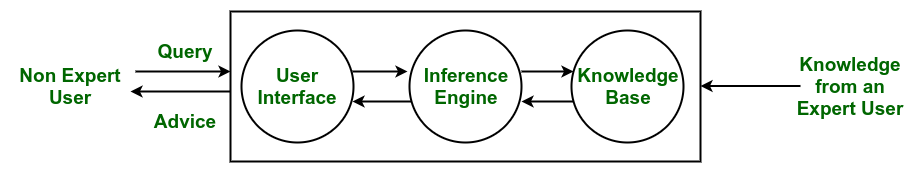
**Artificial Intelligence** is a piece of software that simulates the behaviour and judgement of a human or an organization that has experts in a particular domain is known as an expert system. It does this by acquiring relevant knowledge from its knowledge base and interpreting it according to the user’s problem. The data in the knowledge base is added by humans that are expert in a particular domain and this software is used by a non-expert user to acquire some information. It is widely used in many areas such as medical diagnosis, accounting, coding, games etc.

An expert system is AI software that uses knowledge stored in a knowledge base to solve problems that would usually require a human expert thus preserving a human expert’s knowledge in its knowledge base. They can advise users as well as provide explanations to them about how they reached a particular conclusion or advice. **Knowledge Engineering** is the term used to define the process of building an Expert System and its practitioners are called **Knowledge Engineers**. The primary role of a knowledge engineer is to make sure that the computer possesses all the knowledge required to solve a problem. The knowledge engineer must choose one or more forms in which to represent the required knowledge as a symbolic pattern in the memory of the computer.

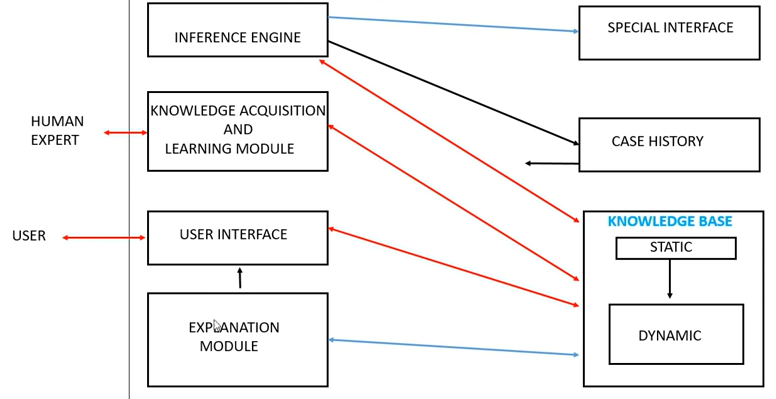
**Example :** There are many examples of an expert system. Some of them are given below –

* **MYCIN –**   
  One of the earliest expert systems based on backward chaining. It can identify various bacteria that can cause severe infections and can also recommend drugs based on the person’s weight.
* **DENDRAL –**  
  It was an artificial intelligence-based expert system used for chemical analysis. It used a substance’s spectrographic data to predict its molecular structure.
* **PXDES –**   
  It could easily determine the type and the degree of lung cancer in a patient based on the data.
* **CaDet –**   
  It is a clinical support system that could identify cancer in its early stages in patients.
* **DXplain –**   
  It was also a clinical support system that could suggest a variety of diseases based on the findings of the doctor.

**Components of an Expert System :**



*Architecture of an Expert System*

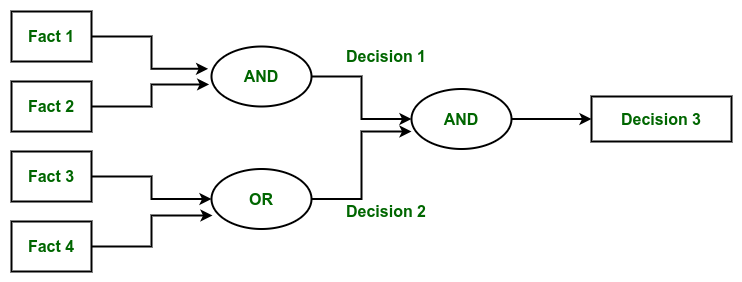
**

* **Knowledge Base –**  
  The knowledge base represents facts and rules. It consists of knowledge in a particular domain as well as rules to solve a problem, procedures and intrinsic data relevant to the domain.
* **Inference Engine –**  
  The function of the inference engine is to fetch the relevant knowledge from the knowledge base, interpret it and to find a solution relevant to the user’s problem. The inference engine acquires the rules from its knowledge base and applies them to the known facts to infer new facts. Inference engines can also include an explanation and debugging abilities.
* **Knowledge Acquisition and Learning Module –**  
  The function of this component is to allow the expert system to acquire more and more knowledge from various sources and store it in the knowledge base.
* **User Interface –**  
  This module makes it possible for a non-expert user to interact with the expert system and find a solution to the problem.
* **Explanation Module –**  
  This module helps the expert system to give the user an explanation about how the expert system reached a particular conclusion.

The Inference Engine generally uses two strategies for acquiring knowledge from the Knowledge Base, namely –

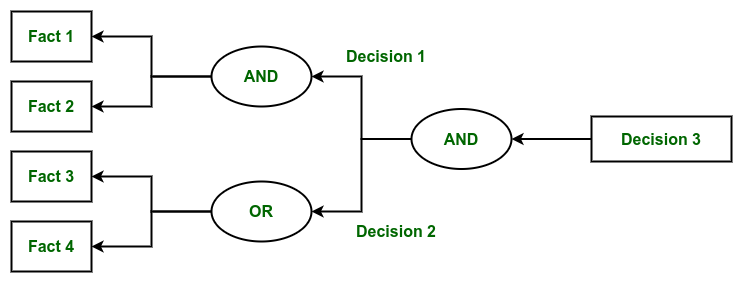
* Forward Chaining
* Backward Chaining

**Forward Chaining –**  
Forward Chaining is a strategic process used by the Expert System to answer the questions – What will happen next. This strategy is mostly used for managing tasks like creating a conclusion, result or effect. Example – prediction or share market movement status.



*Forward Chaining*

**Backward Chaining –**  
Backward Chaining is a storage used by the Expert System to answer the questions – Why this has happened. This strategy is mostly used to find out the root cause or reason behind it, considering what has already happened. Example – diagnosis of stomach pain, blood cancer or dengue, etc.  



*Backward Chaining*

**Characteristics of an Expert System :**

* Human experts are perishable, but an expert system is permanent.
* It helps to distribute the expertise of a human.
* One expert system may contain knowledge from more than one human expert thus making the solutions more efficient.
* It decreases the cost of consulting an expert for various domains such as medical diagnosis.
* They use a knowledge base and inference engine.
* Expert systems can solve complex problems by deducing new facts through existing facts of knowledge, represented mostly as if-then rules rather than through conventional procedural code.
* Expert systems were among the first truly successful forms of artificial intelligence (AI) software.

**Limitations :**

* Do not have human-like decision-making power.
* Cannot possess human capabilities.
* Cannot produce correct result from less amount of knowledge.
* Requires excessive training.

**Advantages :**

* Low accessibility cost.
* Fast response.
* Not affected by emotions, unlike humans.
* Low error rate.
* Capable of explaining how they reached a solution.

**Disadvantages :**

* The expert system has no emotions.
* Common sense is the main issue of the expert system.
* It is developed for a specific domain.
* It needs to be updated manually. It does not learn itself.
* Not capable to explain the logic behind the decision.

**TYPES OF EXPERT SYSTEMS:**

There are five basic types of expert systems. These include a

1. **rule-based expert system** is a straightforward one where knowledge is represented as a set of rules.

**2. frame-based expert system** frames are used to capture and represent knowledge.

The concept of a frame is defined by a collection of slots.

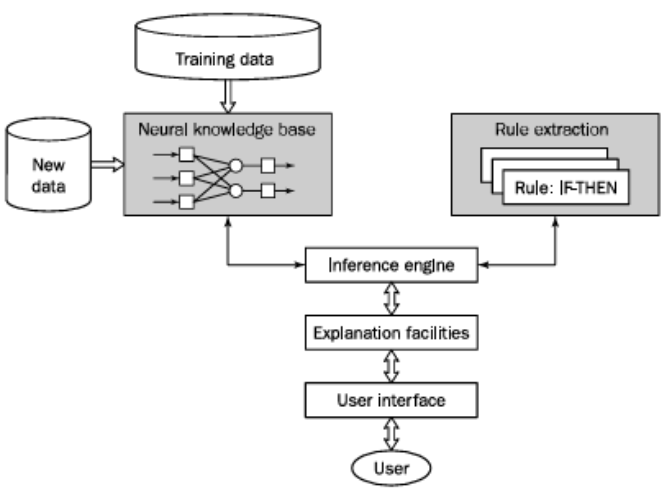
Each slot describes a particular attribute or operation of the frame.

Slots are used to store values. A slot may contain a set of default values or a pointer to another frame,a set of rules or procedure by which the slot value is obtained.

**3. fuzzy expert system** are also called multi-valued logic and differentiate between members of the class from non-members in problem-solving.

**4. neural expert system** replaces a traditional knowledge base with neural knowledge, storing it as weights in neurons. 

Neural expert systems are expert systems that have neural networks for their knowledge bases. The most important features of these systems are the learning algorithm that allows us to generate a knowledge base automatically from training examples, and the ability to handle partial and noisy data

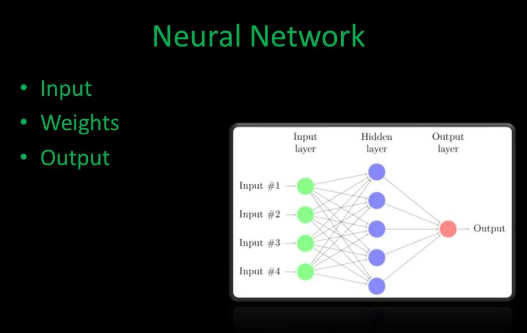


**5. neuro-fuzzy expert system** combines parallel computation and learning abilities with knowledge representation and explanation abilities.

**neuro-fuzzy expert system = NEURAL NETWORK + FUZZY SYSTEMS**

A neuro-fuzzy system is based on a fuzzy system which is trained by a learning algorithm derived from neural network theory. The (heuristical) learning procedure operates on local information, and causes only local modifications in the underlying fuzzy system.

A neuro-fuzzy system can be viewed as a 3-layer feed forward neural network. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables. Fuzzy sets are encoded as (fuzzy) connection weights. It is not necessary to represent a fuzzy system like this to apply a learning algorithm to it. However, it can be convenient, because it represents the data flow of input processing and learning within the model.



* 1. **rule-based expert system**

It is a straightforward one where knowledge is represented as a set of rules.

* The production rules are stored in the long-term memory and the problem-specific information or facts in the short-term memory.

EXAMPLE:

**Recognition of an animal:**

**Has claws**

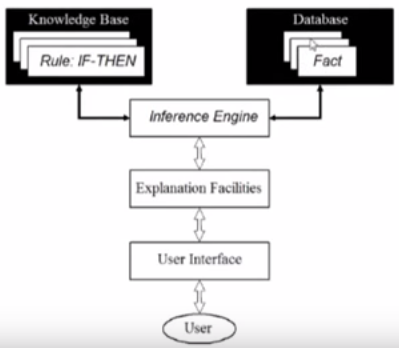
**Has sharp teeth**

**Eats meat carnivores cheetah**

**Has spots**

**Very fast**

Has hair mammals

[](https://4.bp.blogspot.com/-C1PQQlVn740/W6iDsU4XXMI/AAAAAAAAEaI/c-4wC0Fty3E6WceBwlga_BkHHq05Jf8yQCLcBGAs/s1600/m.PNG)

**The Knowledge base**

* The**knowledge base** contains the domain knowledge useful for problem solving.
* In rule-based expert system, the knowledge is represented as a set of rules. Each rule specifies a relation, recommendation, directive, strategy or heuristic and has the IF (condition) THEN (action) structure.
* When the condition part of a rule is satisfied, the rule is said to **fire** and the action part is executed.

**Database**

* The**database** includes a set of facts used to match against the IF (condition) parts of rules stored in the knowledge base.

**Inference engine**

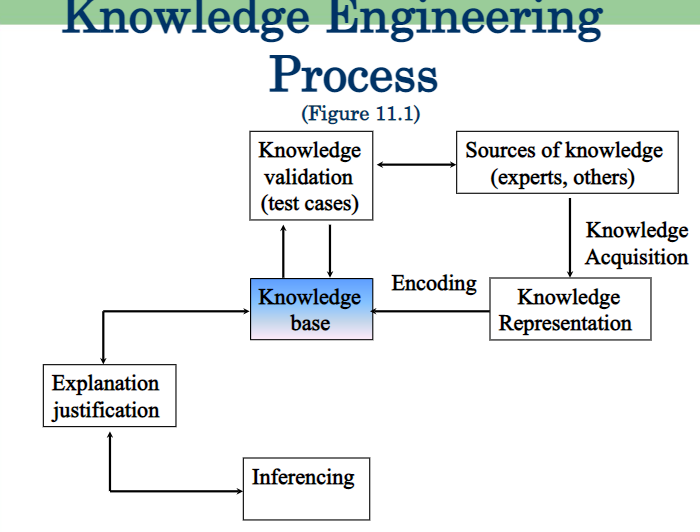
* The **inference engine** carries out the reasoning whereby the expert system reaches a solution.
* It links the rules given in the knowledge base with the facts provided in the database.

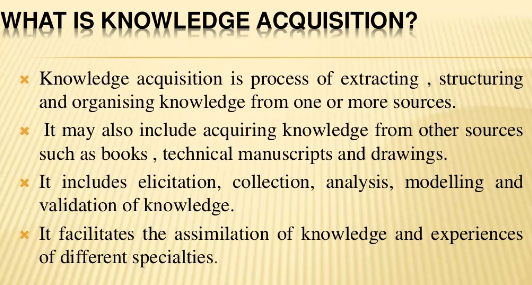
**Explanation facilities**

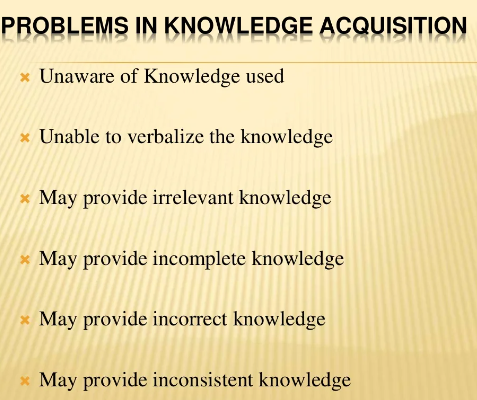
* The **explanation facilities** enable the user to ask the expert system how a particular conclusion is reached and why a specific fact is needed.
* An expert system must be able to explain its reasoning and justify its advice, analysis or conclusion.

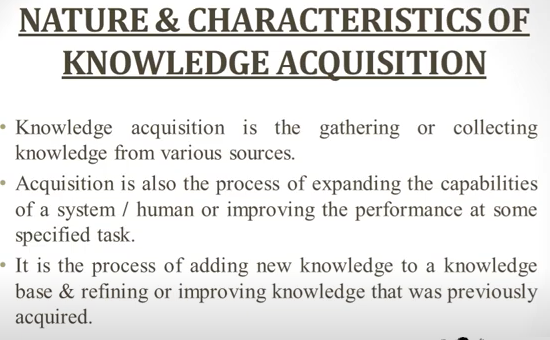
**User interface**

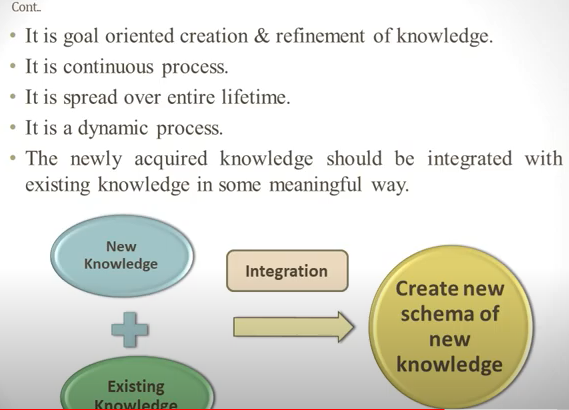
* The **user interface** is the means communication between a user seeking a solution to the problem and an expert system.

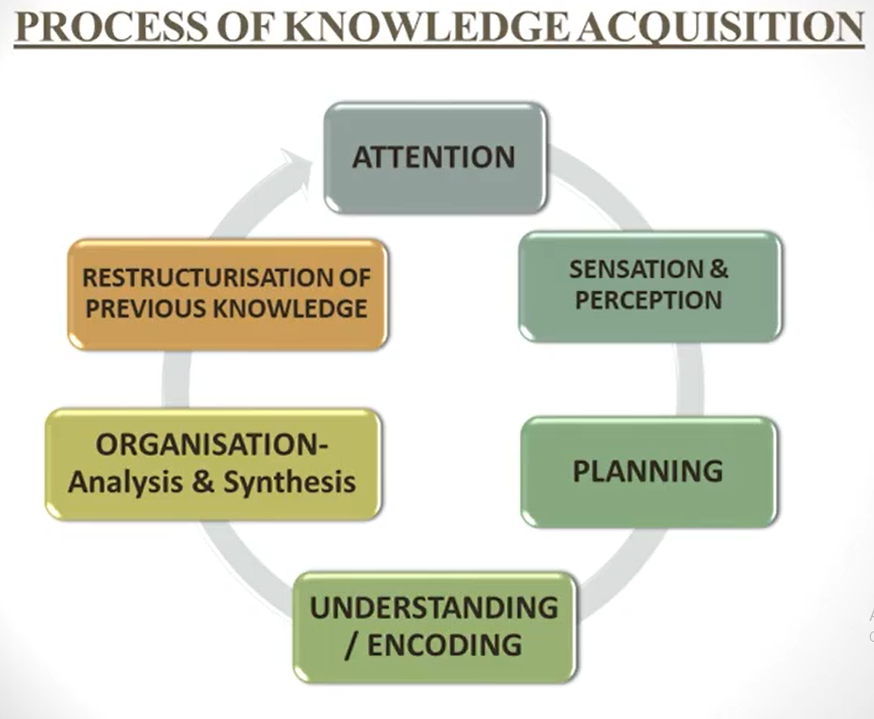












Tasks in KA(CIAD)

1. collect: Acquiring knowledge from expert

2. interpret: Revive & identifying key parts

3. Analyze: Forming theories and strategies to solve the problem

4. Design: Forming better understanding

K.A Techniques:

1. Introspection: Expert act as both expert and knowledge engineer

2. Observation: Expert closely observed the work ex: video recording, analysis

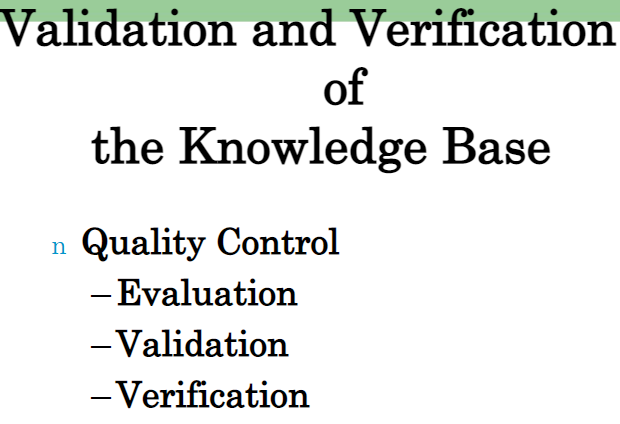
3. Induction: Converting set of examples into rules

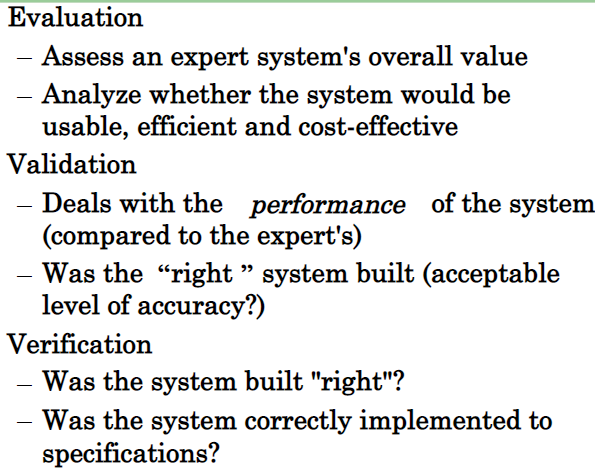
4. Protocol Analysis: Expert is asked to perform task and verbalized through process

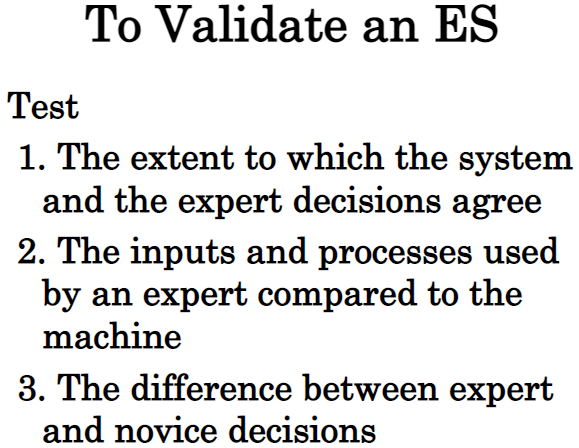
5. Prototyping: Extension of interviewing

Expert +K.E=> System (i.e expert test the system & K.E modify the data)

6. interviewing: It is used in early stages of Acquisition. Experts verify the knowledge







**Expert System Shells**

An Expert system shell is a software development environment. It contains the basic components of expert systems. A shell is associated with a prescribed method for building applications by configuring and instantiating these components.

**Shell components and description**

The generic components of a shell: the knowledge acquisition,

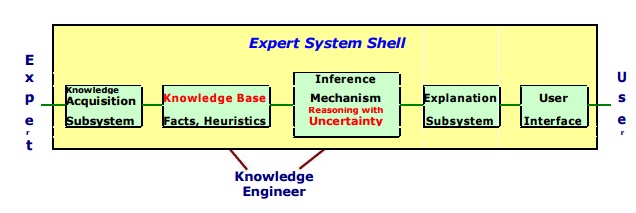
the knowledge Base,

the reasoning,

the explanation and

the user interface

are shown below. The knowledge base and reasoning engine are the core components.



**■ Knowledge Base**

A store of factual and heuristic knowledge. Expert system tool provides one or more knowledge representation schemes for expressing knowledge about the application domain. Some tools use both Frames (objects) and IF-THEN rules. In PROLOG the knowledge is represented as logical statements.

**■ Reasoning Engine**

Inference mechanisms for manipulating the symbolic information and knowledge in the knowledge base form a line of reasoning in solving a problem. The inference mechanism can range from simple modus ponens backward chaining of IF-THEN rules to Case-Based reasoning.

**Knowledge Acquisition subsystem**

A subsystem to help experts in build knowledge bases. However, collecting knowledge, needed to solve problems and build the knowledge base, is the biggest bottleneck in building expert systems.

**Explanation subsystem**

A subsystem that explains the system's actions. The explanation can range from how the final or intermediate solutions were arrived at justifying the need for additional data.

**User Interface**

A means of communication with the user. The user interface is generally not a part of the expert system technology. It was not given much attention in the past. However, the user interface can make a critical difference in the utility of an Expert system.

**AI PROGRAMMING LANGUAGES:**

* **AIML**(meaning "Artificial Intelligence Markup Language")is an XML dialect for use with **A.L.I.C.E**.-type **chatterbots.**
* **C#** can be used to develop high level machine learning models using **Microsoft’**s   **.NET** suite. ML.NET was developed with integration to existing .NET projects in mind, simplifying the process for existing software using the .NET platform.
* **Lisp** was the first language developed for artificial intelligence. It includes features intended to support programs that could perform general problem solving, such as lists, associations, schemas (frames), dynamic memory allocation, data types, recursion, associative retrieval, functions as arguments, generators (streams), and cooperative multitasking.
* **Smalltalk** has been used extensively for simulations, neural networks, machine learning and genetic algorithms. It implements the purest and most elegant form of object-oriented programming using message passing.
* **Prolog** is a declarative language where programs are expressed in terms of relations, and execution occurs by running *queries* over these relations. Prolog is particularly useful for symbolic reasoning, database and language parsing applications. Prolog is widely used in AI today.
* **STRIPS** is a language for expressing **automated planning problem** instances. It expresses an initial state, the goal states, and a set of actions. For each action preconditions (what must be established before the action is performed) and postconditions (what is established after the action is performed) are specified.
* **Planner** is a hybrid between procedural and logical languages. It gives a procedural interpretation to logical sentences where implications are interpreted with pattern-directed inference.
* **POP-11** is a reflective, incrementally compiled programming language with many of the features of an interpreted language. It is the core language of the **Poplog**  programming environment developed originally by the University of Sussex, and recently in the **School of Computer Science** at the **University of** **Birmingham** which hosts the **Poplog website**, It is often used to introduce symbolic programming techniques to programmers of more conventional languages like **Pascal,** who find POP syntax more familiar than that of **Lisp.** One of POP-11's features is that it supports **first-class functions**.
* **R** is widely used in new-style artificial intelligence, involving statistical computations, numerical analysis, the use of Bayesian inference, neural networks and in general **Machine Learning**. In domains like finance, biology, sociology or medicine it is considered one of the main standard languages. It offers several paradigms of programming like vectorial computation, functional programming and object-oriented programming.
* **Python** is widely used for artificial intelligence, with packages for several applications including General AI, **Machine Learning**, **Natural Language Processing**and **Neural Networks**. The application of AI to develop programs that do human-like jobs and portray human skills is Machine Learning. Both Artificial Intelligence and Machine Learning are closely connected and are being used widely today.
* **Haskell** is also a very good programming language for AI. Lazy evaluation and the list and LogicT **monads** make it easy to express non-deterministic algorithms, which is often the case. Infinite data structures are great for search trees. The language's features enable a compositional way of expressing the algorithms. The only drawback is that working with graphs is a bit harder at first because of purity.
* **Wolfram Language** includes a wide range of integrated machine learning capabilities, from highly automated functions like Predict and Classify to functions based on specific methods and diagnostics. The functions work on many types of data, including numerical, categorical, time series, textual, and image.
* **Julia (programming language)**, e.g. for machine learning, using native or non-native libraries.

**OVERVIEW OF PROLOG:**

**Syntax and Basic Fields:**

In prolog, we declare some facts. These facts constitute the Knowledge Base of the system. We can query against the Knowledge Base. We get output as affirmative if our query is already in the knowledge Base or it is implied by Knowledge Base, otherwise we get output as negative. So, Knowledge Base can be considered similar to database, against which we can query. Prolog facts are expressed in definite pattern. Facts contain entities and their relation. Entities are written within the parenthesis separated by comma (,). Their relation is expressed at the start and outside the parenthesis. Every fact/rule ends with a dot (.). So, a typical prolog fact goes as follows:

Format: relation(entity1, entity2, ....k'th entity).

Example :

friends(raju, mahesh).

singer(sonu).

odd\_number(5).

Explanation :

These facts can be interpreted as :

raju and mahesh are friends.

sonu is a singer.

5 is an odd number.

**Key Features :**  
**1. Unification :** The basic idea is, can the given terms be made to represent the same structure.  
**2. Backtracking :** When a task fails, prolog traces backwards and tries to satisfy previous task.  
**3. Recursion :** Recursion is the basis for any search in program.

**Running queries :**

|  |
| --- |
| A typical prolog query can be asked as :  Query 1 : ?- singer(sonu).  Output : Yes.  Explanation : As our knowledge base contains  the above fact, so output was 'Yes', otherwise  it would have been 'No'.  Query 2 : ?- odd\_number(7).  Output : No.  Explanation : As our knowledge base does not  contain the above fact, so output was 'No'. |

**Advantages :**  
**1.**Easy to build database. Doesn’t need a lot of programming effort.  
**2.**Pattern matching is easy. Search is recursion based.  
**3.**It has built in list handling. Makes it easier to play with any algorithm involving lists.

**Disadvantages :**  
**1.** LISP (another logic programming language) dominates over prolog with respect to I/O features.  
**2.** Sometimes input and output is not easy.

**Applications :**

Prolog is highly used in artificial intelligence(AI). Prolog is also used for pattern matching over natural language parse trees.

**LISP**:

John McCarthy invented LISP in 1958, shortly after the development of FORTRAN. It was first implemented by Steve Russell on an IBM 704 computer.

It is particularly suitable for Artificial Intelligence programs, as it processes symbolic information effectively.

Common Lisp originated, during the 1980s and 1990s, in an attempt to unify the work of several implementation groups that were successors to Maclisp, like ZetaLisp and NIL (New Implementation of Lisp) etc.

It serves as a common language, which can be easily extended for specific implementation.

Programs written in Common LISP do not depend on machine-specific characteristics, such as word length etc.

LISP expressions are called symbolic expressions or s-expressions. The s-expressions are composed of three valid objects, atoms, lists and strings.

Any s-expression is a valid program.

LISP programs run either on an **interpreter** or as **compiled code.**

The interpreter checks the source code in a repeated loop, which is also called the read-evaluate-print loop (REPL). It reads the program code, evaluates it, and prints the values returned by the program.

A Simple Program

Let us write an s-expression to find the sum of three numbers 7, 9 and 11. To do this, we can type at the interpreter prompt.

(write (+ 7 9 11))

OUTPUT:27

LISP Uses Prefix Notation

You might have noted that LISP uses **prefix notation.**

In the above program the + symbol works as the function name for the process of summation of the numbers.

In prefix notation, operators are written before their operands. For example, the expression,

a \* ( b + c ) / d

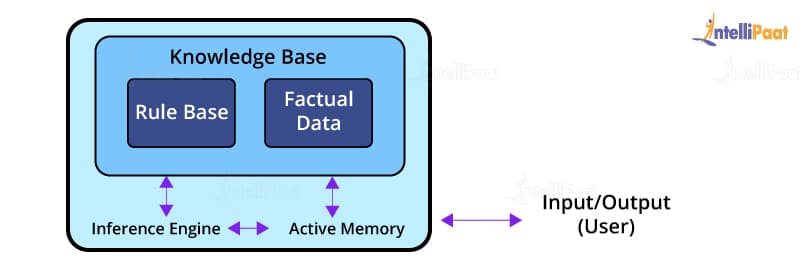
will be written as −

(/ (\* a (+ b c) ) d)

**What is a Production System in AI?**

A production system in AI helps create AI-based computer programs. With the help of it, the automation of various types of machines has become an easy task. The types of machines can be a computer, mobile applications, manufacturing tools, or more. The set of rules in a production system in **Artificial Intelligence defines** the behavior of the machine. It helps the machine respond to the surroundings.

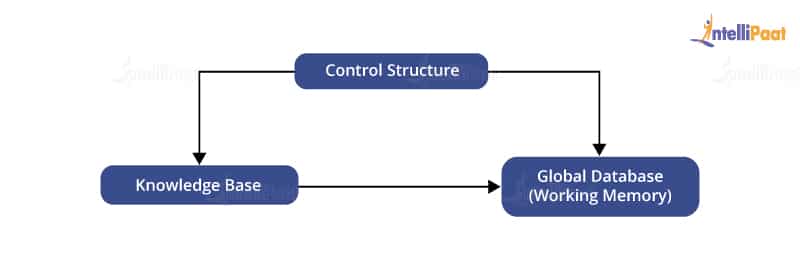
A production system in AI is a type of cognitive architecture that defines specific actions as per certain rules. The rules represent the declarative knowledge of a machine to respond according to different conditions. Today, many **expert systems** and automation methodologies rely on the rules of production systems. Below is the basic architecture of production systems in AI:



The rules in a production system are determined by LHS (left-hand side) and RHS (right-hand side) equations, where LHS denotes the specific condition to be applied, and RHS shows the output of the applied condition.

## ****Components of a Production System in AI****

For making an AI-based intelligent system that performs specific tasks, we need an architecture. The architecture of a production system in Artificial Intelligence consists of production rules, a database, and the control system.



### ****Global Database****

A global database consists of the architecture used as a central data structure. A database contains all the necessary data and information required for the successful completion of a task. It can be divided into two parts as permanent and temporary. The permanent part of the database consists of fixed actions, where as the temporary part alters according to circumstances.

### ****Production Rules****

Production rules in AI are the set of rules that operates on the data fetched from the global database. Also, these production rules are bound with precondition and postcondition that gets checked by the database. If a condition is passed through a production rule and gets satisfied by the global database, then the rule is successfully applied. The rules are of the form A®B, where the right-hand side represents an outcome corresponding to the problem state represented by the left-hand side.

### ****Control System****

The control system checks the applicability of a rule. It helps decide which rule should be applied and terminates the process when the system gives the correct output. It also resolves the conflict of multiple conditions arriving at the same time. The strategy of the control system specifies the sequence of rules that compares the condition from the global database to reach the correct result.

**Features of Production System in Artificial Intelligence**

The main features of the production system include:

**1. Simplicity:** The structure of each sentence in a production system is unique and uniform as they use the “IF-THEN” structure. This structure provides simplicity in knowledge representation. This feature of the production system improves the readability of production rules.

**2. Modularity:** This means the production rule code the knowledge available in discrete pieces. Information can be treated as a collection of independent facts which may be added or deleted from the system with essentially no deleterious side effects.

**3. Modifiability:** This means the facility for modifying rules. It allows the development of production rules in a skeletal form first and then it is accurate to suit a specific application.

**Control/Search Strategies**

How would you decide which rule to apply while searching for a solution for any problem? There are certain requirements for a good control strategy that you need to keep in mind, such as:

* The first requirement for a good control strategy is that it should **cause motion**.
* The second requirement for a good control strategy is that it should be **systematic**.
* Finally, it must be **efficient** in order to find a good answer.

**Classes of Production System in Artificial Intelligence**

There are four major classes of Production System in Artificial Intelligence:

* **Monotonic Production System**: It’s a production system in which the application of a rule never prevents the later application of another rule, that could have also been applied at the time the first rule was selected.
* **Partially Commutative Production System**: It’s a type of production system in which the application of a sequence of rules transforms state X into state Y, then any permutation of those rules that is allowable also transforms state x into state Y. Theorem proving falls under the monotonic partially communicative system.
* **Non-Monotonic Production Systems**: These are useful for solving ignorable problems. These systems are important from an implementation standpoint because they can be implemented without the ability to backtrack to previous states when it is discovered that an incorrect path was followed. This production system increases efficiency since it is not necessary to keep track of the changes made in the search process.
* **Commutative Systems**: These are usually useful for problems in which changes occur but can be reversed and in which the order of operation is not critical. Production systems that are not usually not partially commutative are useful for many problems in which irreversible changes occur, such as chemical analysis. When dealing with such systems, the order in which operations are performed is very important and hence correct decisions must be made at the first attempt itself.

**Knowledge System Building tools in AI:**

Building an expert system involves the correct choice of a knowledge representation strategy and an inferencing process.

One of the earliest tools used in expert system building was LISP.

LISP is a versatile language and can support a rich variety of representational structures and most of the early expert systems, such as MYCIN and PROSPECTOR, were built using some dialect of LISP.

With LISP the designer has total freedom of control over building decisions and the types of representational structures required.

 The three main toolkits in the marketplace are

**KEE (Knowledge Engineering Environment):** Knowledge Engineering Environment (KEE) is a frame-based development tool for expert systems.

**ART (Automated Reasoning Tool):** Automated reasoning is the area of computer science that is concerned with applying reasoning in the form of logic to computing systems

**Knowledge Craft**.

KEE was the first in the field and is therefore the most used of the three. All three are based on LISP so that the designer is free at any time to program in LISP.

However, the tool situation has certainly not stabilized. In the future, there will be a move to more sophisticated shells with a wide range of facilities at reasonably affordable price.