**Student Information**

**Course:** COMP90054 AI Planning for Autonomy

**Semester:** Semester 1, 2025

**Student:** your full name - your student number - your canvas student login

**Important:** Replace the lines above with your correct details. Your student number should only be the **numbers**. For example: Guang Hu - 000000 - ghu1.

**Collaborated With:**

**Important:** If you worked with another student, please include their **full name** and ask them to provide you with the **url to their github codebase**. Their codebase should be private, you will not have access to the code, so there's no issue with knowing their URL, e.g. Collaborated with: Lionel Messi - URL: github.com/happy-with-the-worldcup-until-the-next-one.

**Self Evaluation**

**Note:** Do not exceed 500 words for each Part. This is indicative, no need to have 500 words, and it's not a strict limit.

**Problem 1-3 (Domain Definition)**

**Self Evaluated Marks (2 marks):** 0

**Important:** Please replace the above 0 with the mark you think you earned for this part.

**Learning and Challenges**

**Tip:** Please include your top lessons learnt, and challenges faced. The questions below can help you structure your reflection.

1. How did you implement the domain to realise these scenarios?

* handle the behaviour of not being able to revisit a track segment already visited by the Route S tram.
* handle the switch properly, including ensuring that the Route S tram is on the segment and that the track being switched to is connected to the track.
* handle the request properly, including ensuring that the Route S tram is on the segment that is switched to the segment that the tram requested to move is on and that the requested tram is powered.
* handle the movement properly, including ensuring that the tram is powered, that the destination can be moved to without unpowering the tram.

1. How did you implement the three different problem files?
2. What thing that you've learned are you most excited about? What challenges have you encountered?

**Ideas That Almost Worked Well**

**Tip:** If you tried ideas that did not make it to the final code, please include them here and explain why they didn't make it.

**Comparison with sample solution**

**Justification**

Tip: Please state the reason why you have assigned yourself these marks.

**Problem 1-3 (Problem Definition)**

**Self Evaluated Marks (1 marks):** 0

**Important:** Please replace the above 0 with the mark you think you earned for this part.

**Learning and Challenges**

**Tip:** Please include your top lessons learnt, and challenges faced. The questions below can help you structure your reflection.

1. For problem 1: How did you implement the problem file? Is the plan returned by the planner the same as the sample plan (from README.md)? If not, what is the reason?
2. For problem 2: How did you implement the problem file? Is the plan returned by the planner the same as the sample plan (from README.md)? If not, what is the reason?
3. For problem 3: How did you implement the problem file? Is the plan returned by the planner the same as the sample plan (from README.md)? If not, what is the reason?

**Ideas That Almost Worked Well**

**Tip:** If you tried ideas that did not make it to the final code, please include them here and explain why they didn't make it.

**Comparison with sample solution**

**Justification**

**Tip:** Please state the reason why you have assigned yourself these marks.

**Problem 4 (Domain Extension and Sample Problem)**

**Self Evaluated Marks (4 marks):** 0

**Important:** Please replace the above 0 with the mark you think you earned for this part.

**Learning and Challenges**

**Tip:** Please include your top lessons learnt, and challenges faced. The questions below can help you structure your reflection.

1. Which considerations did you make in terms of realising the following scenarios?

* 1 point for handling more than one tram (with the capability to handle more),
* 1 point for making it time-based within a particular fixed window,
* 1 point for making it include destinations for trams to be at at a particular point in time,
* 1 point for creating a problem with a valid solution for all trams (should be complex enough to demonstrate the above criteria, if you cannot cover all elements with one instance, you can use multiple instances to achieve it).

1. What is the main bottleneck to scale up this problem? What is it that makes this problem hard?

**Ideas That Almost Worked Well**

**Tip:** If you tried ideas that did not make it to the final code, please include them here and explain why they didn't make it.

**Comparison with sample solution**

**Tip:** If you failed to get a valid plan or failed to implement a valid domain/problem, please compare your code with sample code and identify where you didn’t match (in the domain or problem file), and what you would do differently next time. It will potentially gain you more marks even though you did not generate the correct plan.

**Justification**

**Tip:** Please state the reason why you have assigned yourself these marks. If you did not figure it out before checking the sample solution, you need to explain with more details, **including a detailed comparison between your solution and the sample solution**.