UNIT 4

PROCEDURAL vs DECLARATIVE

**Difference between Procedural and Declarative Knowledge**

**Procedural Knowledge:**

* **How to operate**
* **How to make**
* **How to use procedures**

Procedural Knowledge also known as Interpretive knowledge, is the type of knowledge in which it clarifies how a particular thing can be accomplished. It is not so popular because it is generally not used.

Ex: How to make coffee

Emp ID :empts2056 - Declarative

Emp age : 50 - Declarative

Emp tax : 1000,2000 (it varies)

**Declarative Knowledge:**  
Declarative Knowledge also known as Descriptive knowledge, is the type of knowledge which tells the basic knowledge about something and it is more popular than Procedural Knowledge.  
It emphasize **what to do** something to solve a given problem.

-Fact based knowledge.

-Static in nature

Ex: Sun rises in the East.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Key | Procedural Knowledge | Declarative Knowledge |
| 1 | Name | Procedural knowledge is also termed as imperative knowledge. | Declarative knowledge is also termed as functional knowledge |
| 2 | Basis | Procedural knowledge revolves around **How to** the concept. | Declarative knowledge revolves around **What to** the concept. |
| 3 | Communication | Procedural knowledge is difficult to communicate. | Declarative knowledge is easily communicable. |
| 4 | Orientation | Procedural knowledge is process-oriented. | Declarative knowledge is data-oriented. |
| 5 | Validation | Validation is not very easy in procedural knowledge. | Validation is quite easy in declarative knowledge. |
| 6 | Debugging | Debugging is not very easy in procedural knowledge. | Debugging is quite easy in declarative knowledge. |

|  |  |  |
| --- | --- | --- |
| S.NO | Procedural Knowledge | Declarative Knowledge |
| 1. | It is also known as Interpretive knowledge. | It is also known as Descriptive knowledge. |
| 2. | Procedural Knowledge means how a particular thing can be accomplished. | While Declarative Knowledge means basic knowledge about something. |
| 3. | Procedural Knowledge is generally not used means it is not more popular. | Declarative Knowledge is more popular. |
| 4. | Procedural Knowledge can’t be easily communicate. | Declarative Knowledge can be easily communicate. |
| 5. | Procedural Knowledge is generally process oriented in nature. | Declarative Knowledge is data oriented in nature. |
| 6. | In Procedural Knowledge debugging and validation is not easy. | In Declarative Knowledge debugging and validation is easy. |
| 7. | Procedural Knowledge is less effective in competitive programming. | Declarative Knowledge is more effective in competitive programming. |

**LOGIC PROGRAMMING**

Logic programming is a programming paradigm that is based on logic. This means that a logic programming language has sentences that follow logic, so that they express facts and rules. Computation using logic programming is done by making logical inferences based on all available data. In order for computer programs to make use of logic programming, there must be a base of existing logic, called predicates. Predicates are used to build atomic formulas or atoms, which state true facts. Predicates and atoms are used to create formulas and perform queries. Logic languages most often rely on queries in order to display relevant data. These queries can exist as part of machine learning, which can be run without the need for manual intervention.

There are multiple different logic programming languages. The most common language, Prolog (from the French *programmation en logique,*orprogramming in logic), can also interface with other programming languages such as Java and C. On top of being the most popular logic programming language, Prolog was also one of the first such languages, with the first prolog program created in the 1970s for use with interpretations. Prolog was developed using first-order logic, also called predicate logic, which allows for the use of variables rather than propositions. Prolog utilizes artificial intelligence (AI) to help form its conclusions and can quickly process large amounts of data. Prolog can be run with or without manual inputs, meaning in it can be programmed to run automatically as part of data processing.

Logic programming, and especially Prolog, can help businesses and organizations through:

* **Natural language processing:**Natural language processing (NLP) allows for better interactions between humans and computers. NLP can listen to human language in real time, and then processes and translate it for computers. This allows technology to “understand” natural language. However, NLP is not limited just to spoken language. Instead, NLP can also be utilized to read and understand documentation, both in physical print or from word processing programs. NLP is used by technologies such as Amazon Alexa and Google Home to process and understand spoken instructions, as well as by email applications to filter spam emails and warn of phishing attempts.
* **Database management:** Logic programming can be used for the creation, maintenance, and querying of NoSQL databases. Logic programming can create databases out of big data. The programming can identify which information has been programmed as relevant, and store it in the appropriate area. Users can then query these databases with specific questions, such as “What’s the best route to get to New York,” and logic languages can quickly sift through all of the data, run analyses, and return the relevant result with no additional work required by the user.
* **Predictive analysis:** With large data sets, logic languages can search for inconsistencies or areas of differentiation in order to make predictions. This can be useful in identifying potentially dangerous activities (such as going for a bike ride in the middle of a thunderstorm) or for predicting failures of industrial machines. It can also be used to analyze photos and make predictions around the images, such as predicting the identity of objects in satellite photos, or recognizing the patterns that differentiate craters from regular land masses.

**Logic Programming is the combination of two words, logic and programming.**

**Logic Programming is a programming paradigm in which the problems are expressed as facts and rules by program statements but within a system of formal logic.**

**Just like other programming paradigms like object oriented, functional, declarative, and procedural, etc., it is also a particular way to approach programming.**

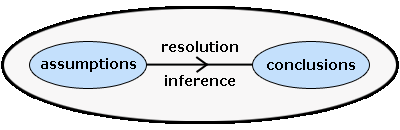
**Logic Programming uses facts and rules for solving the problem. That is why they are called the building blocks of Logic Programming. A goal needs to be specified for every program in logic programming. To understand how a problem can be solved in logic programming, we need to know about the building blocks − Facts and Rules −**

**Facts**

* **Actually, every logic program needs facts to work with so that it can achieve the given goal. Facts basically are true statements about the program and data. For example, Delhi is the capital of India.**

**Rules**

* **Actually, rules are the constraints which allow us to make conclusions about the problem domain. Rules basically written as logical clauses to express various facts. For example, if we are building any game then all the rules must be defined.**

****

**In logic programming, logic is used to represent knowledge and**[**inference**](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html)**is used to manipulate it.**

**The logic used to represent knowledge in logic programming is**[**clausal form**](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html)**which is a subset of**[**first-order predicate logic**](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html)**. It is used because first-order logic is well understood and able to represent all computational problems. Knowledge is manipulated using the resolution inference system which is required for proving theorems in clausal-form logic. The diagram below shows the essence of logic programming.**

**First-Order Logic**

[**First order logic**](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html)**is an extension of**[**propositional logic**](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html)**. It considers whether things are true or false in a partial view of the world, called a**[**domain**](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html)**. Logic programming is based upon an extension of first-order logic. As first order logic is well understood and can be mathematically provable it can be used for representing all computational problems**

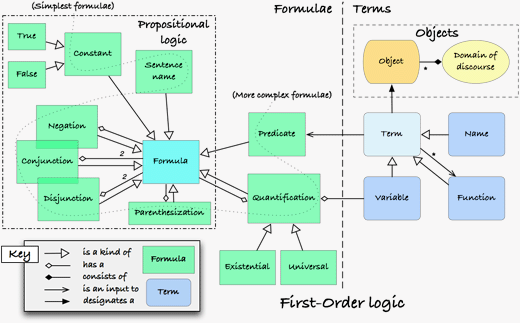
* **Rules are very important to solve any problem in Logic Programming. Rules are basically logical conclusion which can express the facts. Following is the syntax of rule −**
* **A∶− B1,B2,...,Bn.**
* **Here, A is the head and B1, B2, ... Bn is the body.**
* **For example − ancestor(X,Y) :- father(X,Y).**
* **ancestor(X,Z) :- father(X,Y), ancestor(Y,Z).**
* **This can be read as, for every X and Y, if X is the father of Y and Y is an ancestor of Z, X is the ancestor of Z. For every X and Y, X is the ancestor of Z, if X is the father of Y and Y is an ancestor of Z.**

**EXAMPLES :**

* **John likes all kind of food.**
* **Apple and vegetable are food**
* **Anything anyone eats and not killed is food.**
* **Anil eats peanuts and still alive**
* **Harry eats everything that Anil eats.  
  Prove by resolution that:**
* **John likes peanuts.**

First-Order Logic

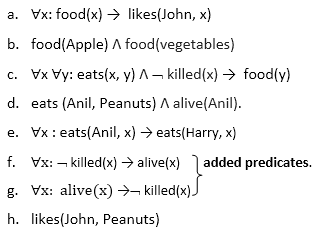
[First order logic](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html#firstorder) is an extension of [propositional logic](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html#propositional). It considers whether things are true or false in a partial view of the world, called a [domain](http://www.doc.ic.ac.uk/~cclw05/topics1/glossary.html#domain). Logic programming is based upon an extension of first-order logic. As first order logic is well understood and can be mathematically provable it can be used for representing all computational problems.



First order logic is made up of syntax and semantics. The syntax of first order logic is a formal language that is used to express concepts. The semantics of first order logic formulae tells us how to determine the truth value of any first order logic formula.

First order logic consists of an alphabet, a first order language, a set of axioms and a set of inference rules. The alphabet consists of seven classes of symbols:

* **Variables** - A sequence of alphanumeric characters that refer to objects in the domain.
  + e.g. u, v, w, x, y, z, foo.
* **Constants** - Represent objects in the domain.
  + e.g. a, b, c, Mary, Green.
* **Function symbols** - A relation among objects, of various arities greater than 0, that maps one object onto another.
  + e.g. f, g, h, fatherof(Mary), isBlue(sky).
* **Predicate symbols** - Maps one or more objects onto truth values.
  + e.g. p, q, r, greater(5,3), colour(Grass, Green).
* **Connectives** - Used to link together other members of the alphabet.
  + e.g. NOT, AND, OR, IMPLIES, IF AND ONLY IF.
* **Quantifiers** - Make an assertion about a formula applied over all objects in the domain.
  + e.g. For All (universal), There Exists (existential).
* **Punctuation symbols** - Used to make sentences more readable.
  + e.g. ')', '('

****

**Logical Programming**is a type of programming paradigm that uses logic circuits to control how facts and rules about the problems within the system are represented or expressed. In it, logic is used to represent knowledge, and inference is used to manipulate it.

It is totally based on formal logic.

In this programming paradigm, program statements usually express or represent facts and rules related to problems within a system of formal logic.

These are specially designed for fault diagnosis, natural language processing, planning, and machine learning.

Its main aim is to allow machines to reason because it is very useful for representing knowledge.

It is data-driven, array-oriented, used to express knowledge, etc.

It usually supports the logic programming paradigm.

**Horn Clause and Definite clause:**

* Horn clause and definite clause are the forms of sentences, which enables knowledge base to use a more restricted and efficient inference algorithm.
* **Definite clause:** A clause which is a disjunction of literals with **exactly one positive literal** is known as a definite clause or strict horn clause.
* **Horn clause:** A clause which is a disjunction of literals with **at most one positive literal** is known as horn clause. Hence all the definite clauses are horn clauses.
* **Example: (¬ p V ¬ q V k)**. It has only one positive literal k.
* It is equivalent to p ∧ q → k.

**FORWARD & BACKWARD REASONING**

****

|  |  |  |  |
| --- | --- | --- | --- |
| **BASIS FOR COMPARISON** | **FORWARD REASONING** | **BACKWARD REASONING** | |
| **Basic** | **Data-driven** | | **Goal driven** |
| **Begins with** | **New Data** | | **Uncertain conclusion** |
| **Objective is to find the** | **Conclusion that must follow** | | **Facts to support the conclusions** |
| **Type of approach** | **Opportunistic** | | **Conservative** |
| **Flow** | **Incipient to consequence** | | **Consequence to incipient** |

**Definition of Forward Reasoning**

* **The solution of a problem generally includes the initial data and facts in order to arrive at the solution. These unknown facts and information is used to deduce the result.**
* **For example, while diagnosing a patient the doctor first check the symptoms and medical condition of the body such as temperature, blood pressure, pulse, eye colour, blood, etcetera.**
* **After that, the patient symptoms are analysed and compared against the predetermined symptoms. Then the doctor is able to provide the medicines according to the symptoms of the patient. So, when a solution employs this manner of reasoning, it is known as forward reasoning.**

**Steps that are followed in the forward reasoning**

* **The inference engine explores the knowledge base with the provided information for constraints whose precedence matches the given current state.**
* **In the first step, the system is given one or more than one constraints.**
* **Then the rules are searched in the knowledge base for each constraint. The rules that fulfil the condition are selected(i.e., IF part).**
* **Now each rule is able to produce new conditions from the conclusion of the invoked one. As a result, THEN part is again included in the existing one.**
* **The added conditions are processed again by repeating step 2. The process will end if there is no new conditions exist.**

**Definition of Backward Reasoning**

* **The backward reasoning is inverse of forward reasoning in which goal is analysed in order to deduce the rules, initial facts and data.**
* **We can understand the concept by the similar example given in the above definition, where the doctor is trying to diagnose the patient with the help of the inceptive data such as symptoms. However, in this case, the patient is experiencing a problem in his body, on the basis of which the doctor is going to prove the symptoms. This kind of reasoning comes under backward reasoning.**

**Steps that are followed in the backward reasoning**

* **In this type of reasoning, the system chooses a goal state and reasons in the backward direction. Now, let’s understand how does it happens and what steps are followed.**
* **Firstly, the goal state and the rules are selected where the goal state reside in the THEN part as the conclusion.**
* **From the IF part of the selected rule the subgoals are made to be satisfied for the goal state to be true.**
* **Set initial conditions important to satisfy all the subgoals.**
* **Verify whether the provided initial state matches with the established states. If it fulfils the condition then the goal is the solution otherwise other goal state is selected.**

**Key Differences Between Forward and Backward Reasoning in AI**

* **The forward reasoning is data-driven approach while backward reasoning is a goal driven.**
* **The process starts with new data and facts in the forward reasoning. Conversely, backward reasoning begins with the results.**
* **Forward reasoning aims to determine the result followed by some sequences. On the other hand, backward reasoning emphasis on the acts that support the conclusion.**
* **The forward reasoning is an opportunistic approach because it could produce different results. As against, in backward reasoning, a specific goal can only have certain predetermined initial data which makes it restricted.**
* **The flow of the forward reasoning is from the antecedent to consequent while backward reasoning works in reverse order in which it starts from conclusion to incipient.**

**Conclusion**

* **The production system structure of the search process facilitates in the interpretation of the forward and backward reasoning.**
* **The forward and backward reasoning are differentiated on the basis of their purpose and process, in which forward reasoning is directed by the initial data and intended to find the goal while the backward reasoning is governed by goal instead of the data and aims to discover the basic data and facts.**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Forward Chaining** | **Backward Chaining** |
| **1.** | **Forward chaining starts from known facts and applies inference rule to extract more data unit it reaches to the goal.** | **Backward chaining starts from the goal and works backward through inference rules to find the required facts that support the goal.** |
| **2.** | **It is a bottom-up approach** | **It is a top-down approach** |
| **3.** | **Forward chaining is known as data-driven inference technique as we reach to the goal using the available data.** | **Backward chaining is known as goal-driven technique as we start from the goal and divide into sub-goal to extract the facts.** |
| **4.** | **Forward chaining reasoning applies a breadth-first search strategy.** | **Backward chaining reasoning applies a depth-first search strategy.** |
| **6.** | **Forward chaining is suitable for the planning, monitoring, control, and interpretation application.** | **Backward chaining is suitable for diagnostic, prescription, and debugging application.** |
| **7.** | **Forward chaining can generate an infinite number of possible conclusions.** | **Backward chaining generates a finite number of possible conclusions.** |
| **8.** | **It operates in the forward direction.** | **It operates in the backward direction.** |
| **9.** | **Forward chaining is aimed for any conclusion.** | **Backward chaining is only aimed for the required data.** |

* **It is a crime for an American to sell weapons to hostile nations. (Let's say p, q, and r are variables)  
  American (p) ∧ weapon(q) ∧ sells (p, q, r) ∧ hostile(r) → Criminal(p)       ...(1)**
* **Country A has some missiles. ?p Owns(A, p) ∧ Missile(p). It can be written in two definite clauses by using Existential Instantiation, introducing new Constant T1.  
  Owns(A, T1)             ......(2)  
  Missile(T1)             .......(3)**
* **All of the missiles were sold to country A by Robert.  
  ?p Missiles(p) ∧ Owns (A, p) → Sells (Robert, p, A)       ......(4)**
* **Missiles are weapons.  
  Missile(p) → Weapons (p)             .......(5)**
* **Enemy of America is known as hostile.  
  Enemy(p, America) →Hostile(p)             ........(6)**
* **Country A is an enemy of America.  
  Enemy (A, America)             .........(7)**
* **Robert is American  
  American(Robert).             ..........(8)**

**EXAMPLE**

* **"As per the law, it is a crime for an American to sell weapons to hostile nations. Country A, an enemy of America, has some missiles, and all the missiles were sold to it by Robert, who is an American citizen."**
* **Prove that "Robert is criminal."**

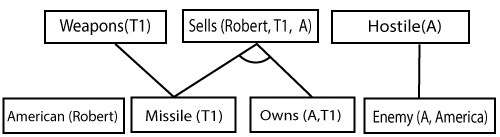
**Step-1:**

* **In the first step we will start with the known facts and will choose the sentences which do not have implications, such as: American(Robert), Enemy(A, America), Owns(A, T1), and Missile(T1).**

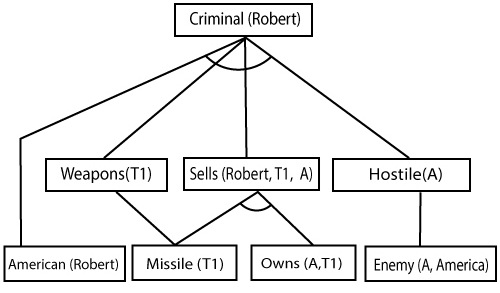
**Forward Chaining and backward chaining in AI**

**Step-2:**

* **At the second step, we will see those facts which infer from available facts and with satisfied premises.**
* **Rule-(1) does not satisfy premises, so it will not be added in the first iteration.**
* **Rule-(2) and (3) are already added.**
* **Rule-(4) satisfy with the substitution {p/T1}, so Sells (Robert, T1, A) is added, which infers from the conjunction of Rule (2) and (3).**
* **Rule-(6) is satisfied with the substitution(p/A), so Hostile(A) is added and which infers from Rule-(7).**

****

**At step-3, as we can check Rule-(1) is satisfied with the substitution {p/Robert, q/T1, r/A}, so we can add Criminal(Robert) which infers all the available facts. And hence we reached our goal statement.**

****

**Matching**

* **Intelligent matching is a type of data management technique in which data is searched, indexed and retrieved from a database through a series of artificial-intelligence-based data sorting and matching algorithms.**
* **Intelligent matching makes it possible to find data based on the principles of semantics, where human-like searching and inference techniques are applied to each query.**
* **Intelligent Matching**
* **Intelligent matching is primarily used in maintaining and extracting databases, specifically those that are very large and complex in nature.**
* **It is generally implemented within database software, business intelligence solutions or a big data analytics application. It works by applying reasoning-based data matching techniques, which eventually deliver ideal or substantially related query results.**

**Some of the services intelligent matching provides include:**

* **The ability to scan each object for duplication within the target database**
* **The ability to remove duplicates within databases**
* **The ability to search and extract relevant information from big data repositories**
* **The ability to compare data, objects or files for similarities**