CSE112 Artificial Intelligence Assignment

Assessment

This assignment contributes 10% to the overall assessment of CSE112.

Submission Deadline

Monday, 13 May 2019 by 23:59

1. Learning Outcomes for this Assessment

This assessment aims at assessing your understanding of the topics and your knowledge on Prolog programming and methods of artificial intelligence. After successfully completed the assignment, you will grab a better understanding of facts, rules, backtracking, depth first search, and querying in Prolog. You will also appreciate the way to formulate problem and translate the formulation into Prolog program.

2. The Task

Write a Prolog program to address the following problem:

Water jug is a classical formal problem and is generally stated as below.

Suppose that you have two jugs with no measure markings: 4-gallon jug (Jug A) and 3-gallon jug (Jug B). We have an infinite supply of water to fill the jugs, and can also pour away water to empty the jug. The task is to measure exactly 2 gallons of water in Jug A.

(1) Write a Prolog program to solve the problem using the depth first search strategy.

(2) Show the solution generated by your program for all the FOUR possible variants of initial states. For example, (0, 0) means both jugs are empty, (4, 0) indicates jug A is full while jug B is empty.

(3) Avoid infinite loop by keeping track of repeated states.

(4) Your code should be easy to understand with proper indentation and comments.

(5) Your code is expected to be self-documenting (e.g., meaningful variable names).

Hints:

The state is represented by (x, y), where x represents the number of gallons of water in Jug A and y represents the number of gallons of water in Jug B.

Initial state: (0, 0)

Example actions:

Action	Condition	Result
Fill up Jug A	If Jug A is not filled up	(4, y)
Empty Jug B	If Jug B has water	(x, 0)

Fill up Jug A with water from Jug B	If Jug B has enough water to fill up Jug A	(4, y – (4 – x))
Fill Jug B with all water in Jug A	If Jug A has not enough water to overflow Jug B	(0, y + x)

Goal state: (2, y)

Path cost: The number of steps to reach the goal, the amount of water pours away, etc.

3. Submission Requirements

(1) Submit a soft copy of your Prolog source code file through ICE.
(2) Submit a soft copy of a test case (one of the FOUR possible variants of initial states) document with expected results through ICE.
(3) Name your source code as WATER-JUG_<student ID>_assignment.pl. For example, student with ID 1234567 should name the file as WATERJUG_1234567_assignment.pl.
(4) Name your test case as WATER-JUG_<student ID>_test.doc (or .docx). For example, student with ID 1234567 should name the file as WATER-JUG_1234567_test.doc.
(5) Failure in submitting any of the required files (source code, test case) could result in 0 points.

Late submission will receive penalty in the marking in accordance with the University Code of Practice on Assessment. For each working day after the deadline, 5 marks (out of 100) will be deducted for up to 5 working days. However, the mark will not be reduced below the pass mark for the assessment. Work assessed below the pass mark will not be penalised for late submission of up to five days. Work received more than 5 working days after the deadline will receive a mark of 0.

4. Plagiarism and Collusion

This assignment is individual work. Plagiarism (e.g. copying materials from other sources without proper acknowledgement) is a serious academic offence. Plagiarism will not be tolerated and will be dealt with in accordance with the University Code of Practice on Assessment.

5. Guide to Marking

Criteria	Marks
(1) The program successfully find a solution for one of the FOUR possible variants of initial states.	20
(2) The program successfully find a solution for any of the other three possible variants of initial states specified by the marker.	25
(3) The program successfully avoids infinite loop.	25
(4) Source code has high readability, constructive comments, and self-document.	10
(5) Sufficient test cases and expected results are documented accurately and concisely.	10
(6) Reliability and stability of the program	10