Qn I. Consider the planning problem from the first question of the first homework (reproduced below for your convenience)

\[
\begin{array}{|c|c|c|}
\hline
\text{operator O1} & \text{operator O2} & \text{operator O3} \\
\text{prec: } P & \text{prec: } Q & \text{prec: } P \\
\text{Eff: } R, \sim S & \text{Eff: } S & \text{Eff: } M \\
\hline
\text{operator O4} & \text{operator O5} \\
\text{prec: } R, S & \text{prec: } R, S \\
\text{Eff: } P, Q, R, S & \text{Eff: } P \\
\hline
\end{array}
\]

The \textit{initial state} is \{P, Q\} and the desired \textit{goals} are \{P, Q, R, S\}

I.A Draw the "relaxed planning graph" for this problem (relaxed planning graph ignores negative interactions--ie, no mutexes).

I.A.1 Answer: Planning Graph Without Mutexes.

Mark a relaxed plan that supports the top level goals in this relaxed planning graph.

I.A.2 Answer: Relaxed plan is: P = \{O1, O2\}

I.A.3. What is the heuristic value of the goal set \{P, Q, R, S\} in terms of:
1. Sum heuristic  
2. Level heuristic  
3. relaxed plan heuristic

\[
\begin{array}{|c|c|c|c|}
\hline
\text{STATE} & \text{SUM} & \text{SET LEVEL} & \text{RELAXED PLAN} \\
\hline
P, Q, R, S & 2 & 1 & 2 \\
\hline
\end{array}
\]
I.B. Now draw the standard mutex graph (as described in the text and used by planning graph—don’t need to use serial graph).

With respect to this standard graph, what is the heuristic cost of goal set\(\{P, Q, R, S\}\) using SUM and Level heuristics? What is the value of the adjusted sum heuristic (recall that it is equal to relaxed plan length + -ve interaction penalty).

<table>
<thead>
<tr>
<th>STATE</th>
<th>SUM</th>
<th>SET LEVEL</th>
<th>ADJSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P, Q, R, S)</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

\[\text{ADjsum} = 2 \text{ (relaxplan) } + 1 \text{ (2 non-mutex level - 1 first level in the graph) } = 3\]

Qn II

Consider the following problem. There are two actions: A1 and A2

A1: prec: \(p\) eff: \(q\)

A2 prec: \(r\) eff: \(\neg q, w\)

We start with init state where \(p\) and \(r\) are true.

**and our goals are \(q\) and \(w\).
II.a. Show how graphplan solves this problem—assuming that only static interference relations are marked. No mutex propagation is done. Show all the steps in the graph construction, search and memo finding. This is a really small problem.

Since only static interference is considered, we stop at level 1 of the graph construction phase, and start searching for a solution. Given that Q and W are not mutexes. So, we can support Q with A1, and W with A2. At this time we stop since A1 and A2 are mutex with each other. Given that there are no more choices for the subgoals, we write a memo at level 1 \{Q,W: 1\} giving the explanation for the failure.

At level 2, we can search again for a solution. This time, we can choose to support Q with A1, and W with its persistent action. We have to satisfy then the preconditions of such actions, subgoals P and W. P can only be supported by its persistent action, and W by A2. This time we have regressed to the initial state, and we have found a solution to our problem: A2-A1.

II.b. Now do this problem assuming that mutex propagation using the normal rules of Graphplan is done: With normal mutex propagation, we have to build our planning graph up to level 2, without search because even tough our goals are present at level 1, they are also mutex to each other.

The search is conducted in a similar way to that one of II.a, finding the same solution: A2-A1.
II.c. Comment on the relation between memoization and mutex propagation as evidenced by parts a and b.

C). Memos can be considered as finding mutexes during search, some of them could be higher order ones (e.g. more than 2 relations). In order to find a Memo, we need to fail first, and find the explanation for such failure. So, memos are built on a demand driven basis while mutexes are computed globally during the propagation of the planning graph. If mutex propagation gets relaxed as in II.a, more memos will be found since we may fail more times (we are not seeing the mutexes up front), such memos will be then used as kind of mutexes to prune inconsistent states later during search.

Qn III

For this problem, you will use the planning graph in the last level II.b above.

III.a. Convert the planning graph into a CSP encoding (the problem is small enough that you can write the entire encoding down). Show a solution for this CSP encoding, and show how it corresponds to a plan.

CSP Encoding

Variables and Domains:


P-1: {#, NooP-P1}, Q-1:{#,A1-1}, R-1:{#,Noop-R1}, ~Q-1:{#,A2-1}, W-1:{#,A2-1}

P-0:{#,T}, R-0:{#,T}

Goals: Q-2!=#, W-2!=#

Activation Constraints:

P-2=NooP-P2 => P-1!=#, Q-2=NooP-Q2 => Q-1!=#, Q-2=A1-2 => P-1!=#  
R-2=NooP-R2 => R-1!=#, ~Q-2=NooP~Q2 => ~Q-1 !=#, ~Q-2=A2-2 => R-1!=#  
W-2=Noop-W2 => W-1 !=#, W-2=A2-2 => R-1!=#

P-1=NooP-P1 => P-0!=#, Q-1=A1-1 => P-0!=#  R-1=NooP-R1 => R-0!=#  
~Q-1=A2-1 => R-0!=#, W-2=A2-1=> R-0!=#

Mutex Constraints:

Q-2=A1-2 => ~Q-2!=Noop~Q2, Q-2=Noop-Q2 => ~Q-2!=A2-2,  
Q-2=Noop-Q2 => W-2!=A2-2, Q-2=Noop-Q2 => ~Q-2!=Noop~Q2,  
~Q-2=A2-2 => Q-2!=A1-2, ~Q-2=A2-2 => Q-2!=Noop-Q2,  
~Q-2=Noop~Q2 => Q-2!=A1-2, ~Q-2=Noop~Q2 => Q-2!=Noop-Q2  
W-2=A2-2 => Q-2!=A1-2, W-2=A2-2 => Q-2!=Noop-Q2,  
Q-1=A1-1 => ~Q-1!=A2-1, Q-1=A1-1 => W-1!=A2-1,
\[ \neg Q-1 = A2-1 \Rightarrow Q-1 = A2-1, \quad W-1 = A2-1 \Rightarrow Q-1 = A1-1, \]

**Solution:**

\[ Q-2 = A1-2, \quad P-2 = \text{NooP-P2}, \quad R-2 = \text{NooP-R2}, \quad \neg Q-2 = \#, \quad W-2 = \# \]

\[ Q-1 = \#, \quad P-1 = \text{NooP-P1}, \quad R-1 = \text{NooP-R1}, \quad \neg Q-1 = A2-1 \quad W-1 = A2-1 \]

\[ P-0 = T, \quad R-0 = T \]

**III.b.** Do it with SAT encoding of the planning graph.

**SAT Encoding**

**Initial State:** P-0 & R-0

**Goal State:** Q-2 & W-2

**Graph Propagation:** (cond x at k => one of its supporting actions)

\[ P-2 \Rightarrow \text{NooP-P2} \quad Q-2 \Rightarrow \text{NooP-Q2} \quad A1-2 \quad \neg Q-2 \Rightarrow \text{NooP-\neg Q2} \quad A2-2 \]

\[ R-2 \Rightarrow \text{NooP-R2} \quad W-2 \Rightarrow \text{NooP-W2} \quad A2-2 \]

\[ P-1 \Rightarrow \text{NooP-P1} \quad Q-1 \Rightarrow A1-1 \quad \neg Q-1 \Rightarrow A2-1 \]

\[ R-1 \Rightarrow \text{NooP-R1} \quad W-1 \Rightarrow A2-1 \]

**Actions => Preconditions**

\[ A1-2 \Rightarrow P-1, \quad A2-2 \Rightarrow R-1, \quad \text{NooP-P2} \Rightarrow P-1, \quad \text{NooP-Q2} \Rightarrow Q-1, \]

\[ \text{NooP-\neg Q2} \Rightarrow Q-1, \quad \text{NooP-R2} \Rightarrow R-1, \quad \text{NooP-W2} \Rightarrow W-1, \]

\[ A1-1 \Rightarrow P-0, \quad A2-1 \Rightarrow R-0, \quad \text{NooP-P1} \Rightarrow P-0, \quad \text{NooP-R1} \Rightarrow R-0, \]

**Mutexes:**

\[ \neg Q-2 \Leftrightarrow \neg(\neg Q-2), \quad \neg A1-2 \Leftrightarrow \neg A2-2, \quad \neg A1-2 \Leftrightarrow \neg \text{NooP-\neg Q2} \]

\[ \neg A2-2 \Leftrightarrow \neg \text{NooP-Q2}, \quad \neg Q-1 \Leftrightarrow \neg(\neg Q-1), \quad \neg Q-1 \Leftrightarrow \neg W-1, \quad \neg A1-1 \Leftrightarrow \neg A2-1 \]

**Solution:** A2-1 & \text{NooP-P1} & \text{NooP-W2} & A1-2

**III.c.** (Added since I extended the deadline to 10/11 ~) Do an "explanatory axiom" (or backward proof based) encoding of this problem (for the same length as the planning graph you used in the previous parts). Mark which constraints are similar, different, stronger etc. compared to III.b.

Propositions corresponding to the initial conditions and goals are true at their respective levels (similar, in fact same):

\[ P_0 & R_0 & Q_2 & W_2 \]

At least one of the actions at each level will occur (stronger, although hard to tell exactly, hence it would be better to say that they could possibly be stronger):

\[ A1_1 \lor A2_1 \]

\[ A1_2 \lor A2_2 \]

Actions imply their preconditions and effects (somewhat different, now actions also imply their effects in addition to only their preconditions):

\[ A1_1 \Rightarrow P_0 \& Q_1 \]

\[ A2_1 \Rightarrow R_0 \& \neg Q_1 \& W_1 \]

\[ A1_2 \Rightarrow P_1 \& Q_2 \]

\[ A2_2 \Rightarrow R_1 \& \neg Q_2 \& W_2 \]
A proposition P changes values between j and j+1 only if an action occurs that makes it so (different)

\(\sim Q_0 \land Q_1 \Rightarrow A1_1\)
\(Q_0 \land \sim Q_1 \Rightarrow A2_1\)
\(\sim W_0 \land W_1 \Rightarrow A2_1\)
\(\sim Q_1 \land Q_2 \Rightarrow A1_2\)
\(Q_1 \land \sim Q_2 \Rightarrow A2_2\)
\(\sim W_1 \land W_2 \Rightarrow A2_2\)

No pair of interacting actions must occur together (somewhat different, in general there could be more mutexes than interactions, recall that interactions are static and mutexes may also be dynamic)

\(\sim A1_1 \lor \sim A2_1\)
\(\sim A1_2 \lor \sim A2_2\)