

CSC520 Fall 2019 Assignment 4

Due November 5th at 11:59pm

This assignment includes both conceptual and code questions. It must be completed individually. You may not collaborate with other students, share code, or exchange partial answers. Questions involving answers or code must be emailed to the instructor or TAs directly or discussed during office hours. Your answers to the conceptual questions must be uploaded to Moodle as a pdf file titled **Assign4_UnityID.pdf**. Your sourcecode must be submitted as a self-contained zip called **Assign4_UnityID.zip**. All code must be clear, readable, and well-commented.

Note: The code will be tested on the NCSU VCL system("CSC520_VCL"). You are advised to test your code there before submission.

Question 1 (20 pts)

Consider the following facts:

All books are either fiction or nonfiction. Dictionaries are nonfiction, so are cookbooks. The OED is a dictionary as are Webster's and Johnston's dictionaries. "Michael Strogoff" is fiction. "The Guide" is a Novel. And all novels are fiction. "Flour Water Salt Yeast" is a cookbook.

- (5 pts) Draw a frame representation of these facts with appropriate labels for the nodes and edges. Include this in your report.
- (5 pts) Using a consistent top level syntax as discussed in class along with other necessary predicates write a Prolog representation of the taxonomy along with one or more rules to infer that each book described is a novel, a dictionary, fiction, or nonfiction etc. as appropriate and that they are all books. Submit this as a file called **q1b.pl**
- (5 pts) Now expand to a frame representation that also incorporates the following information:

All books have editors. All fiction has authors. Webster's and Johnston's dictionaries have authors (Noah and Samuel respectively). The OED does not. The OED has 20 volumes. Most other books have 1. The author of "Michael Strogoff" is Jules while the author of "The Guide" is RK. Also show that Webster's and Johnston's dictionaries agree with each other. While Johnston's agrees with the OED but not vice-versa.
- (5 pts) Enhance your code to include the above knowledge and to allow for the inference that the number of volumes of "The Guide" is 1 and that Johnston's dictionary agrees with everyone but that the OED does not agree with anyone. Submit this as a file called **q1d.pl**

Question 2 (20 pts)

For this question consider the following problem description:

You need to install the new engine in your car, as part of that you need a fresh set of spark plugs and a fresh belt, neither of which you have. However your garage is clean. To to change the plugs you need to have no plugs and a clean garage to work in. To change the belt you need no belt and a clean garage. Changing the belt or changing the plugs will get you a new belt or new plugs but doing either one will mess up your garage. When the garage is dirty you can always clean it. At the end of the day you want an engine and a clean garage.

- (5 pts) Provide a PDDL representation of the above problem specification.
- (15 pts) Apply the Graphplan algorithm to solve this problem showing the complete steps, mutex links, and the final plan. All levels, actions, variables and mutex links must be clearly labeled and their types shown. Add your graph and the plan to your final report.

Question 3 (60 pts)

For this question you must implement and evaluate the critical path scheduling method in Prolog for arbitrary scheduling graphs. Your code must be designed to work with a database of tasks, prerequisites, and durations of the following form:

```
duration(a, 10).  
prerequisite(b, a).
```

Where the first states the duration of task a is 10 units and the second that task a is a necessary prerequisite for task b. You will be given a pair of graph files which you can use to test your code. You are encouraged to develop your own files for testing as your code will be tested on novel databases.

Your final code must implement the following predicates for testing *without using the if-then construct*:

```
criticalPath(<task>, <path>).  
earlyFinish(<task>, <time>).  
lateStart(<task>, <time>).  
maxSlack(<task>, <time>).
```

The first of these should map to true if and only if **<path>** is an ordered list of the items in the critical path to the designated task. The second must map true if and only if **<time>** is the earliest time that **<task>** can complete in the schedule. The third should return true if and only if **<time>** is the latest that a **<task>** can begin without delaying anything on the critical path. And the fourth should return true if and only if **<task>** is the one known task with the most slack in the database and **<time>** is that amount of slack. As always your code must be clear, readable, and well commented, and your own. Your final report should include diagrams of the test graphs with values for the early start, late start, and slack and the critical path indicated.