COS4851 Assignment 4 (2019) [50]

ASSIGNMENT 04

DUE DATE:	14 October 2019
SUBMISSION PROCEDURE:	Written
WEIGHT:	30
UNIQUE NUMBER:	853546
STUDY MATERIAL:	*Bratko,* Chapters 13, 14 and 17 (chapter 13: ignore sections 13.3 and 13.4 chapter 14: ignore section 14.4 c chapter 17: ns 17.4 – 17.6)

The purpose of this assignment is to introduce more advanced search algorithms and to highlight the complexity in terms of time and space of these. You will also learn how to find solutions to AND/OR graphs and learn different ways in which a reasoner can approach finding solutions to planning problems

Test Results: Please note that you need to submit screen shots (using *fn+prt sc* for Windows in most instances) of results for questions where a program/procedure or query is required. This will assist us to see whether you obtained the correct results and if not, try to point out where you went wrong.

Your programs should also be *robust*. This means that it should check whether all the input arguments for a specific procedure are legal. For example, if you know you are working with integers, an input of the constant that is not an integer is not acceptable.

Important note: It is not advisable that you search for Prolog solutions to the questions in this assignment on the internet. You may find a solution to a specific problem but that will not assist you in acquiring the necessary skills for mastering this programming paradigm.

Question 1 [10]

Construct a table listing both the advantages and disadvantages in terms of time and space complexity of each of the search strategies discussed in chapters 11, 12 and 13 (excluding those discussed in the sections that are excluded for examination purposes).

Question 2 [10]

Define in Prolog an AND/OR space for the tower of Hanoi problem and use this definition with the search procedures discussed in Section 14.3 of *Bratko* to find a solution to the problem.

Question 3 [20]

Consider the following blocks-world problem: There are four blocks, named `a`, `b`, `c` and `d` and four places, numbered `1`, `2`, `3` and `4`. The initial state is given by the following list:

```
[clear(a),clear(b),clear(c),clear(d),on(a,1),on(b,2),on(c,3),on(d,4)]
```

Use the following assumptions in your answers to the questions given below:

(i) The add-list for action `move(X,From,To)` is `[on(X,To), clear(From)].`

(ii) The delete-list for action `move(X,From,To)` is `[on(X,From), clear(To)].`

(iii) The list of preconditions for action `move(X,From,To)` is `[clear(X), clear(To), on(X,From)].`

(vi) Always select the leftmost goal in the goal list.

Note that we follow the convention used by *Bratko*: After an action has been completed, the elements in the state space (if present) have the following sequence:

```
[clear(a),…,clear(c),clear(1),…,clear(3),on(a,?),…,on(c,?)]
```

3.1 Give the sequence of moves that a *simple means-ends planner* will generate to achieve the goal `on(d,1)` from the given initial state. Give the preconditions of each proposed action as well as the subgoal that each action in the plan achieves. (10)

3.2 Give the sequence of moves that a *means-ends planner with goal protection* will generate to achieve the goal `on(d,1)` from the given initial state. Give the preconditions of each proposed action and the subgoal that each action in the plan achieves, as well as the protected goals at each stage.

Note: Refer to the notes on goal protection that are available under *Additional Resources* on myUnisa. (10)

Question 4 [10]

Give the sequence of moves that a means-ends planner with *goal regression* will generate to achieve the goal `on(a,1)` from the following initial state:

```
[clear(a),clear(b),clear(c),clear(d),on(a,1),on(b,2),on(c,3),on(d,4)]
```

Also give the subgoal that each action in the plan achieves, as well as the regressed goals at each stage.

Note: Refer to the notes on goal regression that are available under *Additional Resources* on myUnisa.