Table of contents

Midterm(other grp)

BACS2003 ARTIFICIAL INTELLIGENCE

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT AND TECHNOLOGY
FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
MIDTERM TEST (202405)

Name:

Programme & Tutorial Group:

Total: 50

Student Index No.:

Question 1

a) Artificial Intelligence (AI) has been applied in different domains. Provide and explain **ONE (1)** examples of how AI techniques have been involved in the medical industry. (2 marks)

b) Figure 1 shows a graph of the 10 stations to reach his destinations (from Station S to Station I). The Euclidean distances of the stations to Station I are provided in Table 1.

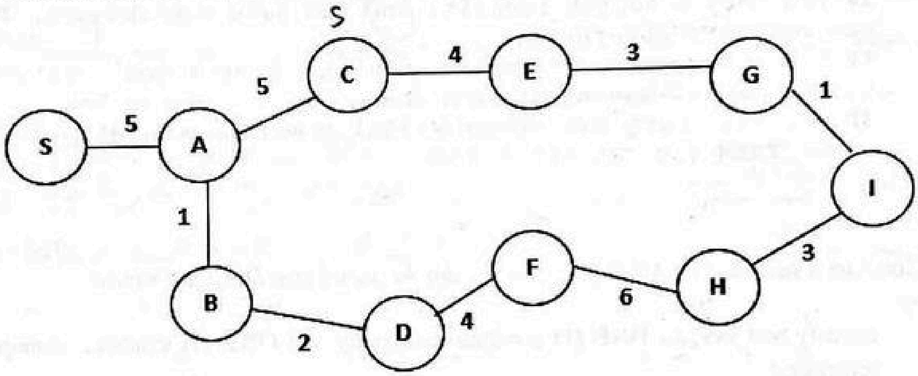


Figure 1: Search Graph

Table 1: Euclidean distance

S	A	B	C	D	E	F	G	H	I
15	22	17	23	11	12	16	10	13	0

Goal Successor Function

i. Formulate the goal and the problem above. Initial State (3 + 5 marks)

ii. Illustrate the resulting tree of depth-first search based on Figure 1. Explore the stations in alphabetical order. (5 marks)

iii. Illustrate the resulting tree with the cost and heuristic information of each step and station, respectively using A* search. (6 marks)

iv. Evaluate the performance, in terms of completeness and optimality, of the depth-first search and the A* as provided in Question 1 b) (ii) and Question 1 b) (iii). (4 marks)

This question paper consists of 2 questions on 2 printed pages.

Question 2

a) The information below describes the animals' domain knowledge.

In the world of animals, we can divide them into different groups to better understand their relationships. At the top of this system, we have two major categories: mammals and birds. Mammals include familiar pets like dogs and cats. Dogs are known for their barking, while cats are famous for their meowing. On the other hand, birds are divided into different types, such as sparrows and penguins. Sparrows are small, agile birds that can fly, whereas penguins are flightless but excel in swimming. All birds, including sparrows and penguins, lay eggs.

i. A semantic network is one of the common knowledge representation methods used to represent information. Justify why the semantic network is suitable for this domain knowledge. (2 marks)

ii. Illustrate the information above with the selected method in Question 2 a) (i). (14 marks)

b) Generate the AND/ OR graph based on the rules given below:

```
IF you have a secret identity and you have superpowers, THEN you
can become a superhero.
IF you can become a superhero and you have a cool costume, THEN
you can start saving the world.
IF you can start saving the world or you have a catchy superhero
name, THEN you can finally get your own action figure.
```

(5 marks)

c) Consider a sentence J = *Mickey said on Sunday he would give Donald a mouse.*

i. Identify and explain **ONE (1)** semantic ambiguity and **ONE (1)** syntactic ambiguity in the sentence *J*. (4 marks)

superhero cool custome

secret identify superpowers

C1: The Nature of AI

Timeline of AI

T1Q2: one key event or major achievement of A.I. development in the year of —

1960–1969

1. Unimate Introduced (1961): The first industrial robot, created by George Devol and Joseph Engelberger, was used by General Motors to lift and weld car parts.
2. ELIZA Chatbot (1966): Developed by Joseph Weizenbaum at MIT, ELIZA simulated conversation using pattern matching and mimicked a psychotherapist.
3. MacHack Chess Program (1966): Created by Richard Greenblatt at MIT, it was one of the first chess programs and the first to compete in a chess tournament.

1970–1979

1. MYCIN Expert System (1972): Developed at Stanford, MYCIN helped diagnose blood infections and showed how AI could assist in medicine.
2. Wabot-1 Humanoid Robot (1973): Built in Japan, Wabot-1 could talk, walk, and move its arms, using early AI to simulate human interaction.

1980–1989

1. Commercial Expert Systems: Used in industries like automotive and tech (e.g., XCON), expert systems helped diagnose problems and make business decisions.
2. Polly Robot (1980s): Created at MIT, Polly showed that robots could display smart behavior without detailed programming.

1990–1999

1. Sony AIBO Robot Dog (1999): A robotic pet that could respond to touch, voice, and act like a dog, showing how AI could interact emotionally with people.
2. ALICE Chatbot (1995): Built by Richard Wallace, ALICE used a simple AI language (AIML) to talk with users and helped improve chatbot research.

2000–2009

1. Recommendation Systems Grow: Sites like Amazon and Netflix used AI to suggest products and content based on user behavior.
2. IBM Watson (2007): Watson was designed to understand human language. It became famous in 2011 for beating top champions on the quiz show Jeopardy!.

2010–2019

1. Siri Launches (2011): Apple’s voice assistant could understand and respond to commands, helping users with daily tasks.
2. Cortana Released (2014): Microsoft’s voice assistant helped users manage tasks and connected with Microsoft apps like Outlook.

2020–Now

1. Rise of Generative AI:
   - Text Tools (e.g., GPT-3, ChatGPT): Generate human-like text for writing, coding, and more.
   - Image Tools (e.g., DALL·E, Stable Diffusion): Create images from text, used in design and art.
   - Code Tools (e.g., GitHub Copilot): Help programmers by suggesting or completing code.

Adoption of AI

Education • Intelligent Tutoring Systems AI tutors assist students 24/7, answering questions and offering explanations. • Automated Grading AI evaluates assignments and exams, saving teachers time.	**Entertainment** • Content Recommendations AI suggests movies, or musics based on user history (e.g., Netflix, Spotify). • Virtual Reality (VR) Experiences AI enhances immersion by adapting to user interactions.
Finance • Risk Assessment AI analyzes data to predict loan or investment risks. • Fraud Detection AI identifies suspicious transactions to prevent financial fraud. • Algorithmic Trading AI executes trades based on real-time market data analysis.	**Manufacturing** • Quality Control AI-powered cameras detect defects in products during production. • Predictive Maintenance AI predicts machine failures, reducing downtime. • Supply Chain Optimization AI streamlines logistics and inventory management.
Healthcare • Medical Imaging Analysis AI detects diseases (e.g., tumors) in X-rays or MRIs faster than humans. • Drug Discovery药物研发 AI accelerates development of new medications by predicting molecular interactions. • Remote Patient Monitoring AI tracks patient health via wearables, alerting doctors to issues.	**Food Industry** • Precision Agriculture AI optimizes crop yields using soil and weather data. • Food Quality Inspection AI ensures food safety by detecting contaminants. • Inventory Management AI reduces waste by predicting demand and optimizing stock.
Marketing • Customer Segmentation AI analyzes data to identify target audiences for tailored campaigns. • Predictive Analytics AI forecasts sales trends and customer behavior. • Personalized Advertising AI customizes ads based on user preferences and behavior.	**Designing** • Design Automation AI generates design prototypes (e.g., logos, layouts) quickly. • Creative Inspiration AI suggests color schemes, fonts, or layouts based on trends. • User Experience (UX) Optimization AI analyzes user behavior to improve product design.

T1Q3: Name an artificial intelligence application that is created by a Malaysian company. Briefly describe its A.I. functions.

AskAILA is an artificial intelligence (AI) legal assistant developed by the Malaysian law firm Shang & Co.

It uses AI to provide legal advice, especially in labor law, and is available 24/7. Trained on Malaysian labor laws and regulations, AskAILA helps make legal services more affordable and accessible.

Define AI

John McCarthy - making a machine to behave in ways that would be called intelligent if a human were so behaving 让机器以人类可以称为智能的方式行事

T1Q1: father of A.I.? reason

John McCarthy

1. made important contributions to computer science and artificial intelligence.
2. In 1956, he created the term "artificial intelligence", which he described as "the science and engineering of making intelligent machines." He introduced this idea at the Dartmouth Conference.
3. The goal of the conference was to find ways to build machines that could think like humans, solve problems, and improve themselves.

Type of AI Systems(4)

T1Q4: Type of system(4) + Example

1. System that Thinks Like Humans – The Cognitive Modelling Approach	2. System that Acts Like Humans – The Turing Test Approach
Automation, Chatbot • study on how computer models could be used to address the psychology of memory, language, and logical thinking • try to think like people by learning, recognizing patterns, and reasoning. • aim to mimic how the human brain works by understanding how we solve problems, remember, and make decisions—just like in cognitive science. • 🧠 Example: **IBM's Watson** Watson uses machine learning and natural language processing to simulate human thinking. It can understand and respond to complex human language, like giving medical support.	Machine learning, Recommender • aim to act like a human would, even if they don't actually understand like a human. • focus on interacting naturally—like speaking in human language or doing physical tasks. • 🗣️ Example 1: **Siri or ChatGPT** These AI assistants respond to voice or text like a person—answering questions, setting reminders, and chatting. • 🧹 Example 2: **Roomba** The robotic vacuum moves around the house, avoiding obstacles and cleaning like a person would.
3. System that Acts Rationally – The Agent-Based Approach	4. System that Thinks Rationally – The Logic-Based Approach
Adaptive Systems, Planning & Optimisation • Agent is something that acts autonomously, sensitive (sense) to its environment, adapt to change, and create/pursue goals	Expert Systems • Rational thinking = Logic • Logic uses a process of inference to derive new representations about the world, and use these new representations to deduce

• **make decisions based on goals** and act to achieve them efficiently. • sense their environment and respond smartly, even if they don't act like humans. • 🚗 Example**: Self-Driving Cars** These cars collect data from sensors, then make safe and smart driving decisions—like stopping for pedestrians or changing lanes.	what to do. Example: Socrates is a man; all men are mortal; therefore Socrates is mortal • **use logical rules** to solve problems. • They think in terms of "if-then" rules and symbols—similar to how a math problem is solved. • 🧪 Example: **MYCIN** An expert system used in medicine. It followed rules to diagnose infections and suggest treatments based on symptoms.

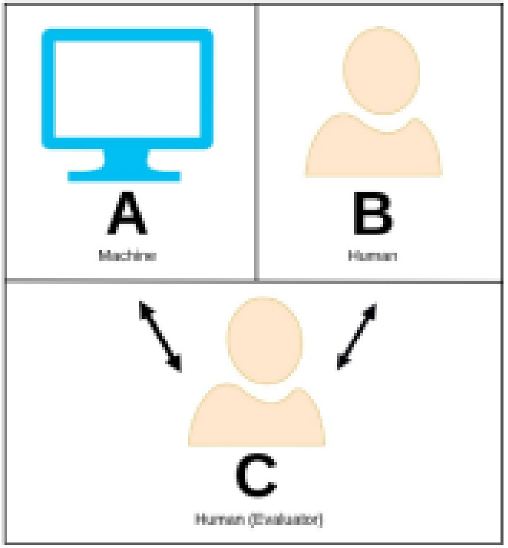
T1Q5: input (conditions), processing layer, output and how the agent react for each of the system below:

Agent of an automated class scheduling system.	**Agent of an intelligent air-conditioning system.**	**Agent of autonomous driving car.**
Input Layer: • Course requirements (subjects, time slots, prerequisites) • Faculty availability • Classroom capacity and resources • Student preferences and availability **Processing Layer:** • **Knowledge Base:** Stores scheduling rules (e.g., avoid time clashes) • **Inference Engine:** Applies rules to match inputs into valid combinations • **Conflict Resolution Module:** Resolves overlapping classes or limited room resources **Output Layer:** • Complete class schedule • Suggestions for conflict resolutions or alternate times **Reaction:** The agent collects input, applies rules and logic to avoid conflicts, and outputs a schedule that fits the constraints of teachers, students, and	**Input Layer:** • Current room temperature (via sensors) • User preferences (target temperature, humidity) • Outdoor weather or environmental conditions **Processing Layer:** • **Control Algorithm:** Calculates needed adjustments for comfort • **Learning Module:** Learns user patterns to make smarter future changes **Output Layer:** • Adjusted AC settings (temperature, fan speed, mode) • Notifications or updates (e.g., "Cooling to 24°C") **Reaction:** The system senses the environment and user preferences, processes them in real-time, and automatically adjusts settings for comfort and energy efficiency.	**Input Layer:** • Sensors (camera, LiDAR, radar, GPS, etc.) to detect surroundings • User input (destination, driving mode) • External data (traffic info, weather, maps) **Processing Layer:** • **Perception Module:** Understands surroundings (lanes, vehicles, pedestrians) • **Decision Module:** Plans driving actions (stop, go, turn) • **Control Module:** Executes actions via brakes, throttle, steering **Output Layer:** • Vehicle movements and route navigation • User updates (e.g., "Turning left in 200 meters") **Reaction:** The car constantly reads its environment and incoming data to decide how to act, adjusting its movement safely and efficiently as conditions change.

classrooms.		

Turing Test 图灵测试

T2Q1:



The Turing Test typically involves three participants:

1. A human judge (interrogator): This person conducts the test by engaging in a conversation with both a human and a machine.
2. A machine: The AI being tested for its ability to exhibit intelligent behavior.
3. A human: To provide a baseline for comparison with the machine's responses.

The judge communicates with both the human and the machine via a text-based interface, such as a computer keyboard and screen, so the judge cannot see or hear them directly. This ensures that the evaluation is based solely on the responses and not on physical appearance or voice. During the test, the judge asks questions and engages in a conversation with both the human and the machine. The machine's goal is to respond in a manner that is indistinguishable from the human participant. If the judge cannot consistently tell which participant is the machine, the machine is considered to have passed the Turing Test (Intelligent)

A test to determine if a machine can exhibit intelligent behavior indistinguishable from a human.

Turing Imitation Game

Standard Interpretation of the Turing Test:

- A human interrogator communicates with two hidden players: A (a machine) and B (a human).

- The interrogator's goal is to determine which is the machine.
- If the machine can successfully fool the interrogator into thinking it is human, it passes the test.
- Conclusion: If a machine can imitate human responses well enough to deceive a human, it is considered intelligent.

APPLICATION: CAPTCHA

CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart):

- Used to prevent bots from abusing websites.
- Tests often include distorted text or images that humans can recognize but machines struggle with.
- If a software can solve the CAPTCHA, it is assumed to be human-like, as it passed a form of Turing Test.

LOEBNER PRIZE – A Real-World Turing Test

The Loebner Prize is the first formal competition based on the Turing Test.

Started in 1990 by Hugh Loebner in partnership with The Cambridge Center for Behavioral Studies.

- It challenges AI programs to convince judges that they are human through conversation.
- Goal: To identify the most human-like AI systems through real-time interaction.

Chinese Room

The "Chinese Room" is a thought experiment proposed by philosopher John Searle to critique评判 the idea that computers or AI can truly understand language or possess consciousness具有意识.

Imagine someone in a room who doesn't know Chinese. They get Chinese symbols as input, use a rulebook to reply, and the output looks intelligent. But the person doesn't actually understand Chinese.

This is like AI: it can mimic模仿 understanding理解 (e.g., chatbots), but it doesn't truly understand language.

Example:

Rule 1: If someone asks "Wie geht es Ihnen?" (How are you?), the reply is "Mir geht es gut" (I'm fine).

Rule 2: If someone says "Auf Wiedersehen" (Goodbye) or "Wiedersehen" (Farewell), the reply is "Tschüss" (Bye).

Question: What happens if someone says "Wiedersehen"? Answer: The system replies "Tschüss" (because Rule 2 matches "Wiedersehen").

Think(in lecnote)?

If the system clearly runs a program and passes the Turing Test, does it really understand anything of its inputs and outputs?

No. Passing the Turing Test only means the system can mimic human-like responses. It doesn't necessarily understand the meaning of inputs or outputs.

Is it necessary for it to understand the inputs and outputs?

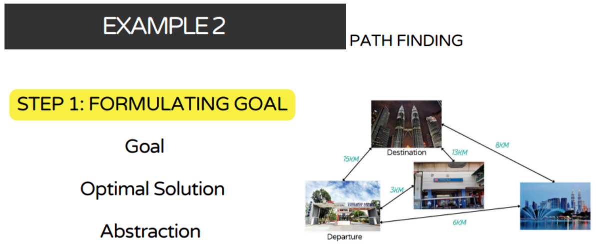
No, it's not required for basic functionality. The system can work effectively by following rules or patterns without true understanding. However, understanding might be necessary for advanced AI goals like consciousness or creativity.

C2: Problem Definition and Problem Solving

Problem-Solving Concept (3)

1. Goal Formulation

- First step in problem solving (Defining the desired outcome.)
- **Goal**(outcome?) **- Optimal Solution**(best solution?) **- Abstraction**(scope?)
- Path Finding:



2. Problem Formulation

- Process of deciding what actions and states to consider

Initial State

- the state that the agent starts in

E.g. Go(_ , In(TARUMT), 0)

Successor Function

- The possible actions available to the agent, that will change the state

  - Go(To(Wangsa Maju, In(Tarumt), 3)
  - Go(To(Titiwangsa, In(Tarumt), 6)

State Space

- The set of all states reachable from the initial state

Goal Test

- To determine whether a given state is a goal state

E.g. current state == goal state

Sample of Goal Test Algorithm

E.G.

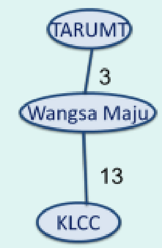
```
If
  Go( _ , In(KLCC), _)    #current state == goal state
Then
  show pathcost

return true
```

Step Cost

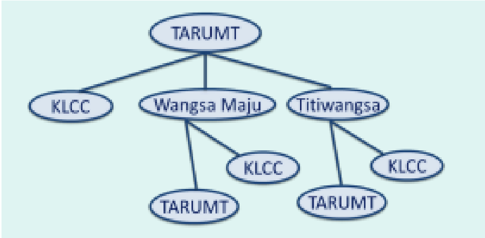
- Route distance, from one state to another state

E.G.
the step cost from TARUMT to WangsaMaju = 3km



Path Cost

- Sum of the costs of the individual actions along the path

<table>
<tr>
<td></td>
<td>

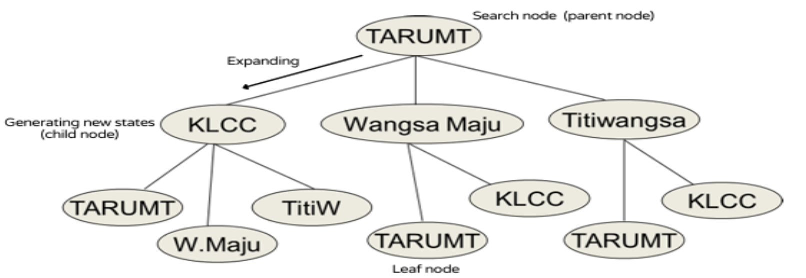
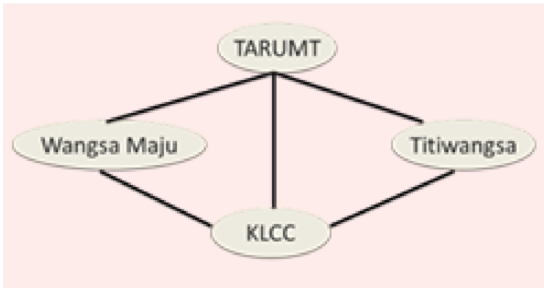
Path
Sequences of states connected by a sequence of actions
E.G. TARUMT --> WM ---> KLCC

</td>
<td></td>
<td>E.G.
From TARUMT to KLCC :16 = 3+13

</td>
</tr>
</table>

3. Search Solution Execution (3)

- Process of looking for sequence of actions
- Search - the process of looking for the best sequence of path
- Solution - A search algorithm takes a problem as input and returns a solution in the form of an action sequence
- Execution - Once a solution is found, the actions it recommends can be carried out.

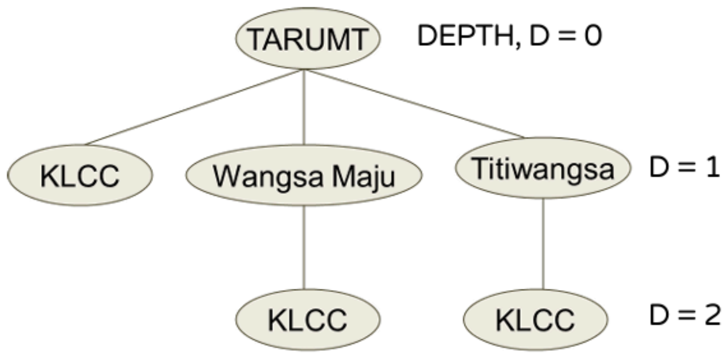
<table>
<tr>
<td>Search tree

</td>
<td>Search graph

</td>
<td>Expanding/ generating states</td>
<td>Search strategy

Uninformed search

Informed search</td>
</tr>
</table>

<table>
<tr>
<td>Generated by the initial state and the successor function that together define state space.

AVOIDING REPEATED STATES
• Wasting time
• Can cause a solvable problem become unsolvable if the algorithm does not detect them

SEARCH TREE WITHOUT REPEATED STATE
</td>
<td>The same state can be reached from multiple paths.</td>
<td></td>
<td></td>
</tr>
</table>

MEASURING PROBLEM-SOLVING PERFORMANCE

Completeness	Is the algorithm guaranteed to find a solution when there is one? 当存在解时，该算法能保证找到解吗？
Optimality	Does the strategy find the optimal solution?
Time complexity	How long does it take to find a solution?
Space Complexity	How much memory is needed to perform the search?

T3Q2

1. **Figure 1** shows a puzzle problem that requires rearrangement of the tiles to transform the order from start to goal state. One is only permitted to slide the empty tile follow the sequence of **left**, **right**, **up** or **down.**	a) Goal formulation Goal: to reach the goal state as shown in Figure 1

Remark: Avoid repeated state

Figure 1: The Puzzle Problem

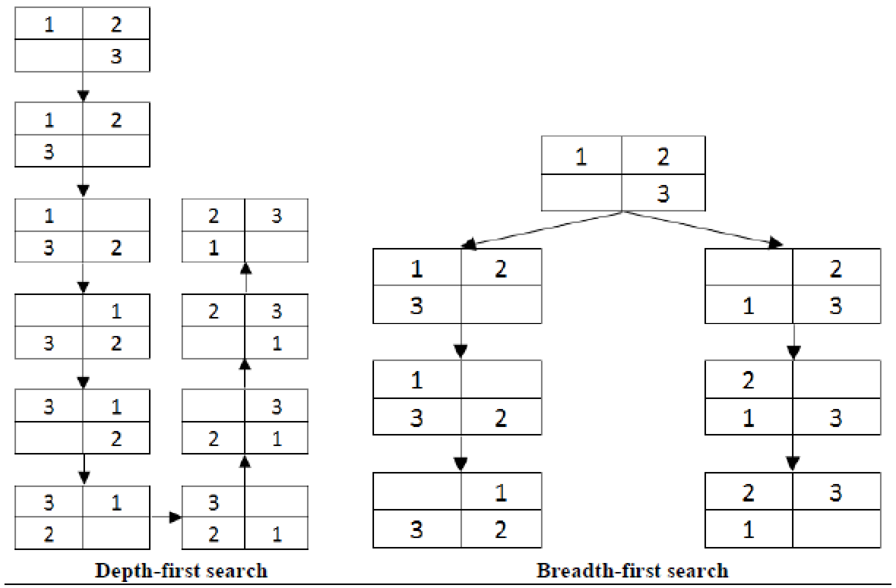
a) Provide the goal formulation of the puzzle problem above.
b) (Problem Formulation) Formulate the puzzle problem above by specifying the initial state, successor functions, goal test, step cost, and path cost.
c) Perform **breadth-first search** and **depth-first search** on the puzzle problem above. Draw the resulting search trees for both.

2	3
1	

Optimal solution: find the shortest path to reach the goal
Abstraction: time, material

b) Problem formulation
Initial state: Start state
Successor function: All possible movements to successor state from current state
Goal test – to check if the goal state has been reached, otherwise continue to search
Step cost – the distance between one node to the other, e.g. each step is 1.
Path cost – the total step cost between start state and the goal state along a path

c)



1. The following graph in Figure 1.1 shows all the nodes in a telecommunication network. The distance (in km) from one node to another is shown on the arc. (9)

a)

Goal	Node G
Optimal Solution	The shortest path used to send data from node S to node G.
Abstraction	Ignore the chance that some data might be lost during transmission.

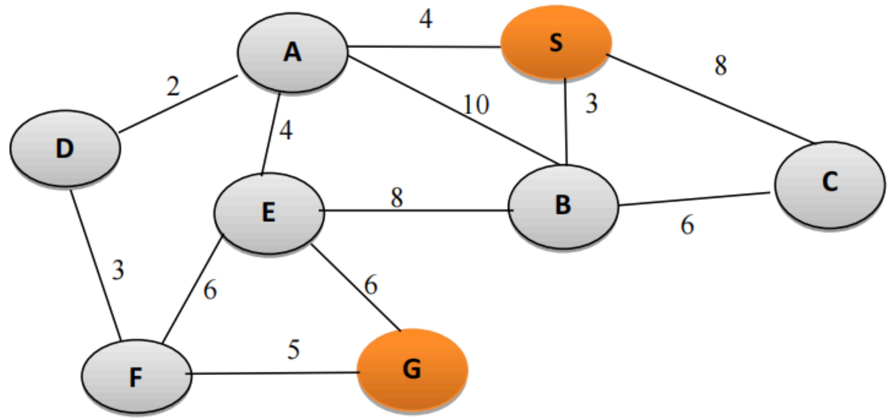


Figure 3: A search graph of a new LRT network

The Euclidean distance (in km), which is used as the heuristic cost (h) for different node, is provided in Table 1 below.

S	A	B	C	D	E	F	G
14	10	13	8	8	11	5	0

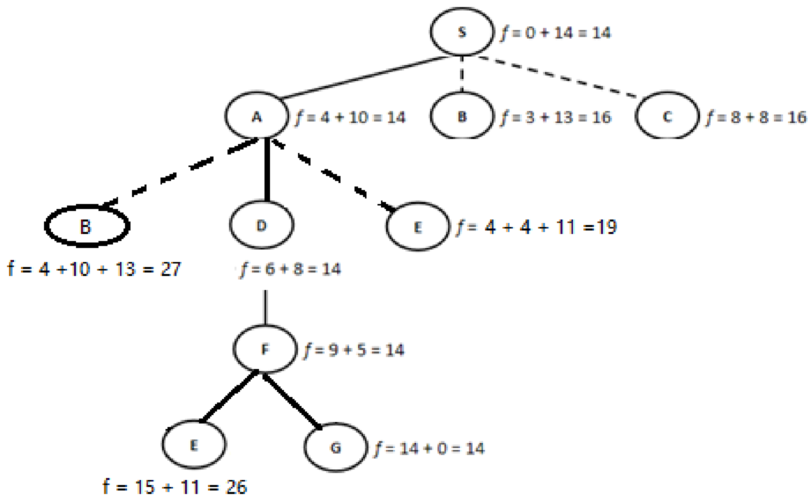
a) Assume that some data are to be sent from node S to node G using the shortest route. Describe the goal formulation and problem formulation (initial state, successor function, goal test, step cost and path cost).

b) Show the resulting search tree of A* search to find the shortest path from S to G. State the shortest path. **(Remark: Avoid repeated state)**

c) Evaluate the efficiency of A* search in solving the path-finding problem above

Initial state	Node S
Successor function	The possible actions available to the agent that will change the state from the current state OR a function to generate a successor state (output) from a current state (input).e.g From A generates S etc. OR all possible movement available to the agent.
Goal test	A test to check if a node is node G.
Step cost	The cost of transforming a state to another state, indicated by the value of an edge connecting any two nodes. E.g., the step cost of transforming S to A is 4.
Path cost	The total step costs along a path. E.g., the path cost for S-A-D is 4 + 2 = 6.

b)



The shortest path discovered b A* is S-A-D-F-G

C3: Uninformed Search (Blind Search) (5)

Uninformed search methods explore state spaces without prior knowledge of the solution.

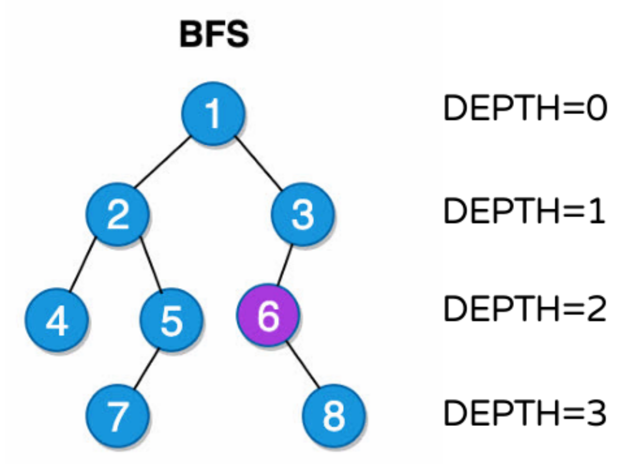
1. Breadth-first search (BFS): Explores all nodes at the current depth before moving to the next level.
2. Depth-first search (DFS): Explores as far as possible along a branch before backtracking.
3. Depth-limited search (DLS): Limits the depth of DFS to avoid infinite paths.
4. Iterative deepening (IDS): Combines BFS and DFS by gradually increasing depth limits.
5. Bidirectional search: Searches from both initial and goal states.

Measuring Problem-Solving Performance

	Breadth-first search	Depth-first search
Complete	Complete – It will always find a solution if one exists, by exploring all possible paths level by level.	Not always complete – May get stuck in infinite loops or miss paths, depending on the problem and implementation.
Optimality 最优性	Optimal – Guarantees the shortest path in terms of edges or steps.	Not optimal – Might find a solution faster, but not necessarily the shortest.
Time Complexity	Both have the same: O(V + E) with an Adjacency List $O(V^2)$ with an Adjacency Matrix Where V = vertices, E = edges	
Space Complexity	Higher – Needs to store all nodes at the current level.	Lower – Only needs to store the path from the root to the current node, limited by recursion depth.

<table>
<tr>
<td>

[1- Breadth-first search]



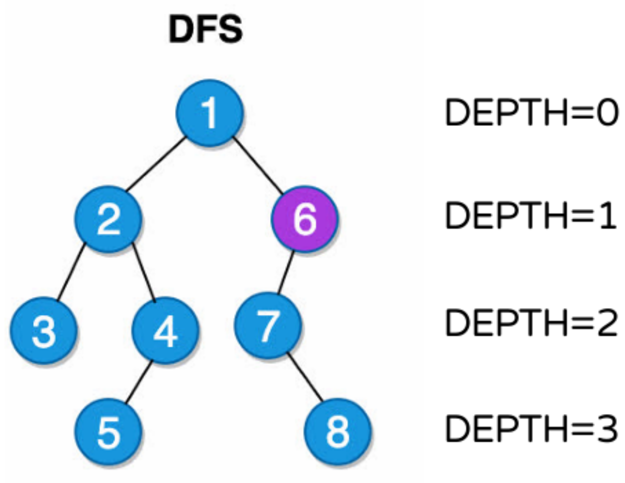
</td>
<td>

All the nodes are expanded at a **given depth** in the search tree **before any nodes at the next level are expanded**. 左到右看

ALGORITHM
1. Begin with the root node.
2. Examine all of the nodes in each level before moving on to the next level.
3. Repeat the process and work downward from left to right until a solution is found.
4. Return Fail if there is no solution

</td>
</tr>
<tr>
<td>

[2- Depth-first search]



</td>
<td>

Always **expands the deepest node in the current fringe** of the search tree.
Nodes leading from dead end normally will be discarded from memory.
Then the search will **back up to the next shallowest node** that still has unexplored successors.
Implemented by calling tree search using LIFO (last-in-first-out) strategy.

</td>
</tr>
<tr>
<td>

[3- Depth-limited search]

</td>
<td>

👣 How it works: Like DFS, but you only go a certain depth (level). After that, you stop going deeper.
📦 Uses: Stack + depth check.
✅ Good for: Problems where you know the solution is not too deep.
🔁 Steps in DLS:
Set a depth limit, say 3.

</td>
</tr>
</table>

	Run DFS, but if you reach level 3 → stop exploring that path. 🧪 Use when: You want to avoid infinite paths or loops.
4- Iterative deepening depth-first search	**Like BFS results, but saves memory like DFS.** 👣 How it works: Do DLS repeatedly, starting from 0 and increasing the depth by 1 each time. 📦 Uses: Stack but in loops. ✅ Good for: When you want shortest path but with low memory. 🔁 Steps in IDDFS: Run DLS with limit = 0 (too shallow). Run DLS with limit = 1. Run DLS with limit = 2. Continue until you find the goal. 🧪 Use when: You want a good solution but don't want to store too many nodes. In depth-limited search, there is no reliable method to decide a depth limit to include at least one solution.Iterative deepening can solve this by checking the entire tree by gradually increasing the depth limit-first depth O, then depth 1, then depth 2 and so on.
5- Bidirectional search	Run two simultaneous searches: 1. One forward from the initial state 2. One backward from the goal (final state) Stop searching when the two searches meet in the middle

Performance Evaluation

Search	Memory	Finds Shortest Path?	Good For
BFS	High	✅ Yes	Shortest path problems
DFS	Low	❌ No	Quick but not always best
DLS	Low	❌ No	DFS with depth control
IDDFS	Medium	✅ Yes	Best of DFS and BFS
Bidirectional	Medium	✅ Yes	When both start & goal known

C4: Informed Search (Heuristic Search)

Heuristic Search : Heuristics rank alternatives based on available information.

Heuristic function h(n) estimates the cost to reach the goal.

Techniques

1. Hill climbing: Simple and steepest ascent variants.
2. Best-first search: Prioritizes nodes based on heuristic values.
3. A* search: Combines path cost and heuristic to find optimal solutions.

Limitations

Hill climbing may get stuck in local maxima, plateaus, or ridges.

Heuristics are not always perfect and may lead to suboptimal solutions.

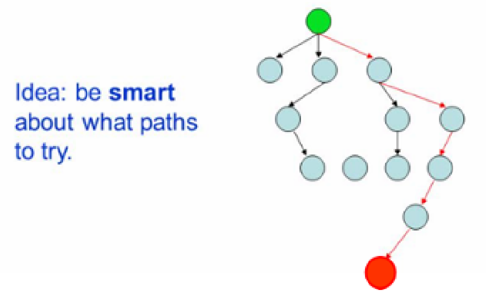
Analysis

Informed search uses heuristics to guide exploration, making it more efficient than uninformed search.

Hill climbing is simple but limited by local optima, while A* search balances cost and heuristics for optimality.

Informed (Heuristic) Search

A **function that ranks alternatives** at each branching step based on available information **to decide which branch to follow.**
The agent is informed which branch(es) in a state space that is(are)most likely to lead to an acceptable problem solution.
Heuristic often design based on experience or intuition
Heuristic is only an informed guess of the next step to be taken– seldom able to predict the exact behavior of the state space.
Can lead to a suboptimal solution or fail

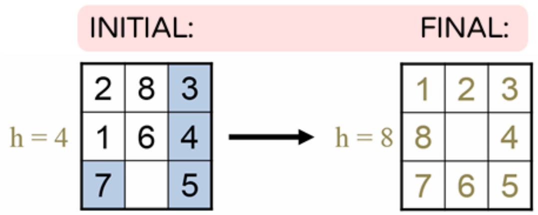


Heuristic Function

h(n) indicates how “close” a state n is to the goal

EXAMPLE

Let the heuristic function, h(n), is given as follows:
h(n) = number of tiles at the right place.
* h(n) determines the heuristic cost of a state



Heuristic Search Techniques

1. Hill climbing (simplest way)
❌less combinatorially explosive as it searches locally rather than globally
❌possibly very inefficient and ineffective
Limitation of Hill Climbing
1a. Simple Hill Climbing
1b. Steepest Ascend Hill Climbing

Designing a Heuristic Function (2)

Local Heuristic	Global Heuristic - Better than local heuristic but challenging to design

Hill climbing is not always effective
Heuristic function not always perfect
Modify the heuristic function
Inefficient in a large, rough problem space

2. Best-first Search (Depth+Breadth)

3. A* Search
Function Cost f(n) = g(n) +h(n)

OR
Look for Minimum f(n) and h(n)=0

Hill Climbing (2)	**Best-first Search**	**A* Search**

C5: Knowledge Representation

Knowledge Representation Tools

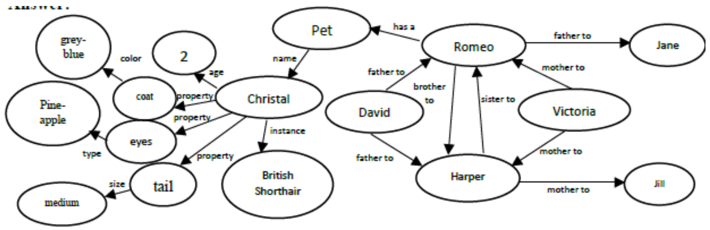
1. Semantic networks: Nodes and arcs represent concepts and relationships.
2. Conceptual graphs: Use nodes for concepts and relations without labeled arcs.
3. And/Or graphs: Represent logical conjunctions and disjunctions.
4. Frames: Structured records for representing entities and their attributes.

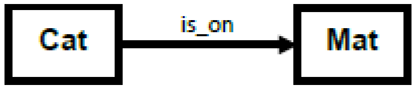
Applications : Examples include representing relationships between objects and solving problems using knowledge structures.

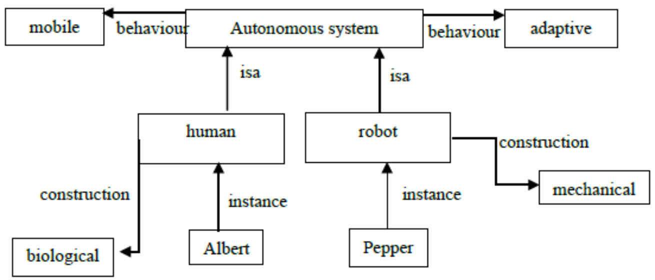
Analysis

- Knowledge representation focuses on structuring information for AI systems to reason and make decisions.
- Semantic networks and frames are intuitive for modeling relationships, while And/Or graphs handle logical decision-making.

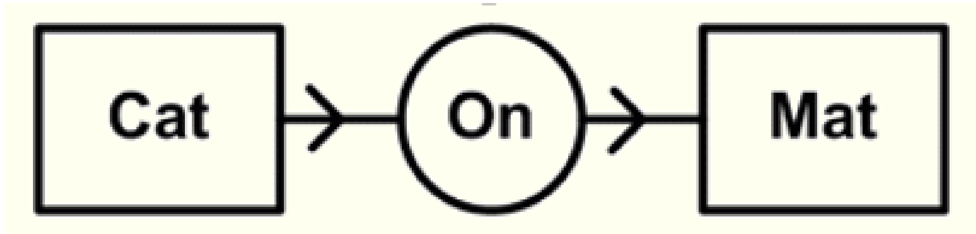
Semantic Network







Conceptual Graph



The nodes of the graph are either concepts, events, or action.
Labeled arcs to show the relationships

The nodes of the graph are either concepts or conceptual relations.
Do not use labeled arcs

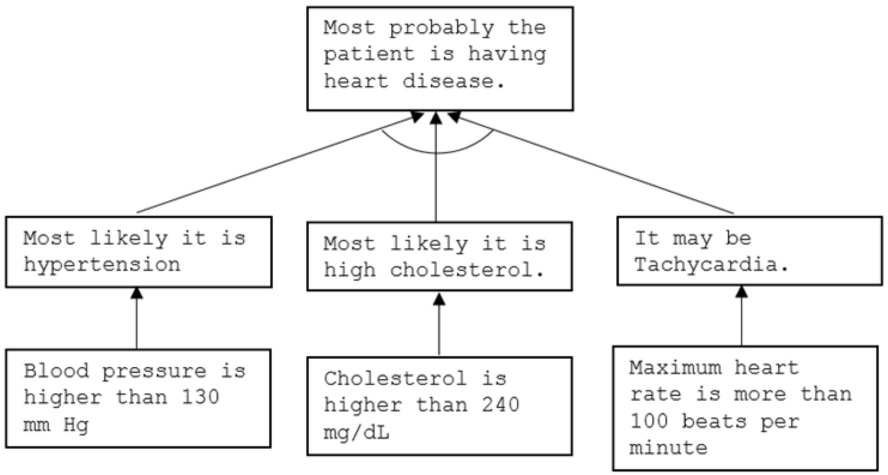
And/Or graphs

A rules based heart failure rule-based system is shown below.

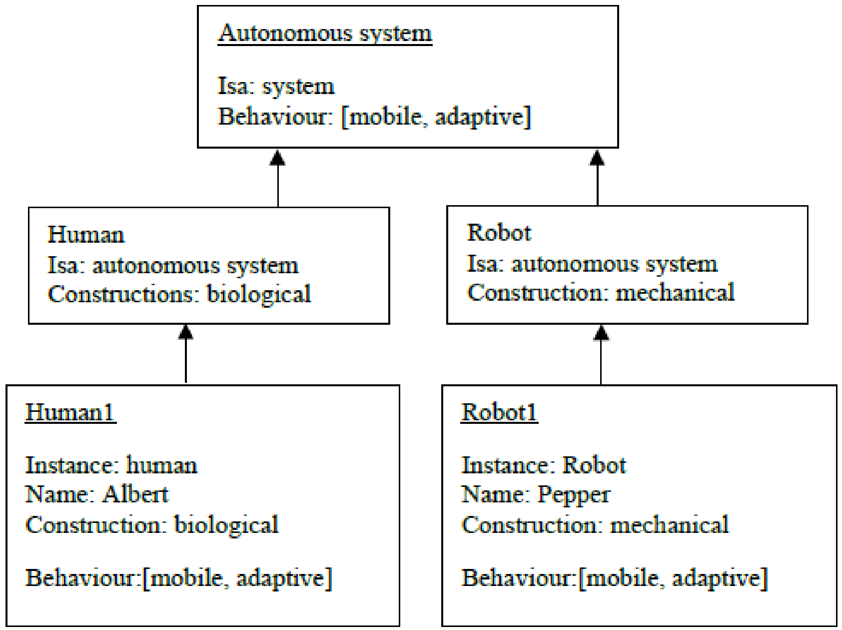
R1:	IF	Blood pressure is higher than 130 mm Hg
	THEN	Most likely it is hypertension.
R2:	IF	Cholesterol is higher than 240 mg/dL
	THEN	Most likely it is high cholesterol.
R3:	IF	Maximum heart rate is more than 100 beats per minute
	THEN	It may be Tachycardia.
R4:	IF	There is hypertension
	AND	High cholesterol
	AND	Tachycardia
	THEN	Most probably the patient is having heart disease.

Proposed **ONE (1)** technique to represent the knowledge above. Then, represent the knowledge by using the proposed technique. ()

And/OR graph (1m)



Frames



Semantic ambiguity

<table>
<tr>
<td>

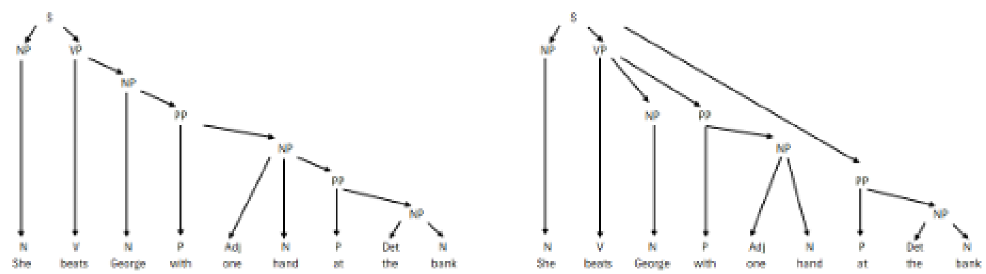
1. Considering the sentence S = "**She beats George with one hand at the bank**".(6)
   a) The sentence S consists of semantic ambiguity and syntactic ambiguity. Identify both of the ambiguities found from the sentence above.
   b) Given the grammar below, construct **ONE** (**1**) parse tree for the sentence S. (**Remark**: grammar in the parentheses **()** means it is optional.)

S -> NP VP (PP)
NP -> (DET) (ADJ) N (PP)
VP -> V NP (PP)
PP -> P NP
ADJ -> [one]
DET -> [a, the]
N -> [She, George, hand, bank]
V -> [beats]
P -> [with, at]

</td>
<td>

a)
Semantic ambiguity: bank means different things, e.g. a river bank or a financial institution
Syntactic ambiguity: It is either she uses one hand to beat George, or George has only 1 hand

b)



</td>
</tr>
</table>

C6: Natural Language Understanding

NLP Examples

Semantic search, chatbots, spelling correction, and named entity recognition.

Challenges

1. Ambiguity in syntax and semantics (e.g., "I saw her duck").
2. Stages of Language Analysis
3. Parsing (syntactic structure), semantic interpretation, and world knowledge.
4. Symbolic Analysis
5. Morphological and syntactic analysis to break down language into meaningful units.

Analysis

NLP involves understanding and processing human language, with applications ranging from chatbots to semantic search.

Challenges include handling ambiguity and context-dependent meanings.

NLP Examples

Semantic and Entity-Based Search (Search Engine)
A.L.I.C.E Chatbot
Spelling Correction and Text Prediction
Word-sense disambiguation- Bank, Goal
Named entity recognition- Canon, Apple
Information retrieval- Search engine
Summarization- QuillBot's Summarizer, Summarizer.org
Machine Translation- Microsoft Translator, Google Translate

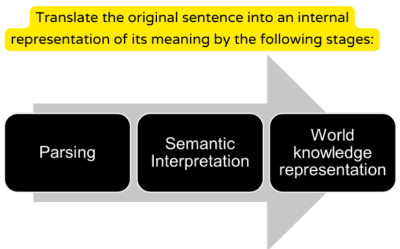
NLU Challenges

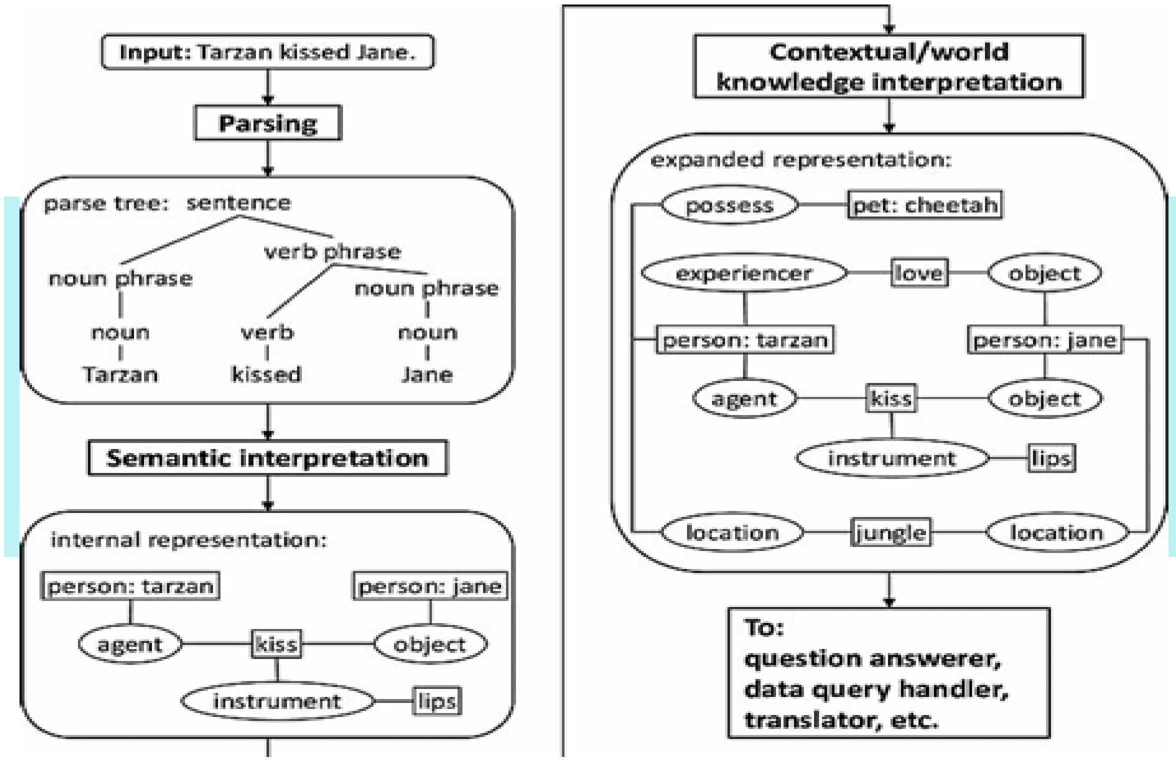
Communication with natural language, whether text or as speech acts, **depends heavily on our knowledge within the domain of discourse**.
It involves:
Transmission of words
Inferences about speaker’s goals
Knowledge
Assumptions
The context of the interaction

Flow of NLU and Symbolic Analysis

Stages of Language Analysis

Parsing > Semantic Interpretation > World Knowledge Representation





1: Parsing

Analyze the syntactic structure of sentences Identifying the major relations such as subject-verb, verb-object, noun-modifier Often represented as parse tree Employs knowledge of language syntax, morphology, and some semantics

2: Semantic Interpretation

3: World Knowledge Representation