Artificial Intelligence

intelligent systems

tutorialspoint

SIMPLY EASY LEARNING

About the Tutorial

This tutorial provides introductory knowledge on Artificial Intelligence. It would come to a great help if you are about to select Artificial Intelligence as a course subject. You can briefly know about the areas of AI in which research is prospering.

Audience

This tutorial is prepared for the students at beginner level who aspire to learn Artificial Intelligence.

Prerequisites

The basic knowledge of Computer Science is mandatory. The knowledge of Mathematics, Languages, Science, Mechanical or Electrical engineering is a plus.

Disclaimer & Copyright

Table of Contents

1. Overview of AI

Since the invention of computers or machines, their capability to perform various tasks went on growing exponentially. Humans have developed the power of computer systems in terms of their diverse working domains, their increasing speed, and reducing size with respect to time.

A branch of Computer Science named *Artificial Intelligence* pursues creating the computers or machines as intelligent as human beings.

What is Artificial Intelligence?

According to the father of Artificial Intelligence John McCarthy, it is *"The science and engineering of making intelligent machines, especially intelligent computer programs".*

Artificial Intelligence is a way of **making a computer, a computer-controlled robot, or a software think intelligently**, in the similar manner the intelligent humans think.

AI is accomplished by studying how human brain thinks, and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

Philosophy of AI

While exploiting the power of the computer systems, the curiosity of human, lead him to wonder, "Can a machine think and behave like humans do?"

Thus, the development of AI started with the intention of creating similar intelligence in machines that we find and regard high in humans.

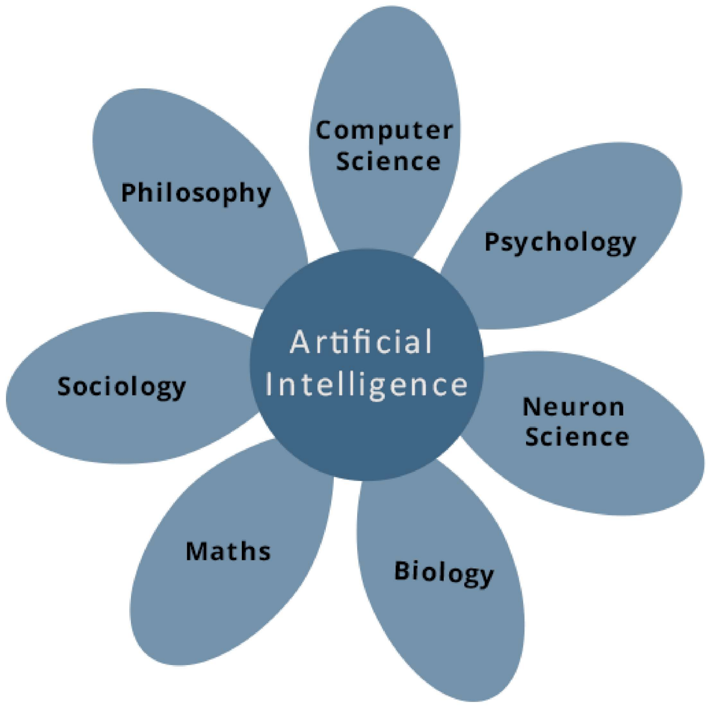
Goals of AI

- **To Create Expert Systems:** The systems which exhibit intelligent behavior, learn, demonstrate, explain, and advice its users.
- **To Implement Human Intelligence in Machines:** Creating systems that understand, think, learn, and behave like humans.

What Contributes to AI?

Artificial intelligence is a science and technology based on disciplines such as Computer Science, Biology, Psychology, Linguistics, Mathematics, and Engineering. A major thrust of AI is in the development of computer functions associated with human intelligence, such as reasoning, learning, and problem solving.

Out of the following areas, one or multiple areas can contribute to build an intelligent system.



Programming Without and With AI

The programming without and with AI is different in following ways:

Programming Without AI	Programming With AI
A computer program without AI can answer the **specific** questions it is meant to solve.	A computer program with AI can answer the **generic** questions it is meant to solve.
Modification in the program leads to change in its structure.	AI programs can absorb new modifications by putting highly independent pieces of information together. Hence you can modify even a minute piece of information of program without affecting its structure.
Modification is not quick and easy. It may lead to affecting the program adversely.	Quick and Easy program modification.

What is AI Technique?

In the real world, the knowledge has some unwelcomed properties:

- Its volume is huge, next to unimaginable.
- It is not well-organized or well-formatted.
- It keeps changing constantly.

AI Technique is a manner to organize and use the knowledge efficiently in such a way that:

- It should be perceivable by the people who provide it.
- It should be easily modifiable to correct errors.
- It should be useful in many situations though it is incomplete or inaccurate.

AI techniques elevate the speed of execution of the complex program it is equipped with.

Applications of AI

AI has been dominant in various fields such as:

- **Gaming**
  AI plays crucial role in strategic games such as chess, poker, tic-tac-toe, etc., where machine can think of large number of possible positions based on heuristic knowledge.
- **Natural Language Processing**
  It is possible to interact with the computer that understands natural language spoken by humans.
- **Expert Systems**
  There are some applications which integrate machine, software, and special information to impart reasoning and advising. They provide explanation and advice to the users.
- **Vision Systems**
  These systems understand, interpret, and comprehend visual input on the computer. For example,
  - A spying aeroplane takes photographs which are used to figure out spatial information or map of the areas.
  - Doctors use clinical expert system to diagnose the patient.
  - Police use computer software that can recognize the face of criminal with the stored portrait made by forensic artist.

- **Speech Recognition**

  Some intelligent systems are capable of hearing and comprehending the language in terms of sentences and their meanings while a human talks to it. It can handle different accents, slang words, noise in the background, change in human's noise due to cold, etc.

- **Handwriting Recognition**

  The handwriting recognition software reads the text written on paper by a pen or on screen by a stylus. It can recognize the shapes of the letters and convert it into editable text.

- **Intelligent Robots**

  Robots are able to perform the tasks given by a human. They have sensors to detect physical data from the real world such as light, heat, temperature, movement, sound, bump, and pressure. They have efficient processors, multiple sensors and huge memory, to exhibit intelligence. In addition, they are capable of learning from their mistakes and they can adapt to the new environment.

History of AI

Here is the history of AI during 20^{th} century:

Year	Milestone / Innovation
1923	Karel Čapek's play named "Rossum's Universal Robots" (RUR) opens in London, first use of the word "robot" in English.
1943	Foundations for neural networks laid.
1945	Isaac Asimov, a Columbia University alumni, coined the term *Robotics.*
1950	Alan Turing introduced Turing Test for evaluation of intelligence and published *Computing Machinery and Intelligence.* Claude Shannon published *Detailed Analysis of Chess Playing* as a search.
1956	John McCarthy coined the term *Artificial Intelligence.* Demonstration of the first running AI program at Carnegie Mellon University.
1958	John McCarthy invents LISP programming language for AI.
1964	Danny Bobrow's dissertation at MIT showed that computers can understand natural language well enough to solve algebra word problems correctly.
1965	Joseph Weizenbaum at MIT built *ELIZA*, an interactive problem that carries on a dialogue in English.
1969	Scientists at Stanford Research Institute Developed *Shakey*, a robot, equipped with locomotion, perception, and problem solving.

1973	The Assembly Robotics group at Edinburgh University built *Freddy*, the Famous Scottish Robot, capable of using vision to locate and assemble models.
1979	The first computer-controlled autonomous vehicle, Stanford Cart, was built.
1985	Harold Cohen created and demonstrated the drawing program, *Aaron*.
1990	Major advances in all areas of AI: • Significant demonstrations in machine learning • Case-based reasoning • Multi-agent planning • Scheduling • Data mining, Web Crawler • natural language understanding and translation • Vision, Virtual Reality • Games
1997	The Deep Blue Chess Program beats the then world chess champion, Garry Kasparov.
2000	Interactive robot pets become commercially available. MIT displays *Kismet*, a robot with a face that expresses emotions. The robot *Nomad* explores remote regions of Antarctica and locates meteorites.

2. IntelligenT Systems

While studying artificially intelligence, you need to know what intelligence is. This chapter covers Idea of intelligence, types, and components of intelligence.

What is Intelligence?

The ability of a system to calculate, reason, perceive relationships and analogies, learn from experience, store and retrieve information from memory, solve problems, comprehend complex ideas, use natural language fluently, classify, generalize, and adapt new situations.

Types of Intelligence

As described by Howard Gardner, an American developmental psychologist, the Intelligence comes in multifold:

Intelligence	Description	Example
Linguistic intelligence	The ability to speak, recognize, and use mechanisms of phonology (speech sounds), syntax (grammar), and semantics (meaning).	Narrators, Orators
Musical intelligence	The ability to create, communicate with, and understand meanings made of sound, understanding of pitch, rhythm.	Musicians, Singers, Composers
Logical-mathematical intelligence	The ability of use and understand relationships in the absence of action or objects. Understanding complex and abstract ideas.	Mathematicians, Scientists
Spatial intelligence	The ability to perceive visual or spatial information, change it, and re-create visual images without reference to the objects, construct 3D images, and to move and rotate them.	Map readers, Astronauts, Physicists
Bodily-Kinesthetic intelligence	The ability to use complete or part of the body to solve problems or fashion products, control over fine and coarse motor skills, and manipulate the objects.	Players, Dancers
Intra-personal intelligence	The ability to distinguish among one's own feelings, intentions, and motivations.	Gautam Buddha

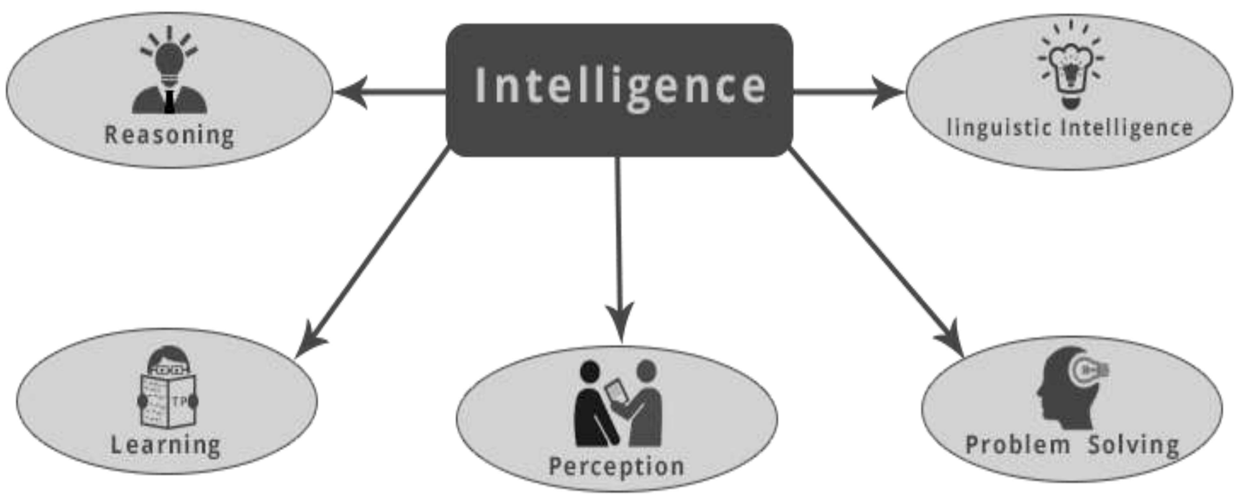
Interpersonal intelligence	The ability to recognize and make distinctions among other people's feelings, beliefs, and intentions.	Mass Communicators, Interviewers

You can say a machine or a system is **artificially intelligent** when it is equipped with at least one and at most all intelligences in it.

What is Intelligence Composed of?

The intelligence is intangible. It is composed of:

1. Reasoning
2. Learning
3. Problem Solving
4. Perception
5. Linguistic Intelligence



Let us go through all the components briefly:

1. **Reasoning:** It is the set of processes that enables us to provide basis for judgement, making decisions, and prediction. There are broadly two types:

Inductive Reasoning	Deductive Reasoning
It conducts specific observations to makes broad general statements.	It starts with a general statement and examines the possibilities to reach a specific, logical conclusion.
Even if all of the premises are true in a statement, inductive reasoning allows for the conclusion to be false.	If something is true of a class of things in general, it is also true for all members of that class.

Example: “Nita is a teacher. All teachers are studious. Therefore, Nita is studious.”	Example: "All women of age above 60 years are grandmothers. Shalini is 65 years. Therefore, Shalini is a grandmother."

2. **Learning:** It is the activity of gaining knowledge or skill by studying, practising, being taught, or experiencing something. Learning enhances the awareness of the subjects of the study.

   The ability of learning is possessed by humans, some animals, and AI-enabled systems. Learning is categorized as:

   - **Auditory Learning:** It is learning by listening and hearing. For example, students listening to recorded audio lectures.

   - **Episodic Learning:** To learn by remembering sequences of events that one has witnessed or experienced. This is linear and orderly.

   - **Motor Learning:** It is learning by precise movement of muscles. For example, picking objects, Writing, etc.

   - **Observational Learning:** To learn by watching and imitating others. For example, child tries to learn by mimicking her parent.

   - **Perceptual Learning:** It is learning to recognize stimuli that one has seen before. For example, identifying and classifying objects and situations.

   - **Relational Learning:** It involves learning to differentiate among various stimuli on the basis of relational properties, rather than absolute properties. For Example, Adding ‘little less’ salt at the time of cooking potatoes that came up salty last time, when cooked with adding say a tablespoon of salt.

   - **Spatial learning:** It is learning through visual stimuli such as images, colors, maps, etc. For Example, A person can create roadmap in mind before actually following the road.

   - **Stimulus-Response Learning:** It is learning to perform a particular behavior when a certain stimulus is present. For example, a dog raises its ear on hearing doorbell.

3. **Problem solving:** It is the process in which one perceives and tries to arrive at a desired solution from a present situation by taking some path, which is blocked by known or unknown hurdles.

   Problem solving also includes **decision making,** which is the process of selecting the best suitable alternative out of multiple alternatives to reach the desired goal are available.

4. **Perception**: It is the process of acquiring, interpreting, selecting, and organizing sensory information.

Perception presumes **sensing**. In humans, perception is aided by sensory organs. In the domain of AI, perception mechanism puts the data acquired by the sensors together in a meaningful manner.

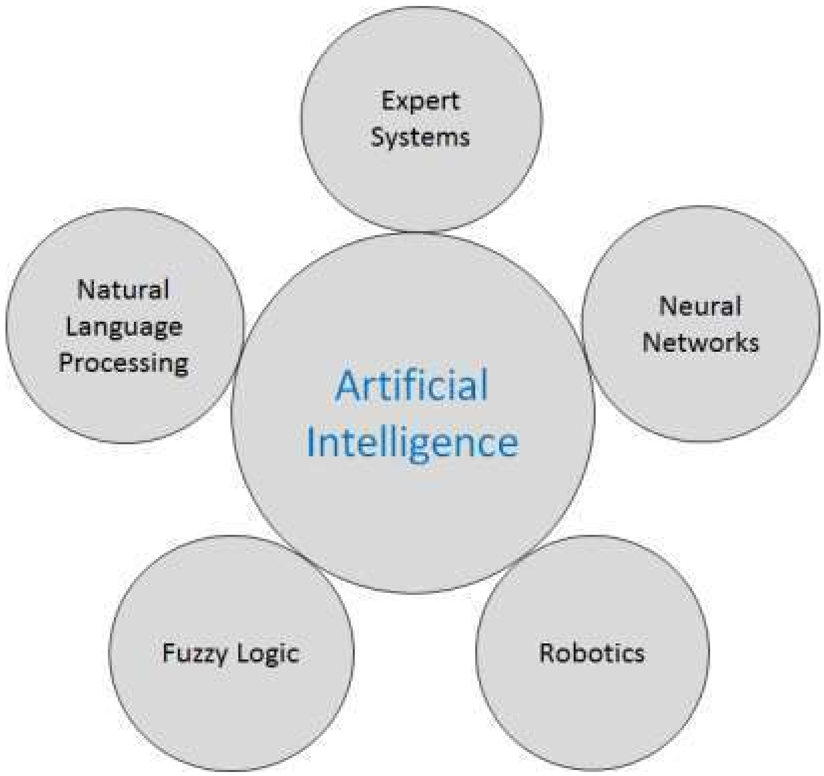
5. **Linguistic Intelligence:** It is one's ability to use, comprehend, speak, and write the verbal and written language. It is important in interpersonal communication.

Difference between Human and Machine Intelligence

- Humans perceive by patterns whereas the machines perceive by set of rules and data.
- Humans store and recall information by patterns, machines do it by searching algorithms. For example, the number 40404040 is easy to remember, store and recall as its pattern is simple.
- Humans can figure out the complete object even if some part of it is missing or distorted; whereas the machines cannot correctly.

3. Research Areas of AI

The domain of artificial intelligence is huge in breadth and width. While proceeding, we consider the broadly common and prospering research areas in the domain of AI:



Speech and Voice Recognition

These both terms are common in robotics, expert systems and natural language processing. Though these terms are used interchangeably, their objectives are different.

Speech Recognition	Voice Recognition
The speech recognition aims at understanding and comprehending WHAT was spoken.	The objective of voice recognition is to recognize WHO is speaking.
It is used in hand-free computing, map or menu navigation	It analyzes person's tone, voice pitch, and accent, etc., to identify a person.
Machine does not need training as it is not speaker dependent.	The recognition system needs training as it is person-oriented.

Speaker independent Speech Recognition systems are difficult to develop.	Speaker-dependent Speech Recognition systems are comparatively easy to develop.

Working of Speech and Voice Recognition Systems

The user input spoken at a microphone goes to sound card of the system. The converter turns the analog signal into equivalent digital signal for the speech processing. The database is used to compare the patterns to recognize the words. Finally, a reverse feedback is given to the database.

This source-language text becomes input to the Translation Engine, which converts it to the target language text. They are supported with interactive GUI, large database of vocabulary etc.

Real Life Applications of Research Areas

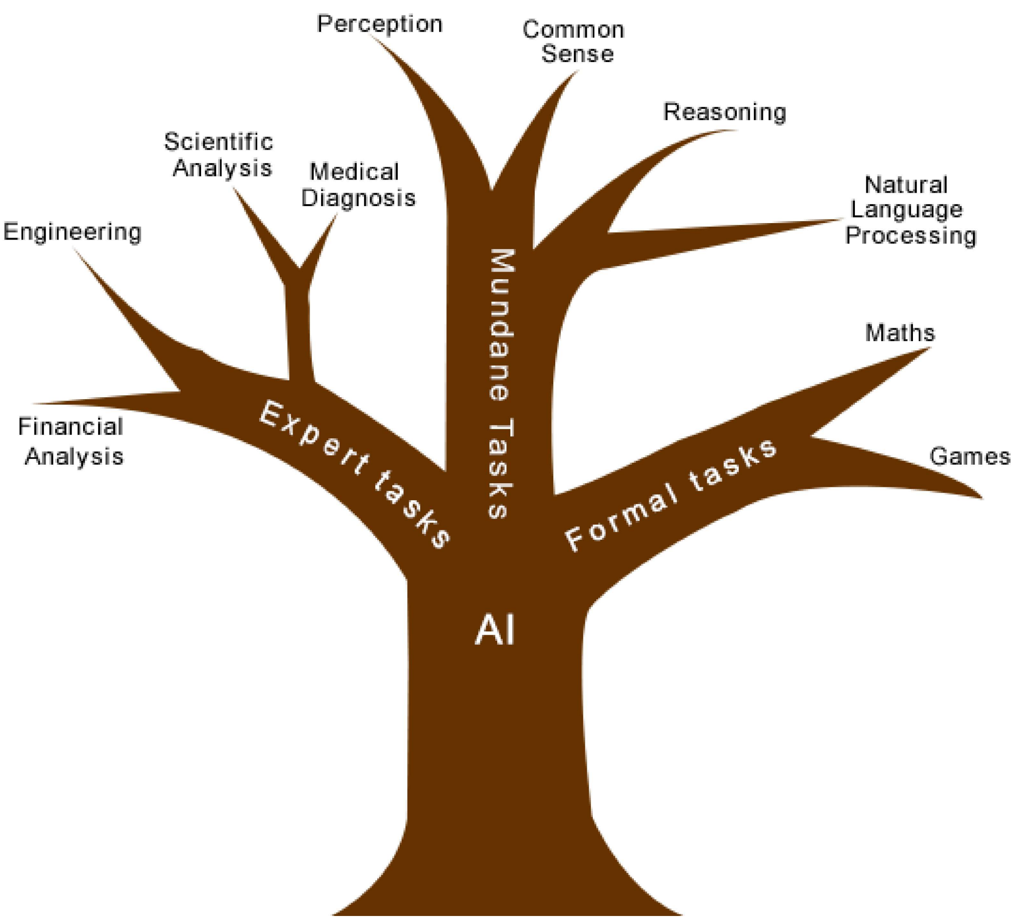
There is a large array of applications where AI is serving common people in their day-to-day lives:

Sr. No.	Research Area	Real Life Application
1	**Expert Systems** **Examples**: Flight-tracking systems, Clinical systems	
2	**Natural Language Processing** **Examples**: Google Now feature, speech recognition, Automatic voice output	
3	**Neural Networks** **Examples**: Pattern recognition systems such as face recognition, character recognition, handwriting recognition.	

4	**Robotics** **Examples**: Industrial robots for moving, spraying, painting, precision checking, drilling, cleaning, coating, carving etc.	
5	**Fuzzy Logic** **Examples**: Consumer electronics, automobiles, etc.	

Task Classification of AI

The domain of AI is classified into *Formal* tasks, *Mundane* tasks, and *Expert* tasks.



Task Domains of Artificial Intelligence		
Mundane (Ordinary) Tasks	Formal Tasks	Expert Tasks
Perception • Computer Vision • Speech, Voice	• Mathematics • Geometry • Logic • Integration and Differentiation	• Engineering • Fault finding • Manufacturing • Monitoring
Natural Language Processing • Understanding • Language Generation • Language Translation	Games • Go • Chess (Deep Blue) • Checkers	Scientific Analysis
Common Sense	Verification	Financial Analysis
Reasoning	Theorem Proving	Medical Diagnosis
Planning		Creativity
Robotics • Locomotive		

Humans learn **mundane (ordinary) tasks** since their birth. They learn by perception, speaking, using language, and locomotives. They learn Formal Tasks and Expert Tasks later, in that order.

For humans, the mundane tasks are easiest to learn. The same was considered true before trying to implement mundane tasks in machines. Earlier, all work of AI was concentrated in the mundane task domain.

Later, it turned out that the machine requires more knowledge, complex knowledge representation, and complicated algorithms for handling mundane tasks. This is the reason **why AI work is more prospering in the Expert Task** domain now, as the expert task domain needs expert knowledge without common sense, which can be easier to represent and handle.

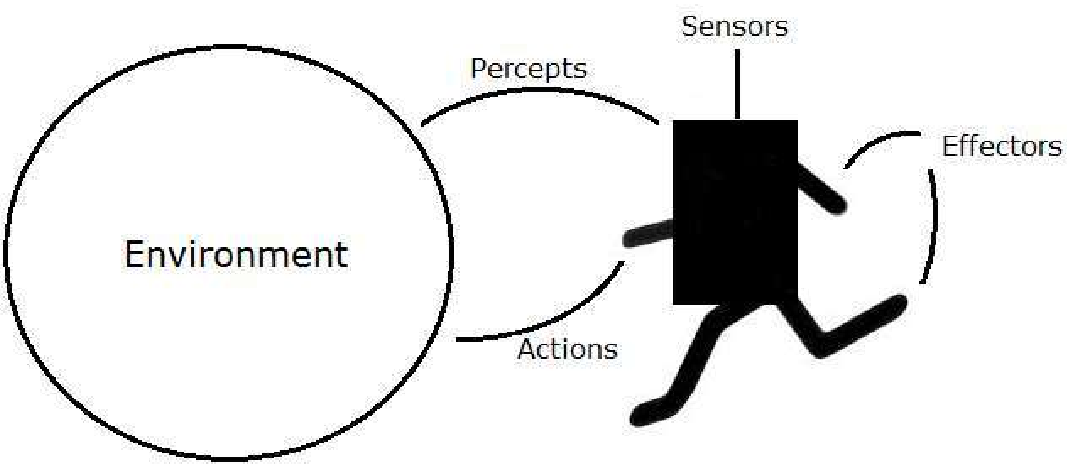
4. Agents and Environments

An AI system is composed of an agent and its environment. The agents act in their environment. The environment may contain other agents.

What are Agent and Environment?

An **agent** is anything that can perceive its environment through **sensors** and acts upon that **environment** through **effectors**.

- A **human agent** has sensory organs such as eyes, ears, nose, tongue and skin parallel to the sensors, and other organs such as hands, legs, mouth, for effectors.
- A **robotic agent** replaces cameras and infrared range finders for the sensors, and various motors and actuators for effectors.
- A **software agent** has encoded bit strings as its programs and actions.



Agents Terminology

- **Performance Measure of Agent:** It is the criteria, which determines how successful an agent is.
- **Behavior of Agent:** It is the action that agent performs after any given sequence of percepts.
- **Percept**: It is agent's perceptual inputs at a given instance.
- **Percept Sequence:** It is the history of all that an agent has perceived till date.

- **Agent Function:** It is a map from the precept sequence to an action.

Rationality

Rationality is nothing but status of being reasonable, sensible, and having good sense of judgment.

Rationality is concerned with expected actions and results depending upon what the agent has perceived. Performing actions with the aim of obtaining useful information is an important part of rationality.

What is Ideal Rational Agent?

An ideal rational agent is the one, which is capable of doing expected actions to maximize its performance measure, on the basis of:

- Its percept sequence
- Its built-in knowledge base

Rationality of an agent depends on the following:

1. The **performance measures**, which determine the degree of success.
2. Agent's **Percept Sequence** till now.
3. The agent's **prior knowledge about the environment**.
4. The **actions** that the agent can carry out.

A rational agent always performs right action, where the right action means the action that causes the agent to be most successful in the given percept sequence. The problem the agent solves is characterized by Performance Measure, Environment, Actuators, and Sensors (PEAS).

The Structure of Intelligent Agents

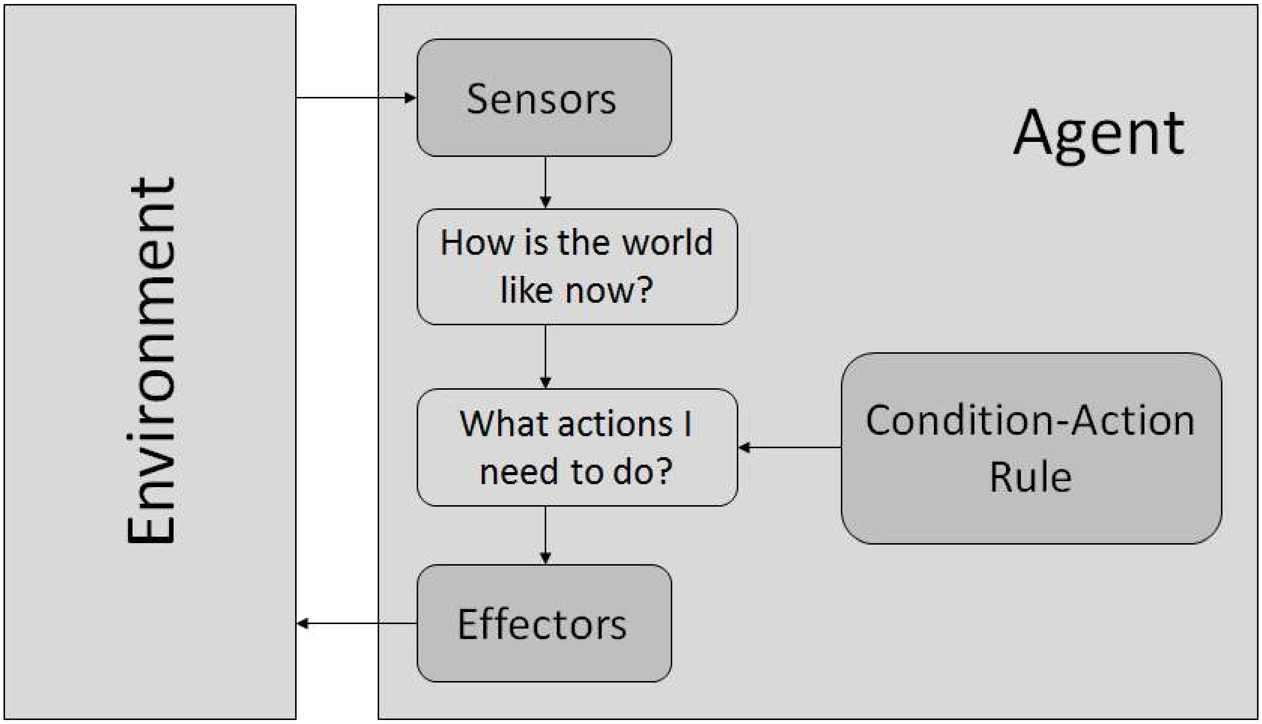
Agent's structure can be viewed as:

- Agent = Architecture + Agent Program
- Architecture = the machinery that an agent executes on.
- Agent Program = an implementation of an agent function.

Simple Reflex Agents

- They choose actions only based on the current percept.
- They are rational only if a correct decision is made only on the basis of current precept.
- Their environment is completely observable.

Condition-Action Rule – It is a rule that maps a state (condition) to an action.



Model-Based Reflex Agents

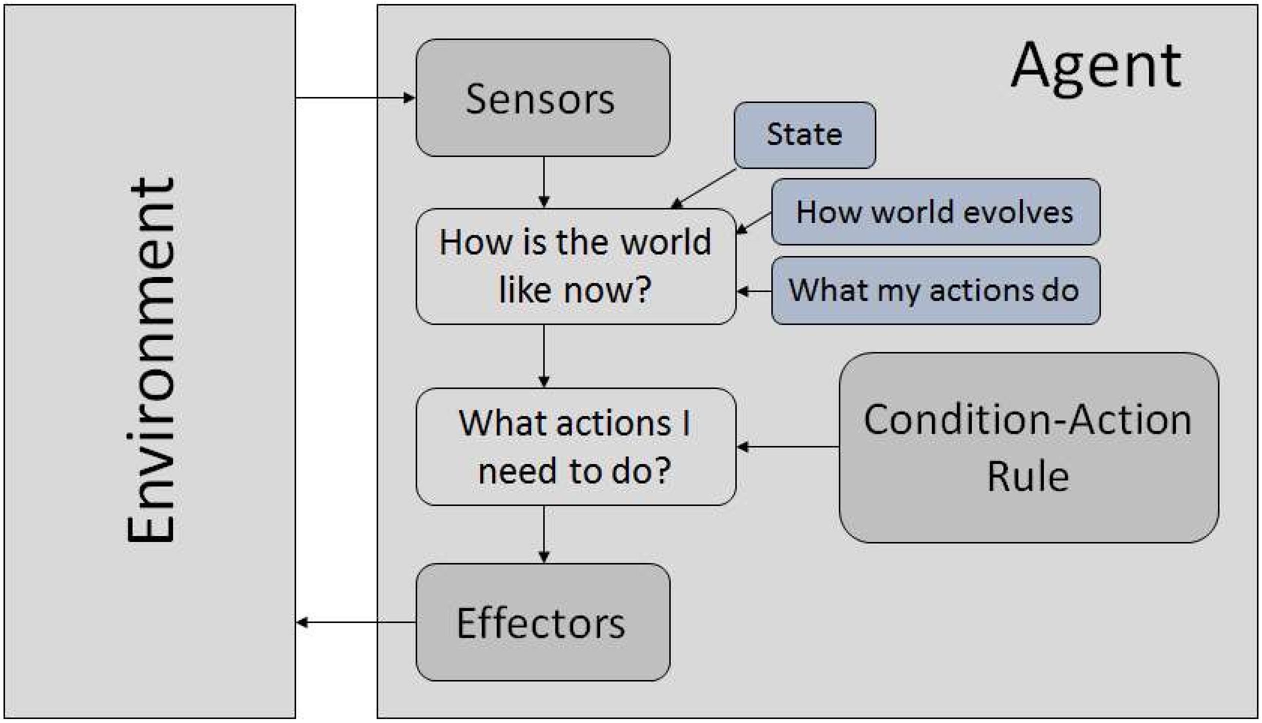
They use a model of the world to choose their actions. They maintain an internal state.

Model: knowledge about "how the things happen in the world".

Internal State: It is a representation of unobserved aspects of current state depending on percept history.

Updating state requires the information about

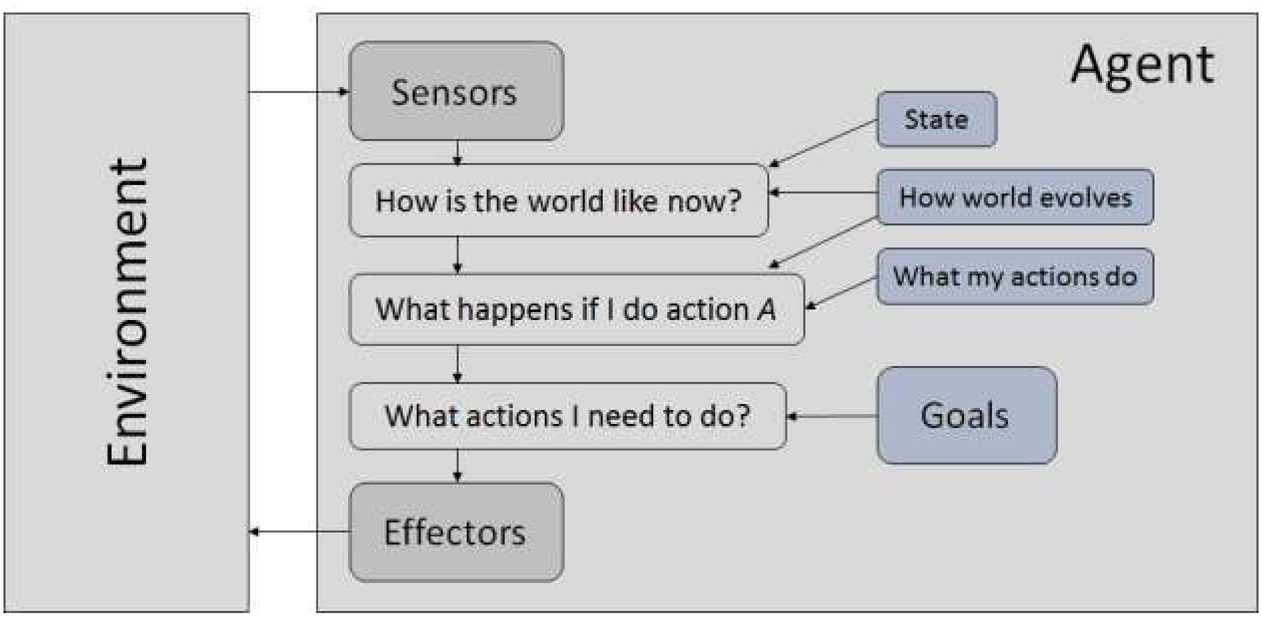
- How the world evolves.
- How the agent's actions affect the world.



Goal-Based Agents

They choose their actions in order to achieve goals. Goal-based approach is more flexible than reflex agent since the knowledge supporting a decision is explicitly modeled, thereby allowing for modifications.

- **Goal:** It is the description of desirable situations.

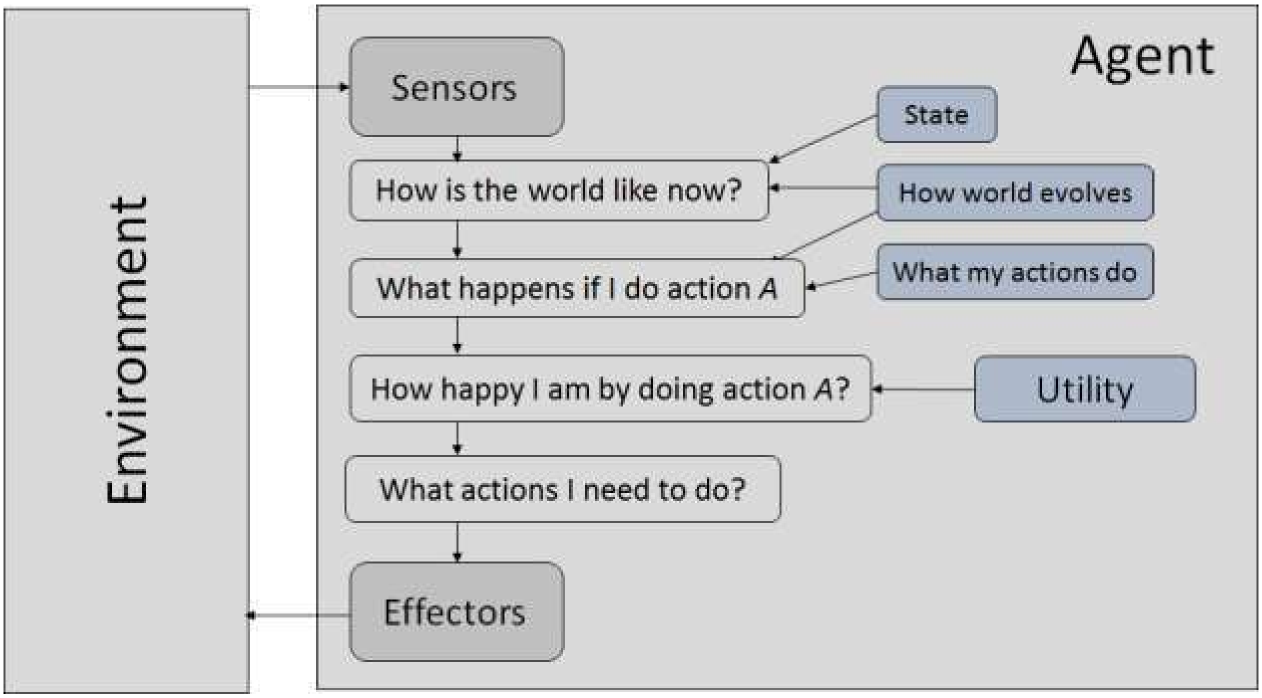


Utility-Based Agents

They choose actions based on a preference (utility) for each state.

Goals are inadequate when:

- There are conflicting goals only some of which can be achieved.
- Goals have some uncertainty of being achieved and one needs to weigh likelihood of success against the importance of a goal.



The Nature of Environments

Some programs operate in the entirely **artificial environment** confined to keyboard input, database, computer file systems and character output on a screen.

In contrast, some software agents (software robots or softbots) exist in rich, unlimited softbots domains. The simulator has a **very detailed, complex environment.** The software agent needs to choose from a long array of actions in real time. A softbot designed to scan the online preferences of the customer and show interesting items to the customer works in the **real** as well as an **artificial** environment.

The most famous **artificial environment** is the **Turing Test environment**, in which one real and other artificial agents are tested on equal ground. This is a very challenging environment as it is highly difficult for a software agent to perform as well as a human.

Turing Test

The success of an intelligent behavior of a system can be measured with Turing Test.

Two persons and a machine to be evaluated participate in the test. Out of the two persons, one plays the role of the tester. Each of them sits in different rooms. The tester is unaware of who is machine and who is a human. He interrogates the questions by typing and sending them to both intelligences, to which he receives typed responses.

This test aims at fooling the tester. If the tester fails to determine machine's response from the human response, then the machine is said to be intelligent.

Properties of Environment

The environment has multifold properties:

- **Discrete / Continuous:** If there are a limited number of distinct, clearly defined, states of the environment, the environment is discrete (For example, chess); otherwise it is continuous (For example, driving).
- **Observable / Partially Observable**: If it is possible to determine the complete state of the environment at each time point from the percepts it is observable; otherwise it is only partially observable.
- **Static / Dynamic**: If the environment does not change while an agent is acting, then it is static; otherwise it is dynamic.
- **Single agent / Multiple agents**: The environment may contain other agents which may be of the same or different kind as that of the agent.
- **Accessible vs. inaccessible**: If the agent's sensory apparatus can have access to the complete state of the environment, then the environment is accessible to that agent.
- **Deterministic vs. Non-deterministic**: If the next state of the environment is completely determined by the current state and the actions of the agent, then the environment is deterministic; otherwise it is non-deterministic.
- **Episodic vs. Non-episodic**: In an episodic environment, each episode consists of the agent perceiving and then acting. The quality of its action depends just on the episode itself. Subsequent episodes do not depend on the actions in the previous episodes. Episodic environments are much simpler because the agent does not need to think ahead.

5. Popular Search Algorithms

Searching is the universal technique of problem solving in AI. There are some single-player games such as tile games, Sudoku, crossword, etc. The search algorithms help you to search for a particular position in such games.

Single Agent Pathfinding Problems

The games such as 3X3 eight-tile, 4X4 fifteen-tile, and 5X5 twenty four tile puzzles are single-agent-path-finding challenges. They consist of a matrix of tiles with a blank tile. The player is required to arrange the tiles by sliding a tile either vertically or horizontally into a blank space with the aim of accomplishing some objective.

The other examples of single agent pathfinding problems are Travelling Salesman Problem, Rubik's Cube, and Theorem Proving.

Search Terminology

Problem Space: It is the environment in which the search takes place. (A set of states and set of operators to change those states)

Problem Instance: It is Initial state + Goal state

Problem Space Graph: It represents problem state. States are shown by nodes and operators are shown by edges.

Depth of a problem: Length of a shortest path or shortest sequence of operators from Initial State to goal state.

Space Complexity: The maximum number of nodes that are stored in memory.

Time Complexity: The maximum number of nodes that are created.

Admissibility: A property of an algorithm to always find an optimal solution.

Branching Factor: The average number of child nodes in the problem space graph.

Depth: Length of the shortest path from initial state to goal state.

Brute-Force Search Strategies

They are most simple, as they do not need any domain-specific knowledge. They work fine with small number of possible states.

Requirements –

- State description

- A set of valid operators
- Initial state
- Goal state description

Breadth-First Search

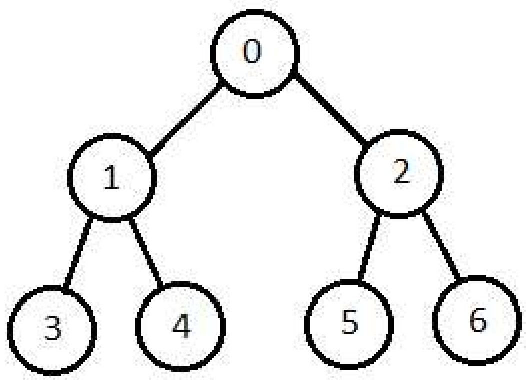
It starts from the root node, explores the neighboring nodes first and moves towards the next level neighbors. It generates one tree at a time until the solution is found. It can be implemented using FIFO queue data structure. This method provides shortest path to the solution.

If **branching factor** (average number of child nodes for a given node) = b and depth = d, then number of nodes at level d = b^d.

The total no of nodes created in worst case is $b + b^2 + b^3 + \ldots + b^d$.

Disadvantage: Since each level of nodes is saved for creating next one, it consumes a lot of memory space. Space requirement to store nodes is exponential.

Its complexity depends on the number of nodes. It can check duplicate nodes.



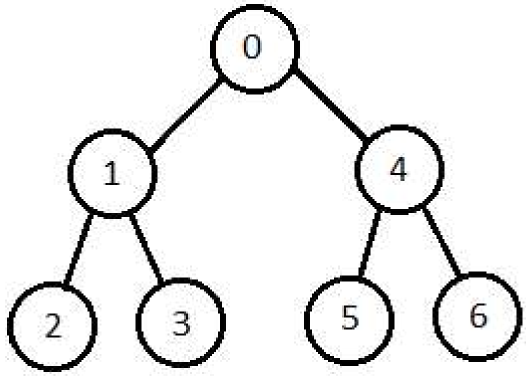
Depth-First Search

It is implemented in recursion with LIFO stack data structure. It creates the same set of nodes as Breadth-First method, only in the different order.

As the nodes on the single path are stored in each iteration from root to leaf node, the space requirement to store nodes is linear. With branching factor *b* and depth as *m*, the storage space is *bm*.

Disadvantage: This algorithm may not terminate and go on infinitely on one path. The solution to this issue is to choose a cut-off depth. If the ideal cut-off is *d,* and if chosen cut-off is lesser than *d*, then this algorithm may fail. If chosen cut-off is more than *d*, then execution time increases.

Its complexity depends on the number of paths. It cannot check duplicate nodes.



Bidirectional Search

It searches forward from initial state and backward from goal state till both meet to identify a common state.

The path from initial state is concatenated with the inverse path from the goal state. Each search is done only up to half of the total path.

Uniform Cost Search

Sorting is done in increasing cost of the path to a node. It always expands the least cost node. It is identical to Breadth First search if each transition has the same cost.

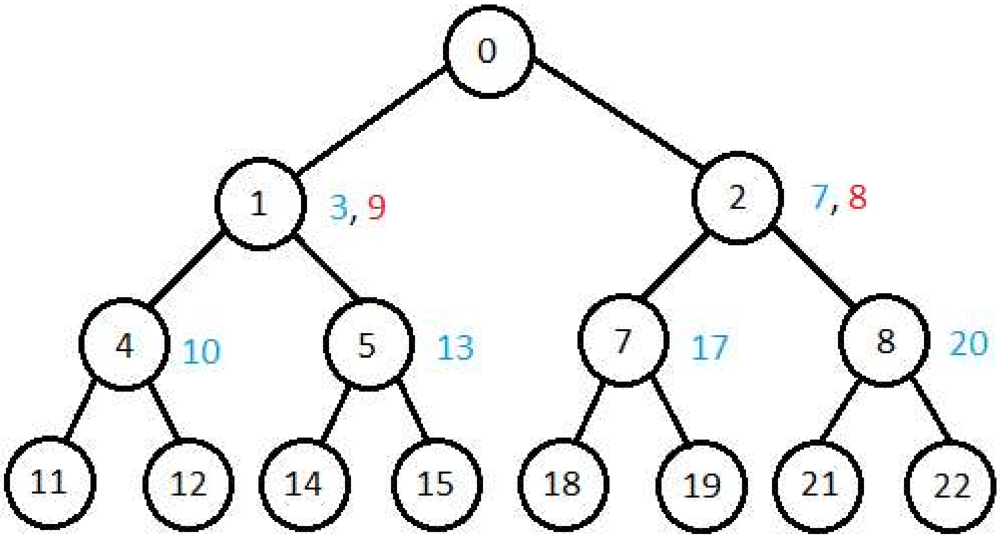
It explores paths in the increasing order of cost.

Disadvantage: There can be multiple long paths with the cost ≤ C*. Uniform Cost search must explore them all.

Iterative Deepening Depth-First Search

It performs depth-first search to level 1, starts over, executes a complete depth-first search to level 2, and continues in such way till the solution is found.

It never creates a node until all lower nodes are generated. It only saves a stack of nodes. The algorithm ends when it finds a solution at depth *d*. The number of nodes created at depth *d* is b^d and at depth d-1 is b^{d-1}.



Comparison of Various Algorithms Complexities

Let us see the performance of algorithms based on various criteria:

Criterion	Breadth First	Depth First	Bidirectional	Uniform Cost	Iterative Deepening
Time	b^d	b^m	$b^{d/2}$	b^d	b^d
Space	b^d	b^m	$b^{d/2}$	b^d	b^d
Optimality	Y	N	Y	Y	Y
Completeness	Y	N	Y	Y	Y

Informed (Heuristic) Search Strategies

To solve large problems with large number of possible states, problem-specific knowledge needs to be added to increase the efficiency of search algorithms.

Heuristic Evaluation Functions

They calculate the cost of optimal path between two states. A heuristic function for sliding-tiles games is computed by counting number of moves that each tile makes from its goal state and adding these number of moves for all tiles.

Pure Heuristic Search

It expands nodes in the order of their heuristic values. It creates two lists, a closed list for the already expanded nodes and an open list for the created but unexpanded nodes.

In each iteration, a node with a minimum heuristic value is expanded, all its child nodes are created and placed in the closed list. Then, the heuristic function is applied to the child nodes and they are placed in the open list according to their heuristic value. The shorter paths are saved and the longer ones are disposed.

A* Search

It is best-known form of Best First search. It avoids expanding paths that are already expensive, but expands most promising paths first.

f(n) = g(n) + h(n), where

- g(n) the cost (so far) to reach the node
- h(n) estimated cost to get from the node to the goal
- f(n) estimated total cost of path through n to goal. It is implemented using priority queue by increasing f(n).

Greedy Best First Search

It expands the node that is estimated to be closest to goal. It expands nodes based on f(n) = h(n). It is implemented using priority queue.

Disadvantage: It can get stuck in loops. It is not optimal.

Local Search Algorithms

They start from a prospective solution and then move to a neighboring solution. They can return a valid solution even if it is interrupted at any time before they end.

Hill-Climbing Search

It is an iterative algorithm that starts with an arbitrary solution to a problem and attempts to find a better solution by changing a single element of the solution incrementally. If the change produces a better solution, an incremental change is taken as a new solution. This process is repeated until there are no further improvements.

function Hill-Climbing (problem), returns a state that is a local maximum.

inputs: problem, a problem

local variables: *current*, a node

neighbor, a node

current ←Make_Node(Initial-State[problem])

loop

do *neighbor* ← a highest_valued successor of *current*

if Value[*neighbor*] ≤ Value[*current*] then

return State[*current*]

current ← *neighbor*

end

Disadvantage: This algorithm is neither complete, nor optimal.