12	Simple Turing machine as a model of computation and its configuration, computing with Turing machine and its working.
13	Building simple Turing machines, combining Turing machines, Church Turing Thesis.
14-15	Universal Turing machine, semi-decidability and decidability, recursively-enumerable and recursive languages, halting problem.

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Regular expressions and languages, finite automata, context free grammar and languages, pushdown automata, Turing machine.

Artificial Intelligence (BHCS13) Discipline Specific Core Course - (DSC)

Credit: 06

Course Objective

This course introduces the basic concepts and techniques of Artificial Intelligence (AI). The course aims to introduce intelligent agents and reasoning, heuristic search techniques, game playing, knowledge representation, reasoning with uncertain knowledge.

Course Learning Outcomes

On successful completion of this course, students will be able to:

1. Identify problems that are amenable to solution by specific AI methods
2. Represent knowledge in Prolog and write code for drawing inferences.
3. Identify appropriate AI technique for the problem at hand
4. Compare strengths and weaknesses of different artificial Intelligence techniques.
5. Sensitive towards development of responsible Artificial Intelligence

Detailed Syllabus

Unit 1

Introduction: Introduction to artificial intelligence, background and applications, Turing test, rational agents, intelligent agents, structure, behaviour and environment of intelligent agents.

Unit 2

Knowledge Representation: Propositional logic, first order predicate logic, resolution principle, unification, semantic nets, conceptual dependencies, frames, scripts, production rules, conceptual graphs.

Unit 3

Reasoning with Uncertain Knowledge: Uncertainty, non-monotonic reasoning, truth maintenance systems, default reasoning and closed world assumption, Introduction to probabilistic reasoning, Bayesian probabilistic inference, introduction to fuzzy sets and fuzzy logic, reasoning using fuzzy logic.

Unit 4

Problem Solving and Searching Techniques: Problem characteristics, production systems, control strategies, breadth first search, depth first search, hill climbing and its variations, heuristics search techniques: best first search, A* algorithm, constraint satisfaction problem, means-end analysis.

Unit 5

Game Playing: introduction to game playing, min-max and alpha-beta pruning algorithms.

Prolog Programming: Introduction to Programming in Logic (PROLOG), Lists, Operators, basic Input and Output.

Unit 6

Understanding Natural Languages: Overview of linguistics, Chomsky hierarchy of grammars, parsing techniques.

Unit 7

Ethics in AI, Fairness in AI, Legal perspective

Practical

1. Write a prolog program to calculate the sum of two numbers.
2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.
3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.

4. Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series.
5. Write a Prolog program to implement GCD of two numbers.
6. Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.
7. Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result.
8. Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.
9. Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.
10. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.
11. Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.
12. Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement nth_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L.
15. Write a Prolog program to implement maxlist(L, M) so that M is the maximum number in the list.
16. Write a prolog program to implement insert_nth (I, N, L, R) that inserts an item I into Nth position of list L to generate a list R.
17. Write a Prolog program to implement delete_nth (N, L, R) that removes the element on Nth position from a list L to generate a list R.
18. Write a program in PROLOG to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.

References

1. Rich, E. & Knight,K. (2012). *Artificial Intelligence*. 3rd edition. Tata McGraw Hill.

2. Russell, S.J. & Norvig, P. (2015) *Artificial Intelligence - A Modern Approach*. 3rd edition. Pearson Education

Additional Resources:

1. Bratko, I. (2011). *Prolog Programming for Artificial Intelligence*. 4th edition. Pearson Education

2. Clocksin, W.F. & Mellish (2003), *Programming in PROLOG*. 5th edition. Springer

3. Kaushik, S. (2011). *Artificial Intelligence*. Cengage Learning India.

4. Patterson, D.W. (2015). *Introduction to Artificial Intelligence and Expert Systems.* 1st edition. Pearson Education.

Web Resources

1. https://cyber.harvard.edu/topics/ethics-and-governance-ai
2. https://royalsocietypublishing.org/doi/full/10.1098/rsta.2018.0085
3. https://arxiv.org/abs/1812.02953

Course Teaching Learning Process

- Use of ICT tools in conjunction with traditional class room teaching methods
- Interactive sessions
- Class discussions

Tentative weekly teaching plan is as follows:

Week	Content
1	Introduction to artificial intelligence, background and applications, Turing test and rational agent approaches to AI, introduction to intelligent agents.
2	Structure, behavior and environment of intelligent agents, problem characteristics, production systems, control strategies.
3	Introduction to programming in logic (PROLOG).
4	Programming in logic (PROLOG), breadth first search, depth first search introduction of heuristic search techniques.
5	Propositional logic, first order predicate logic.
6	Unification, clausal form, resolution principle.
7	Semantic nets, conceptual graphs, conceptual dependencies.
8	Frames, scripts, Uncertainty: non-monotonic reasoning, truth maintenance systems, default reasoning and closed world assumption.

9	Bayesian probabilistic inference, Bayesian networks, Dempster-Shafer theory, Introduction to fuzzy sets and fuzzy logic.
10	Basic reasoning using fuzzy concepts, production rules, Chomsky hierarchy of grammars, context-free grammars.
11	Hill climbing and its variations, best first search.
12	A* algorithm, constraint satisfaction problem, means-end analysis.
13	Introduction to game playing, min-max procedure, alpha-beta pruning.
14-15	Overview of linguistics, Chomsky hierarchy of grammars, parsing techniques..

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Artificial Intelligence, Problem Solving, Knowledge Representation, Reasoning, Uncertainty, Natural Language Processing

Computer Graphics (BHCS14) Discipline Specific Core Course - (DSC)

Credit: 06

Course Objective

This course introduces fundamental concepts of Computer Graphics with focus on modelling, rendering and interaction aspects of computer graphics. The course emphasizes the basic principles needed to design, use and understand computer graphics system.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Describe Standard raster and vector scan devices as well as Graphical Input and output devices
2. Implement algorithms for drawing basic primitives such as linecircle and ellipse.
3. Implement algorithms for line clipping and polygon clipping and filling.