December 20, 2017

Core Paper XIII: Artificial Intelligence

Sr. No.	Topic	Chapter	Reference	# of Lecture
1.	**Introduction:** Introduction to Artificial Intelligence, Background, Turing Test and Rational Agent approaches to AI, Applications of AI. Introduction to Intelligent Agents, their structure, behavior and environment.	1 (1.1, 1.4) 2 (Complete)	2 2	6
2.	**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. Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.	2.1-2.3 3.1 - 3.3.2, 3.5, 3.6 12.1 - 12.3	3 3 3	12
3.	**Knowledge Representation:** Introduction to First Order Predicate Logic, Resolution Principle, Unification. Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG).	4 (4.1 - 4.9) 7 (Complete) 1, 2, 3, 4	1 1 4	14
4.	**Dealing with Uncertainty and Inconsistencies:** Truth Maintenance System, Default Reasoning, Probabilistic Reasoning: Bayesian Probabilistic Inference, Possible World Representations.	5 (5.1 - 5.3) 6 (6.1 - 6.3)	1	8
5.	**Understanding Natural Languages:** Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.	12 (12.1 - 12.4)	1	10

References:

1. DAN. W. Patterson, Introduction to A.I and Expert Systems – PHI, 2004.
2. Russell & Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Ptentice Hall, 2nd edition, 2005.
3. Rich & Knight, Artificial Intelligence- Tata McGraw Hill, 3rd Edition, 2009.
4. William F. Clocksin, Christopher S. Mellish, Programming in Prolog, Springer-Verlag Berlin, 5th edition, 2003.
5. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3rd Edition, 2000.

LIST OF PRACTICALS CORE PAPER XIII: ARTIFICIAL INTELLIGENCE

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 program in PROLOG to implement towerofhanoi (N) where N represents the number of discs
9. Consider a cyclic directed graph [edge (p, q), edge (q, r), edge (q, r), edge (q, s), edge (s,t)] where edge (A,B) is a predicate indicating directed edge in a graph from a node A to a node B. Write a program to check whether there is a route from one node to another node.
10. Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.
11. 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.
12. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.
13. Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.
14. Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L.
15. 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
16. 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.

17. Write a program in PROLOG to implement remove_dup (L, R) where L denotes the list with some duplicates and the list R denotes the list with duplicates removed.

18. Write a Prolog program to implement maxlist(L, M) so that M is the maximum number in the list

19. 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.

20. Write a Program in PROLOG to implement sublist(S, L) that checks whether the list S is the sublist of list L or not. (Check for sequence or the part in the same order).

21. 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.

22. Write a program in PROLOG to implement delete_all (X, L, R) where X denotes the element whose all occurrences has to be deleted from list L to obtain list R.

23. 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.

24. Write a PROLOG program that will take grammar rules in the following format:

    NT □ (NT | T)*

    Where NT is any nonterminal, T is any terminal and Kleene star (*) signifies any number of repetitions, and generate the corresponding top-down parser, that is:

    sentence □ noun-phrase, verb-phrase

    determiner □ [the]

    will generate the following:

    sentence (I, O) :- noun-phrase(I,R), verb-phrase

    (R,O). determiner ([the|X], X) :- !.

25. Write a prolog program that implements Semantic Networks (ATN/RTN).