1.2.1 Acting Humanly: The Turning Test Approach

The test projected by Ellen Alan Mathison Turing was designed to produce satisfactory operational definition of intelligence. Alan Mathison Turing outlined intelligent behavior, as the ability to realize human-level performance altogether in cognitive tasks, capable to fool the interrogator.

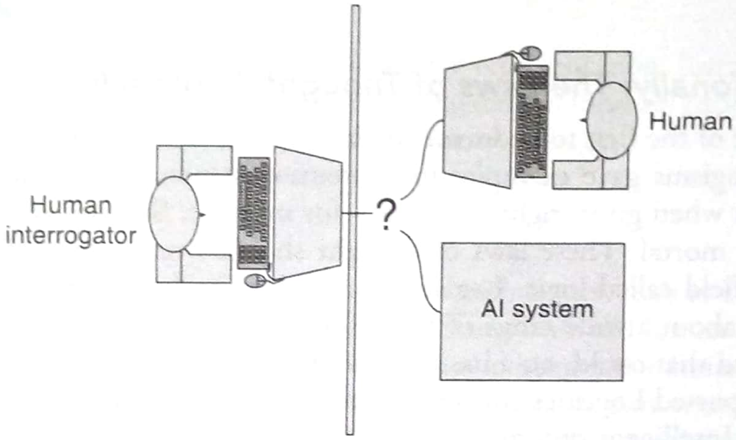


Figure 1.3 Turing's test.

The test proposed by Turing is that the computer should be interrogated by a human via a teletype. The computer is said to have passed the test if the interrogator cannot tell if there is a computer or a human at the other end as shown in Fig. 1.3.

For programming a computer to pass the test, it would need to possess the following capabilities:

1. Natural language processing to empower it to convey in English or in some other human language.
2. Knowledge representation to store data gave previously or during the cross examination.
3. Automated reasoning to utilize the static data to answer addresses and to reach inferences.
4. Machine Learning figuring out how to adjust to new conditions and to distinguish and extrapolate designs.

Turing test intentionally maintained a strategic distance from coordinate physical communication between the investigative specialist and PC on the grounds that physical reproduction off the individual is unnecessary for intelligent.

For Truing test, basic requirements of the system are:

1. Computer vision to see objects
2. Robotics to move them about

1.2.2 Thinking Humanly: The Cognitive Modeling Approach

To make a program think like a human, it is necessary to find out how humans think. The actual working of human's mind can be determined in two ways:

to resolve issues or solve problems. To begin the discussion, initially we would discuss what is normally a problem and the way it is solved.

1.1.2 Problem

In general, a problem is an obstacle that makes it troublesome/difficult to attain a desired goal, objective, or purpose. It refers to a state of affairs, condition, or issue that is however unresolved. In an exceedingly broader sense, a problem exists once an individual become tuned in to important distinction between what really is and what is desired. Each problem has an answer or solution.

1.1.3 Solution

A procedure that makes the problem navigation towards the goal is called *solution*. It can be algorithm or a hard-core implementation to achieve the goal defined by the problem.

1.1.4 Types of Problems and Solution

Depending on the way in which the problems are solved, problems are classified into the following types:

1. Structured problems
2. Unstructured problems

Structural Problems

Structural problems are the ones for which there exists a specific algorithm to achieve the goal. The same algorithm is run against variety input data still giving a guarantee of the problem being solved. Since the structure of the solution (that is in the algorithm) remains the same, even if the input data changes, these problems are called *structured problems.*

For example, find a shortest path using Prim's algorithm is a structured problem, as the same Prim's algorithm will be executed for all graphs.

In general, structured problem is solved by a solution by giving a specific algorithm and the algorithm may use some databases as its back end, by storing static data.

For example, to sort the students according to their total marks, the details about the students may be stored in the database, the sort algorithm will then get the details of students from the database and will solve the problem. Figure 1.1 represents a general solution structured problem.

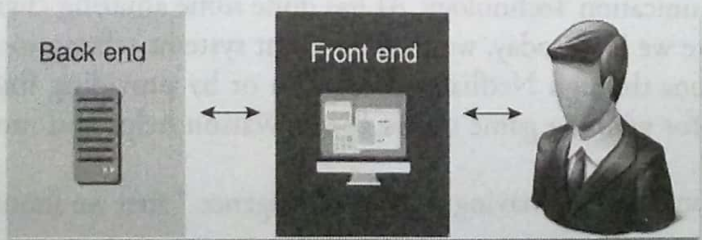


Figure 1.1 General solution structured problem.

Back end: Backend is a unit which can store data in static fashion. Static data is nothing but a fact (e.g., a backend can be a database system or a file system).

Front end: Front end is a unit that provides user interface and performs the complete business logic. It can be a program written in any programming language, such as C, C++, Java, etc. As a result, a user interacts with the front end of the system to access the backend.

Unstructured Problems

Unstructured problems are the problems for which there does not exist a specific algorithm to achieve the goal. What step to take to achieve the goal depends on what is the current state of the problem. For example, a problem of playing chess or a problem to write a program to perform heart surgery, etc., are unstructured problems, because there does not exist any specific algorithm to solve it. Such problems are solved using a Knowledge Base which is discussed in the later part of this chapter.

AI is an attempt to make a computer to solve unstructured problems. Since at this point of time these problems can be solved by human beings more better than a computer, for instance, a heart surgery can be performed by doctors better than a computer, but we want computers to perform heart surgery, that is, a computer should solve all such problems which human beings at the moment do better, this is achieved by AI.

A general solution to an unstructured problem can be represented as shown in Fig. 1.2.

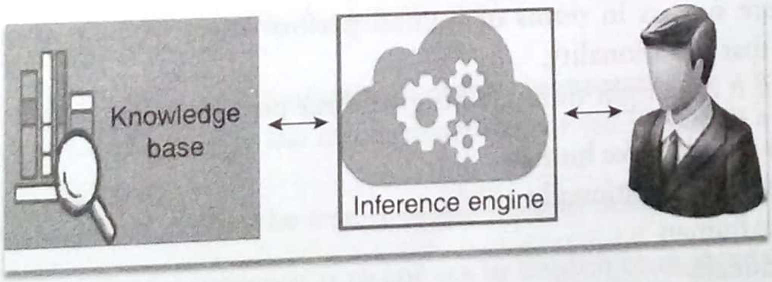


Figure 1.2 A general solution to an unstructured problem.

Knowledge base: It is the collection of facts (static data) as well as rules. Unlike database, Knowledge Base does not store only static data. Knowledge base is discussed in the later part of this chapter.

Inference engine: It is used to prove a goal (new fact) from the given facts and rules in the knowledge base. Inference engine is discussed in the later part of this chapter.

1.6 Applications of AI

1. **Game playing:** IBM's Deep Blue defeated the world chess champion Garry Kasparov in 1997. The value of IBM's stock increased by $18 billion.
2. **Mathematics:** Proved a mathematical conjecture (Robbins conjecture) unsolved for decades.
3. **Autonomous control:** The ALVINN computer vision system was trained to steer car to keep it following a lane. It was placed in CMU's NAVLAB computer-controlled minivan and used to navigate across the United States from Pittsburgh to San Diego. For 2,850 miles, it was in control of the vehicle 98% percent of the time. NAVLAB has video cameras that transmit road images to ALVIN, which then computes the best direction to steer, based on the experience from previous training run.

4. **Diagnosis:** Medical diagnosis programs based on probabilistic analysis have been able to perform at the level of an expert physician in several areas of medicine.
5. **Logistics planning:** During the 1991 Gulf War, the U.S. forces deployed an AI logistics planning scheduling program (DART) to do automated logistics planning and scheduling for transportation that involved up to 50,000 vehicles, cargo, and people.
6. **Autonomous planning and scheduling:** NASA's on-board autonomous planning program controlled the scheduling of operations for a spacecraft.
7. **Language understanding and problem solving:** Proverb is a computer program that solves crossword puzzles better than most humans.
8. **Robotics:** Many surgeons now use robot assistance like (HipNav) in microsurgery.
9. **Natural Language Generation:** NLG is a technique to produce text from computer data. It is currently used in report generation, customer service, and summarizing business intelligence insights.
10. **Speech recognition:** Speech recognition transcribes and transforms human speech into format useful for computer applications, which is used in interactive voice response systems and mobile applications.
11. **Virtual agents:** This technique has become so famous that companies call it "The current darling of the media," from simple chatbots to advanced systems that can network with humans. Virtual agents are currently used in customer service and support and as a smart home manager.
12. **Machine-learning platforms:** ML provides algorithms, APIs, development and training toolkits, data, as well as computing power to design, train, and deploy models into applications, processes, and other machines. It is currently used in a wide range of enterprise applications, mostly involving prediction or classification.
13. **AI-optimized hardware:** Graphics processing units (GPUs) and appliances are specifically designed and architected to efficiently run AI-oriented computational jobs. Currently, it is primarily making a difference in deep learning applications.
14. **Decision management:** Engines that insert rules and logic into AI systems and used for initial setup/ training and ongoing maintenance and tuning. As amateur technology, it is used in a wide variety of enterprise applications, assisting in or performing automated decision-making.
15. **Deep learning platforms:** Deep learning is special type of machine learning which have concept of neural network, it is used in pattern recognization and other many applications.
16. **Biometrics:** Biometric is bridge between human and machine which is not only using image or touch characteristics but also involves speech and other human body part recognizations.
17. **Robotic process automation:** AI is also used to create the automated machine which works same as human and also supports in many business functions.
18. **Text analytics and NLP:** Natural language processing (NLP) uses text analytics by making easier the comprehension of sentence structure and meaning, sentiment, and intent by statistical and machine-learning methods.

1.7 Categorization of AI

...thinking or processing them, and ...oning, and acting. We

Definition of AI vary along two main dimensions:

1. Thought process and reasoning
2. Behavior

Some definitions measure success in terms of human performance, whereas some measure against i concept of intelligence, that is, rationality.

A system is rational if it does right things. This gives four possible goals to peruse AI:

1. System that reasons (thinks) like human
2. System that reasons (thinks) rationally
3. System that acts like human
4. System that acts rationally

To Reason Human	*To Reason Rationally*
The exciting new effort to make computers think … machines with minds, in the full and literal sense (Haugeland, 1985). The automation of activities that we associate with human thinking, activities such as decision-making, problem solving, learning (Bellman, 1978).	The study of mental faculties through the use of computational models (Charniak and McDermott, 1985). The study of the computations that make it possible to perceive, reason, act (Winston, 1992).
To Act Human	***To Act Rationally***
The art of creating machines that perform functions that require intelligence when performed by people. (Kurzwil, 1990). The study of how to make computers do things at which, at the moment, people do better (Rich Knight, 1991).	The field of study that seeks to explain and emulate intelligent behavior in terms of computational processes (Schalkoff, 1990). The branch of computer science concerned with automation of intelligent behavior. (Luger Stubblefield, 1993).

AI attempts to understand and build intelligent entities. These constructed intelligent entities are interesting and useful as discussed in the following sections.

2.6 *PEAS* Representation for an Agent

PEAS stands for the following:

P, Performance: It shows how the system's achievements is measured.
E, Environment: It represents what the agent is interacting with.
A, Actuators: It is what produces the outputs of the system.
S, Sensors: It is what provides the inputs to the system.

In designing an agent, the first step must always be to specify the task environment as completely observable. To understand *PEAS* in a better way let us try to analyse the complex problem of automatic taxi driver which is currently beyond the capabilities of existing technology. The characteristics of *PEAS* for the description of task environment of the taxi will be considered.

2.6.1 PEAS Descriptors for Automated Taxi Driver

Figure 2.8 depicts automated taxi.

PEAS Descriptors for Automated Taxi Driver	
Performance measure	Safety, time, legal drive, comfort
Environment	Roads, other cars, pedestrians, road signs
Actuators	Steering, accelerator, brake, signal, horn
Sensors	Camera, sonar, GPS, Speedometer, odometer, accelerometer, engine sensors, keyboard

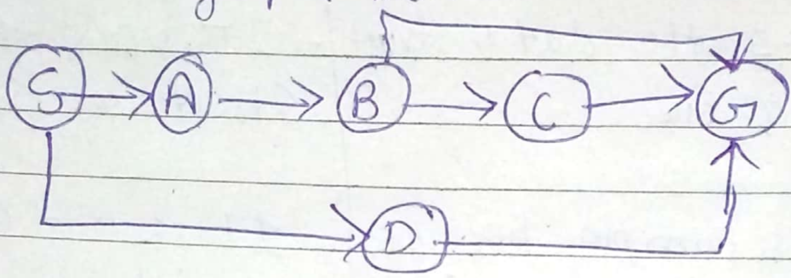
Figure 2.8 Automated taxi.

2.6.2 PEAS for Vacuum Cleaner

PEAS for Vacuum Cleaner	
Performance measure	Cleanness, efficiency: distance travelled to clean, battery life, security
Environment	Room, table, wood floor, carpet, different obstacles
Actuators	Wheels, different brushes, vacuum extractor
Sensors	Camera, dirt detection sensor, cliff sensor, bump sensors, infrared wall sensors

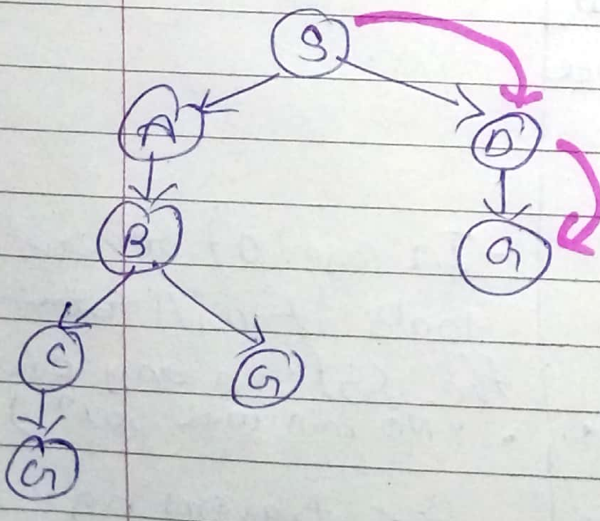
BASIS FOR COMPARISON	BFS	DFS
Basic	Vertex-based algorithm	Edge-based algorithm
Data structure used to store the nodes	Queue	Stack
Memory consumption	Inefficient	Efficient
Structure of the constructed tree	Wide and short	Narrow and long
Traversing fashion	Oldest unvisited vertices are explored at first.	Vertices along the edge are explored in the beginning.
Optimality	Optimal for finding the shortest distance, not in cost.	Not optimal
Application	Examines bipartite graph, connected component and shortest path present in a graph.	Examines two-edge connected graph, strongly connected graph, acyclic graph and topological order.

eg:- Find the Sol^n that will be used to move from node S to node G if run on the graph below



BFS

- BFS always pick the shallower branch until it reaches the Sol^n

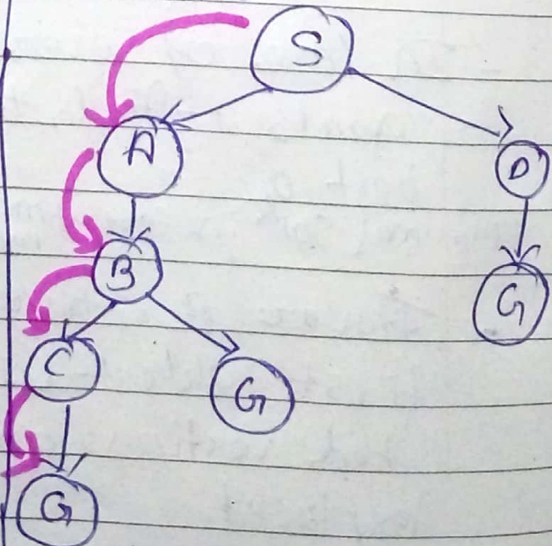


Path taken
= S - D - G

DFS

- DFS always pick the deeper branch until it reaches Sol^n

∴



Path taken is
S - A - B - C - G

QB

1) 2M

1) Represent foll^n Eng into propositional logic

a) Rani is hungry
$\Rightarrow$ P = Rani is hungry

b) If Rani is hungry, she barks
$\Rightarrow$ Q = She barks
P = Rani is hungry
$\therefore$ $P \Rightarrow Q$

c) It is hot
$\Rightarrow$ A = It is hot

d) It is humid
$\rightarrow$ B = It is humid

~~2) Represen~~ e) If it is hot then it is humid
$\therefore$ $A \Rightarrow B$

2) Represent Foll^n Eng Statement into Predicate logic / First order logic (FOL)

i) All purple mushroom are Poisionous.
$\rightarrow$ $\forall x\,(mushroom(x) \land purple(x)) \rightarrow Poisonous(x)$

ii) No Purple mushroom are Poisonous

→ ~~$\exists$~~

$\neg(\exists x)$ purple$(x) \wedge$ mushroom$(x) \wedge$ Poisonous(x)

3) All Students are Smart

$(\forall x)$ student$(x) \Rightarrow$ Smart(x)

4) Jim is not tall

→ ~~$\neg$tall(Jo~~ $\neg$ tall(Jim)

5) Every rose has an thorn

→ $\forall x$ (rose$(x) \rightarrow \exists y$ (has$(x, y) \wedge$ thorn(y)))

6) All romans ~~where~~ where either loyal to ceaser or hated him

→ $\forall x$ Roman$(x) \rightarrow$ loyal$(x$, caesar$) \vee$ hated$(x$, Caesar$)$

7) You can fool all of the people some of the time

→ $\forall x$ (person$(x) \rightarrow \exists t$ (time$(t) \wedge$ can fool(x, t)))

8) Everyone who is alike eats Something

$\forall x$ alike$(x) \longrightarrow (\exists y)$ eats(x, y)

Page No
Date

Q3) Represent in FOL & infer using forward Chaining

KB

- All cats like fish. fact 1.
- Cats eats everything they like fact 2

Data

- Ziggy is a cat fact 3

Goal :- Does ziggy eat fish?

→

FOL

i) All cats like fish

= $\forall x \; cat(x) \rightarrow likes(x, fish)$ —— ①

ii) cats eats everything they like

$(\forall x)(\forall x)(cat(x) \wedge likes(x,y)) \rightarrow eats(x,y)$

—— ②

iii) ziggy is a cat

→ $cat(Ziggy)$ —— ③

Forward chaining

By combining rules ① & ③ we get

$cat(ziggy) \Rightarrow Like(ziggy, fish)$

$Like(ziggy, fish)$ —— ④

By combining rules ②, ③ & ④ we get

$cat(Ziggy) \wedge like(ziggy, fish) \Rightarrow$

$Eats(Ziggy, fish)$

Hence, ziggy eats fish.

Q7) Backward chaining

By matching goal with consequent of rule 2, we get

cat (ziggy) $\wedge$ likes(ziggy, fish) $\Rightarrow$ eats (ziggy, fish)

$\therefore$ Sub goals are
cat(ziggy),
likes (ziggy, fish)

Now,
cat(ziggy) is resolved by rule 3

By matching subgoal likes (ziggy, fish) with consequent of rule 1

Cat (ziggy) $\Rightarrow$ likes (ziggy, fish)

Now, the subgoal cat (ziggy) is already resolved

Hence, goal is proved.

Page No
Date / /

1) ~~fr~~

Forward chaining	Backward chaining
- It is a data driven approach	- It is Goaldriven approach
- The process starts with new data & facts	- It begins with the result or goals
- The objective is to find the conclusion that ~~~~ is followed by some sequence	- The objective is to ~~find~~ find the fact that support conclusion
- It uses opportunistic approach	- It uses Conservative approach
- Flow of Forward chaining is from antecedent to consequent ~~(Initial)~~ (Incipent)	- Flow of Backward chaining is from ~~consequent~~ conclusion or consequence to incipient

Scanned by CamScanner

Eg :- Rani is hungry
If Rani is hungry, she barks
If Rani barks Raja is hungry

Step I
→ i) P = Rani is hungry
ii) Q = Rani barks
i.e. $P \Rightarrow Q$
iii) R = ~~Rani is~~ Raja is angry
$Q \Rightarrow R$

Step II
Logical connectives → $\neg, \vee$
i) P
ii) $\neg P \vee Q$
iii) $\neg Q \vee R$

Step III
Negation of Raja is angry
$= \neg R$

Step IV :-

$\cancel{\neg R}$ $\quad$ $\neg Q \vee \cancel{R}$

$\cancel{\neg Q}$ $\quad$ $\neg P \vee Q$

$\neg P$ $\quad$ P

ϕ

We arrived at the null ~~so~~ statement
Therefore
we see that Negation of the Conclusion ($\neg R$) has proved complete contradiction with given set of facts (clause)

Hence Negation is completely invalid & assertion is valid

$\therefore$ R is true

$\therefore$ Raja is angry ~~true~~ proved

III) It is humid
If it is humid then it is hot
If it is hot & humid then it will rain

~~P ⇒~~ P = It is humid
Q = It is hot $P \Rightarrow Q$
R = $P \wedge Q \Rightarrow R$
It will rain

Step 2 :-
i) P
ii) $\neg P \vee Q$
iii) $\cancel{\neg P \vee \neg Q \vee R}$
$\neg(P \wedge Q) \vee R$
$\neg P \vee \neg Q \vee R$

Step 3 :-
Negation of It will rain
$= \neg R$

Step 4 :-
$\cancel{\neg R}$ $\neg P \vee \neg Q \vee \cancel{R}$

$\neg P \vee \cancel{\neg Q}$ $\neg P \vee \cancel{Q}$

$\neg P$, P

ϕ