Artificial Intelligence

Short notes for B.Tech CSE / IT · BCA · MCA · M.Tech

Uncodemy Handwritten Notes · free for students

1. Introduction

- **Artificial Intelligence (AI)** = making machines that can think, learn and act intelligently.
- The term was coined by John McCarthy in 1956 (Dartmouth conference).
- **Turing Test** (Alan Turing, 1950): if a human judge cannot tell a machine from a human in a text chat, the machine is said to be intelligent.

➤ Four approaches to AI

	Human-like	Rational
Thinking	Think like humans (cognitive modelling)	Think rationally (laws of logic)
Acting	Act like humans (Turing test)	Act rationally (rational agent)

- **Weak / Narrow AI**: does one task (chess, face unlock). **Strong / General AI**: human-level on any task (not achieved yet).
- Applications: games, NLP, robotics, expert systems, computer vision, recommendation.

2. Intelligent Agents

- **Agent** = anything that perceives its environment through **sensors** and acts through **actuators**.
- **Rational agent** = chooses the action that maximises expected performance.
- **PEAS** description: Performance measure, Environment, Actuators, Sensors.

> **EXAMPLE** Self-driving taxi. P: safe, fast, legal trip. E: roads, traffic, people. A: steering, brake, horn. S: cameras, GPS, speedometer.

➤ Types of agents

1. **Simple reflex**: acts only on the current percept (condition-action rules).

2. **Model-based reflex**: keeps an internal state of the world.
3. **Goal-based**: chooses actions that reach a goal.
4. **Utility-based**: chooses the action with the highest utility (happiness).
5. **Learning agent**: improves with experience.

➤ Environment types

Property	Options
Observability	Fully vs partially observable
Agents	Single vs multi-agent
Outcome	Deterministic vs stochastic
Episodes	Episodic vs sequential
Change	Static vs dynamic
Values	Discrete vs continuous

EXAM POINT Chess = fully observable, deterministic, sequential, static (without clock), discrete, multi-agent.

3. Problem Solving and Uninformed Search

- A problem = initial state, actions, transition model, goal test, path cost.
- **State space** = all states reachable from the initial state.
- Examples: 8-puzzle, water jug, missionaries and cannibals, 8-queens.

➤ Uninformed (blind) search

Algorithm	Data structure	Complete	Optimal	Time	Space
BFS	Queue (FIFO)	Yes	Yes (equal step cost)	O(b^d)	O(b^d)
DFS	Stack (LIFO)	No (infinite paths)	No	O(b^m)	O(bm)
Uniform cost	Priority queue by g(n)	Yes	Yes	O(b^(1+C*/e))	same
Depth-limited	Stack, limit l	No	No	O(b^l)	O(bl)
Iterative deepening	Repeated DLS	Yes	Yes (equal step cost)	O(b^d)	O(bd)
Bidirectional	Two BFS	Yes	Yes	O(b^(d/2))	O(b^(d/2))

- b = branching factor, d = depth of shallowest goal, m = maximum depth.

TIP Iterative deepening gives BFS's completeness with DFS's small memory. It is the preferred blind search for big spaces.

4. Informed (Heuristic) Search

- **Heuristic h(n)** = estimated cost from node n to the goal.
- **Greedy best-first**: expands the node with the smallest h(n). Fast but not optimal.
- **A-star (A*) search**: expands the node with the smallest f(n).

FORMULA A^* uses f(n) = g(n) + h(n), where g(n) = cost from start to n, h(n) = estimated cost from n to goal.

- **Admissible** heuristic: never overestimates, h(n) <= true cost. Then A^* (tree search) is optimal.
- **Consistent** heuristic: h(n) <= cost(n, n') + h(n'). Consistent implies admissible.
- 8-puzzle heuristics: h1 = number of misplaced tiles, h2 = sum of Manhattan distances. Both admissible, h2 dominates h1.

Local search

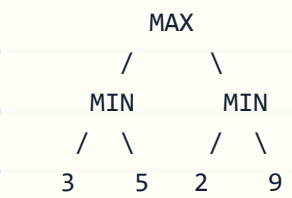
- **Hill climbing**: always move to a better neighbour. Problems: local maxima, plateau, ridge.
- Fixes: random restart, stochastic hill climbing.
- **Simulated annealing**: sometimes accepts a worse move with probability $e^{(-\Delta E/T)}$; T falls over time.
- **Genetic algorithm**: population, fitness, selection, crossover, mutation.

5. Adversarial Search (Games)

- **Minimax**: MAX picks the highest value, MIN picks the lowest, values backed up from leaves.
- Time $O(b^m)$, space $O(bm)$.

Alpha-beta pruning

- α = best value MAX can guarantee so far, β = best value MIN can guarantee so far.
- Prune a branch when $\alpha >= \beta$.
- Gives the same answer as minimax. With perfect ordering, time becomes $O(b^{(m/2)})$.



EXAMPLE Left MIN = min(3,5) = 3. So $\alpha = 3$ at root. Right MIN sees 2 first; $2 < \alpha$ so its value is at most 2 and leaf 9 is pruned. Root = max(3, 2) = 3.

6. Knowledge Representation and Logic

Propositional logic

- Connectives: ¬ (not), ∧ (and), ∨ (or), → (implies), ↔ (iff).
- $p \to q$ is false only when p is true and q is false. $p \to q \equiv \neg p \vee q$.
- **Tautology**: always true. **Contradiction**: always false.
- Inference rules: Modus Ponens ($p, p \to q \vdash q$), Modus Tollens ($\neg q, p \to q \vdash \neg p$).

First-order predicate logic (FOPL)

- Adds objects, predicates, functions and quantifiers: ∀ (for all), ∃ (there exists).
- "All students are smart": ∀x Student(x) → Smart(x).
- "Some student is smart": ∃x Student(x) ∧ Smart(x).

EXAM POINT With ∀ use →. With ∃ use ∧. Mixing them is the most common mistake.

Resolution

1. Convert all sentences to **CNF** (clause form).
2. Add the negation of the goal.
3. Keep resolving pairs of clauses.
4. If the empty clause appears, the goal is proved (proof by contradiction).

- FOPL needs **unification** (find substitutions that make terms equal) and **Skolemization** (remove ∃).

Other representations

- **Semantic network**: graph of concepts with IS-A and HAS-A links.
- **Frames**: slot-filler structures for objects.
- **Rules**: IF-THEN. **Forward chaining** = data-driven (facts to goal). **Backward chaining** = goal-driven (goal back to facts).

7. Reasoning under Uncertainty

- Real knowledge is often uncertain, so we use probability.

FORMULA Bayes' theorem: $P(A|B) = P(B|A) \times P(A) / P(B)$.

EXAMPLE 1% of people have a disease. A test catches 90% of sick people and gives 10% false positives. $P(+) = 0.9 \times 0.01 + 0.1 \times 0.99 = 0.009 + 0.099 = 0.108$. $P(\text{sick}|+) = 0.009 / 0.108 = 0.083$ (about 8.3%).

- **Bayesian network**: directed acyclic graph; each node has a conditional probability table given its parents.
- **Fuzzy logic**: truth values between 0 and 1 (not just true / false). Steps: fuzzification, rule evaluation, defuzzification.

- **Certainty factors**: used in early expert systems like MYCIN.

8. Planning, Expert Systems and Learning

➤ Planning

- **STRIPS** representation: each action has preconditions, add list and delete list.
- Example domain: blocks world.

➤ Expert system

- A program that acts like a human expert in a narrow field.
- Parts: knowledge base, inference engine, user interface, explanation facility, knowledge acquisition.
- Examples: MYCIN (medical diagnosis), DENDRAL (chemistry).

➤ Learning (overview)

Type	Data	Example
Supervised	Labelled	Spam detection
Unsupervised	Unlabelled	Customer clustering
Reinforcement	Rewards from actions	Game playing robot

- **NLP** steps: tokenisation, parsing, semantic analysis, pragmatics.

9. Most Asked Questions (practice)

1. Define AI. Explain the Turing test and four approaches to AI.
2. What is an agent? Explain types of agents and PEAS with an example.
3. Compare BFS, DFS and iterative deepening.
4. Explain A* search. What is an admissible heuristic?
5. Explain hill climbing and its problems.
6. Solve a minimax tree with alpha-beta pruning.
7. Convert English sentences to FOPL and prove a goal by resolution.
8. Forward vs backward chaining.
9. Explain Bayes' theorem and Bayesian networks with an example.

10. Architecture of an expert system.

> **TIP** Long answers usually come from search algorithms, alpha-beta, resolution and agents. Practise one numerical tree and one resolution proof.