

EECS4315-Z Winter 2025
Mission Critical Systems
Example Exam Questions

Name (Print): _____

PPY Login _____

Signature _____

This exam contains 7 pages (including this cover page) and 2 problems.

Check to see if any pages are missing.

Do not detach any question pages from the booklet.

Enter **all** requested information on the top of this page before you start the exam, and put your **initials** on the top of every page, in case the pages become separated.

Attempt **all** questions. Answer each question in the boxed space provided.

The following rules apply:

- **NO QUESTIONS DURING THE EXAM.** If a question is ambiguous or unclear, then write your assumptions and proceed to answer the question.
- Do **not** write your answers in the questions booklet. **Only answers written in the separate answers booklet will be graded.**
- Do **not** sketch your work in the answers booklet. **Only sketch on the blank pages attached to the questions booklet.**
- At the end of the exam, be sure to submit **all** the following: **1)** Exam questions booklet; **2)** Exam answers booklet(s); and **3)** Data sheet.
Each one of the above submissions **must** be written with your **full name** and **student number**.
If any of the above submissions is missing, your exam will not be graded.
- Where descriptive answers are requested, use complete sentences and paragraphs. Be precise and concise.
- **Organize your work**, in a reasonably neat and coherent way, in the space provided. Work scattered all over the page without a clear ordering will receive very little credit.
- **Mysterious or unsupported answers will not receive credit.** A correct answer, unsupported by calculations or explanation will receive no credit; an incorrect answer supported by substantially correct calculations and explanations might still receive partial credit.

Do not write in this table which contains your raw mark scores.

Problem	Points	Score
1	75	
2	25	
Total:	100	

1. Consider the following algorithm which computes the maximum value from an input tuple of integers:

```

----- MODULE findMax -----
EXTENDS Integers, Sequences, TLC
CONSTANT input
/* defines LI and invariant here
I(i, result) == \A j \in 1..i-1: result >= input[j]
V(i, inp) == Len(inp) - i + 1
(*)
--algorithm FindMax {
  variables result = input[1], i = 1, variant_pre = 0, variant_post = 0;
  {
    assert Len(input) > 0; /* precondition
    assert I(i, result); /* invariant
    while (i <= Len(input)) {
      variant_pre := V(i, input);

      if (input[i] > result) { result := input[i] };
      i := i + 1;

      variant_post := V(i, input);
      assert variant_post >= 0;
      assert variant_post < variant_pre;
      assert I(i, result); /* invariant
    };
    /* postcondition
    assert \A j \in 1..Len(input): result >= input[j]
  }
}
*)

```

- (a) State formally the obligation for proving that the loop invariant is established.

Requirement. Where a predicate is stated, it must be written in math form (translated from the given PlusCal syntax).

Solution:

$$\begin{aligned}
 & \{ \text{Len}(\text{input}) > 0 \} \\
 & \quad \text{result} := \text{input}[1]; \text{ i} := 1 \\
 & \{ \forall j \bullet j \in 1..i-1 \Rightarrow \boxed{1 \leq j \wedge j \leq \text{Len}(\text{input})} \wedge \text{result} \geq \text{input}[j] \}
 \end{aligned}$$

Notice that the augmented constraint $\boxed{1 \leq j \wedge j \leq \text{Len}(\text{input})}$ is for the tuple indexing expression $\text{input}[j]$ to be *well-defined*. Similar augmentation is required for each occurrence of tuple indexing.

[of 10 marks]

- (b) Prove or disprove the stated proof obligation from Part (a).

Requirement. Calculation and proof steps should be presented in the equational style. Each step should be as *atomic* as possible: do not skip or perform multiple steps at a time.

Solution:

- First, calculate:

$$\begin{aligned}
 & wp(\text{result} := \text{input}[1]; i := 1, \forall j \bullet j \in 1..i-1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]) \\
 = & \{ \text{wp rule of sequential composition} \} \\
 & wp(\text{result} := \text{input}[1], wp(i := 1, \forall j \bullet j \in 1..i-1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j])) \\
 = & \{ \text{wp rule of assignment} \} \\
 & wp(\text{result} := \text{input}[1], \forall j \bullet j \in 1..0 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]) \\
 = & \{ \text{wp rule of assignment} \} \\
 & \forall j \bullet j \in 1..0 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{input}[1] \geq \text{input}[j] \\
 = & \{ \text{arithmetic} \} \\
 & \forall j \bullet \text{false} \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{input}[1] \geq \text{input}[j] \\
 = & \{ \text{false} \Rightarrow p \equiv \text{true} \} \\
 & \forall j \bullet \text{true} \\
 = & \{ \text{arithmetic} \} \\
 & \text{true}
 \end{aligned}$$

- Then, prove that the precondition is no weaker than the calculate *wp*:

$$\text{Len}(\text{input}) > 0 \Rightarrow \text{true}$$

This is proved as $p \Rightarrow \text{true} \equiv \text{true}$ for any proposition p .

[of 20 marks]

- (c) State formally the obligation for proving that the loop invariant is maintained.

Requirement. Where a predicate is stated, it must be written in math form (translated from the given PlusCal syntax).

Solution:

$$\begin{aligned}
 & \{ i \leq \text{Len}(\text{input}) \wedge (\forall j \bullet j \in 1..i-1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]) \} \\
 & \quad \text{if}(\text{input}[i] > \text{result}) \{ \text{result} := \text{input}[i] \}; i := i + 1; \\
 & \{ \forall j \bullet j \in 1..i-1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j] \}
 \end{aligned}$$

Notice that the augmented constraint $1 \leq j \wedge j \leq \text{Len}(\text{input})$ is for the tuple indexing expression $\text{input}[j]$ to be *well-defined*. Similar augmentation is required for each occurrence of tuple indexing.

[of 10 marks]

- (d) Prove or disprove the stated proof obligation from Part (c).

Requirement. Calculation and proof steps should be presented in the equational style. Each step should be as *atomic* as possible: do not skip or perform multiple steps at a time.

[of 20 marks]

Solution to Part (d)

We first calculate the wp for the loop body to maintain the LI:

$$\begin{aligned}
 & wp(\text{if}(\text{input}[i] > \text{result}) \{ \text{result} := \text{input}[i] \}; i := i + 1; \boxed{\forall j \bullet j \in 1 \dots i - 1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]}) \\
 = & \{ wp \text{ rule for sequential composition} \} \\
 & wp(\text{if}(\text{input}[i] > \text{result}) \{ \text{result} := \text{input}[i] \}, wp(i := i + 1, \boxed{\forall j \bullet j \in 1 \dots i - 1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]})) \\
 = & \{ wp \text{ rule for assignment} \} \\
 & wp(\text{if}(\text{input}[i] > \text{result}) \{ \text{result} := \text{input}[i] \}, \boxed{\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]}) \\
 = & \{ wp \text{ rule for conditional} \} \\
 & \text{input}[i] > \text{result} \Rightarrow wp(\text{result} := \text{input}[i], \boxed{\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]}) \\
 & \wedge \\
 & \text{input}[i] \leq \text{result} \Rightarrow wp(\text{result} := \text{result}, \boxed{\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]}) \\
 = & \{ wp \text{ rule for assignment, twice} \} \\
 & \text{input}[i] > \text{result} \Rightarrow (\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{input}[i] \geq \text{input}[j]) \\
 & \wedge \\
 & \text{input}[i] \leq \text{result} \Rightarrow (\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j])
 \end{aligned}$$

We then prove that the precondition (i.e., Stay Condition $\wedge$ LI) is no weaker than the above calculated wp :

- To prove the left conjunct:
$$\begin{aligned}
 & i \leq \text{Len}(\text{input}) \wedge (\forall j \bullet j \in 1 \dots i - 1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]) \Rightarrow \\
 & \quad \text{input}[i] > \text{result} \Rightarrow \boxed{\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{input}[i] \geq \text{input}[j]} \\
 & \quad \equiv \{ \text{Shunting: } p \Rightarrow (q \Rightarrow r) \equiv (p \wedge q) \Rightarrow r \} \\
 & \quad i \leq \text{Len}(\text{input}) \wedge (\forall j \bullet j \in 1 \dots i - 1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]) \wedge \text{input}[i] > \text{result} \Rightarrow \\
 & \quad \boxed{\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{input}[i] \geq \text{input}[j]}
 \end{aligned}$$

Proof via Assuming the Antecedent:

$$\begin{aligned}
 & \boxed{\forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{input}[i] \geq \text{input}[j]} \\
 = & \{ \text{split range: } \forall j \bullet j \in 1 \dots i \Rightarrow P(j) \equiv (\forall j \bullet j \in 1 \dots i - 1 \Rightarrow P(j)) \wedge P(i) \} \\
 & (\forall j \bullet j \in 1 \dots i - 1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{input}[i] \geq \text{input}[j]) \wedge (1 \leq i \wedge i \leq \text{Len}(\text{input}) \wedge \text{input}[i] \geq \text{input}[i]) \\
 = & \{ \text{antecedent: } \text{input}[i] > \text{result}; \text{ and RHS of precondition: } \forall j \bullet j \in 1 \dots i - 1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j] \} \\
 & \text{true} \wedge (1 \leq i \wedge i \leq \text{Len}(\text{input}) \wedge \text{input}[i] \geq \text{input}[i]) \\
 = & \{ \text{LHS of precondition: } i \leq \text{Len}(\text{input}) \text{ and } \text{input}[i] \geq \text{input}[i] \equiv \text{true} \} \\
 & \text{true}
 \end{aligned}$$

- (Exercise) To prove the right conjunct:

$$\begin{aligned}
 & i \leq \text{Len}(\text{input}) \wedge (\forall j \bullet j \in 1 \dots i - 1 \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]) \\
 & \Rightarrow \text{input}[i] \leq \text{result} \Rightarrow \forall j \bullet j \in 1 \dots i \Rightarrow 1 \leq j \wedge j \leq \text{Len}(\text{input}) \wedge \text{result} \geq \text{input}[j]
 \end{aligned}$$

- (e) Refer to the algorithm **findMax** at the start of this question. Consider a change of the loop invariant to:

$$\forall j \in 1..i: \text{result} \geq \text{input}[j]$$

Say the algorithm is run on an input tuple $\langle 20, 10, 40, 30 \rangle$. Describe how a loop invariant violation, if any, will occur.

Solution:

- After the initialization steps, value of i becomes 1 and value result becomes 20 ($\text{input}[1]$), and the loop invariant $\forall j \in 1..1 \Rightarrow \text{result} \geq a[j]$ reduces to $a[1] \geq a[1]$, which is true.
- At the end of the 1st iteration, value of result remains 20 and value of i gets incremented to 2, and the loop invariant $\forall j \in 1..2 \Rightarrow \text{result} \geq a[j]$ is true ($\because 20 \geq 20 \wedge 20 \geq 10$).
- At the end of the 2nd iteration, value of result remains 20 and value of i gets incremented to 3, and the loop invariant $\forall j \in 1..3 \Rightarrow \text{result} \geq a[j]$ is **false** ($\because 20 \not\geq \text{input}[3] = 40$).

[of 15 marks]

2. Consider the following claim relating two path satisfactions:

$$\pi \models \mathbf{G} \phi \iff \pi \models \neg (\mathbf{F} \neg \phi)$$

where π is any path that is valid for the model (i.e., some LTS) in question, and ϕ is any arbitrary LTL formula that is syntactically correct. Prove or disprove the above claim.

Solution:

The claim is valid and here's a proof:

$$\begin{aligned}
 & \pi \models \mathbf{G} \phi \\
 \iff & \{ \text{Definition of path satisfaction of } \mathbf{G} \} \\
 & \forall i \bullet i \geq 1 \Rightarrow \pi^i \models \phi \\
 \iff & \{ \text{Known Theorem: } \forall X \bullet R(X) \Rightarrow P(X) \equiv \neg(\exists X \bullet R(X) \wedge \neg P(X)) \} \\
 & \neg(\exists i \bullet i \geq 1 \wedge \neg(\pi^i \models \phi)) \\
 \iff & \{ \pi \models \neg \phi \iff \neg(\pi \models \phi) \} \\
 & \neg(\exists i \bullet i \geq 1 \wedge \pi^i \models \neg \phi) \\
 \iff & \{ \text{Definition of path satisfaction of } \mathbf{F} \} \\
 & \neg (\mathbf{F} \neg \phi)
 \end{aligned}$$

[of 25 marks]

This is a blank page for sketching purpose. You may detach it from the exam booklet.

Do **not** detach other question pages from the exam booklet.

This is a blank page for sketching purpose. You may detach it from the exam booklet.

Do **not** detach other question pages from the exam booklet.