

CSC520 Spring 2018 - Assignment 4

Due by **April 4th 2019**.

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_FirstLastName.pdf”. Your sourcecode must be submitted as a self-contained zip titled “Assign4_FirstLastName.zip”. All code must be clear, readable, and well-commented. The code will be tested on the NCSU VCL system with the image named “CSC520 VCL”. “CSC520 VCL” has Java, Prolog, and Python installed. You are advised to test your code there before submission.

Question 1: Graphplan Planning (60 points)

In the first scenario we have a cooking problem similar to the one covered in class. Imagine you want to cook dinner with only one table, which is clean at first. Consider the following propositional planning problem:

```
Init(Hungry  $\wedge$  CleanTable),
Goal( $\neg$ Hungry  $\wedge$  CleanTable  $\wedge$   $\neg$ Dinner),
action( Cook,
    PRECOND: CleanTable
    EFFECT: Dinner)
action( Eat,
    PRECOND: Dinner  $\wedge$  Hungry
    EFFECT:  $\neg$ Hugry  $\wedge$   $\neg$ CleanTable  $\wedge$   $\neg$ Dinner)
action( Clean,
    PRECOND:  $\neg$ CleanTable
    EFFECT: CleanTable)
```

In the second scenario, we have two locations: A and B, and only one car for moving packets from A to B. The packet can be loaded in the car only if the car is empty.

```

Init( $\neg PacketAt(A) \wedge CarAt(B) \wedge \neg CarAt(A)$ ),
Goal( $PacketAt(B) \wedge CarAt(A) \wedge \neg CarAt(B) \wedge \neg PacketInCar$ ),
action( Load,
    PRECOND:  $PacketAt(A) \wedge \neg PacketInCar \wedge CarAt(A) \wedge \neg CarAt(B)$ 
    EFFECT:  $PacketInCar \wedge \neg PacketAt(A) \wedge CarAt(A) \wedge \neg CarAt(B)$ )
action( Drive(A,B),
    PRECOND:  $CarAt(A) \wedge \neg CarAt(B)$ 
    EFFECT:  $CarAt(B) \wedge \neg CarAt(A)$ )
action( Drive(B,A),
    PRECOND:  $CarAt(B) \wedge \neg CarAt(A)$ 
    EFFECT:  $CarAt(A) \wedge \neg CarAt(B)$ )
action( Produce,
    PRECOND:  $\neg PacketAt(A)$ 
    EFFECT:  $PacketAt(A)$ )
action( Discharge,
    PRECOND:  $CarAt(B) \wedge PacketInCar \wedge \neg CarAt(A)$ 
    EFFECT:  $PacketAt(B) \wedge \neg PacketInCar \wedge CarAt(B) \wedge \neg CarAt(A)$ )

```

In an approved language of your choice (Java, C, C++, Python), implement the graphplan algorithm. Your program should get the initial state, goal and each action consisting of the pre-condition and effect as the format in attached planning problem file. Your program should:

- (a) (5 pts) Parse the specified file and print out all the variables and actions.
- (b) (10 pts) Generate each layer of the planning graph until a solution can be found or no solution will be found and print out all actions or state variables in each layer.
- (c) (15 pts) Print out all of the mutex links that appear in each layer.
- (d) (20 pts) Return a viable solution plan as a partially-ordered sequence of actions if one exists or indicate when no solution can be found.
- (e) (10 pts) be clear readable and well-documented and run without error.

Please note that your program must be able to work for any planning graph problem of the same format.

Question 2: Scheduling (24 points)

Consider the following activities of a project and their estimated duration in days. Each activity has some prerequisite(s) which need to be completed before that activity can begin.

Activity	Prerequisite	Estimated Duration (days)
Start	-	0
C	Start	6
B	Start	4
P	Start	3
A	C, B, P	7
U	P	4
T	A	2
R	A	3
N	U	6
End	T, R, N	0

- (a) (4 points) Represent these ordering constraints as a directed graph relating the activities (as shown in the textbook) .
- (b) (8 points) Apply the Critical Path Method to this graph to calculate the earliest possible start time (ES) and latest possible start time (LS) of each activity. Also calculate the slack/float of each activity and annotate each node in the graph with [LS, ES] and the duration of the activity as shown in the slides. Show your work.

- (c) (4 points) Identify the Critical path and its' duration. What is the total duration of the project assuming no delay over the estimated duration of the activities?
- (d) (8 points) Assuming that the activities start at their Earliest start time (ES) but activity P takes two days longer to finish than estimated. How would it impact the duration of the project? What if instead of activity P, activity R takes a day longer than the estimated time?

Question 3: FOPL (16 points)

Consider the following sentences for NC State Wolfpack women's basketball team:

1. Jessica, Sally, Hannah, and Sarah are players.
 2. Christina is the captain.
 3. The captain is a player.
 4. All players consider the captain as a friend or hate the captain.
 5. A player will criticize another player if and only if she does not consider them to be her friend.
 6. If one player is criticized then they will return the favor.
 7. Jessica has criticized Christina.
 8. Christina has criticized Hannah and Sarah.
- (a) (8 points) Convert those sentences into first-order predicate logic, then convert to CNF. Use the following lexicon:
 - $\text{player}(X)$ – X is a player.
 - $\text{captain}(Y)$ – Y is the captain.
 - $\text{friend}(X, Y)$ – X considers Y as a friend.
 - $\text{hates}(X, Y)$ – X hates Y.
 - $\text{criticizes}(X, Y)$ – X criticizes Y.
 - (b) (8 points) Use FOPL resolution to derive the result that “Jessica hates Christina.”. Number your clauses, and indicate explicitly step-by-step what resolves together, and under what substitution.

Question 4: FOPL (Bonus: 10 points)

Implement the facts and rules from question 3 in Prolog as a file called "q44.pl" using the above lexicon. Your system shall allow the inference to answer who hates whom?