

For all questions that mention FSP, you must write them in the LTSA and your code must be runnable. Except for Q5.

Recommendations for students.

- Students will have a rich format text box for each question,
- This text box allows students to copy and paste solutions directly from Microsoft Word into any file format's text box or attach any file format to that box if needed.
- Please finish the assignment on your local computer, and save each solution as a separate file in an appropriate format. And, in the end, **either copy and paste your solution to each avenue textbox or attach your solution.**
- **PLEASE DO NOT USE THE SUBMIT button every time to SAVE your work. Only use the submit button for your final submission.**
- Make sure to **save your work** for each question in the avenue **by clicking on the outside of the textbox box** in the avenue, which will allow avenue you to save your response.
- **You must** submit Java files for any Java question.
- You must submit *.txt* or *.LTSA* file(s) for your LTSA questions and paste your code inside the text box. We must be able to copy and paste your code into their respective IDEs and run it without any issues. Please **double-check your code** (i.e. Java, LTSA, etc. before submitting it). The non-working code will be **heavily** penalized or given a zero. The TA's should be able to copy/paste your code and run it without any issues.
- For hand-drawn diagrams, you can attach a photo, we recommend **PNG** over JPG.
- Students can easily create a snapshot of their LTS diagrams by using **the windows snipping tool (Grab for Mac)**.
- Students, who do not have Microsoft Word on their computer, are suggested to use google document editor ([Google Docs](#)).
- For questions that do not involve using the LTSA tool or producing FSPs, hand-drawn pictures are allowed, but a solution should be in image format inserted or attached into the textbox (if a student need to scan a handwritten note and insert it into the textbox is encouraged to use a smartphone app called [CamScanner](#)).
- There will be a mark deduction for not following the submission instruction. Students must submit their assignments to [Avenue](#). Any problem with Avenue, please discuss with Mahdee Jodayree <mahdijaf@yahoo.com>, a TA for this course.

Question 1)

There are a total of six states in this question

In this question

Instead of creating 3 different philosophers, you must first define a single philosopher (There is a single state for the philosopher and you must describe its actions similar to the example in the tutorial, however, read the question carefully.) and then use guarded actions (there are 3 guarded actions.) to describe the composition of all 3 philosophers by using guarded actions. Please see the **LTSA examples chapter3-->Switches. its example**. This example shows you how to create a single State for a switch then create a composition that shows how 3 switches would run concurrently.

```
PHIL = (think -> PHIL | get_cookie -> eat_cookie -> PHIL
```

```
| getCola -> drinkCola -> PHIL ).
```

```
const K = 3
```

```
range Phil = 1..K
```

```
|| PHILS = ( forall[i: Phil] phil[i]:PHIL ).
```

```
( This is how can create 3 compositions of a single state. )
```

After the above code, you must create a State for **Servant**

Then create a State for **Cookies** that have **4 guarded actions** (0..3) but only two **statements** (1 of them is an if statement and the second one is a condition without any if statements) for the guarded actions. There is a single bar for Cookies. (**See assignment 1 question 10**).

Then create a State for Cola with a total of 3 guarded actions (0 . . 2) and similar to the cookies (See assignment 1 question 10)

Now you must define two properties called SCOLA and SCOOKIE (**Lecture 11**)

```
property SCOLA = SCOLA[M],
```

```
SCOLA[s: Cola] = (when (s==0) fillCola-> SCOLA[M] ).
```

After this line writes a property for **SCOOKIE** , it is very similar to the previous line for the Scola property.

And at the end, You must create a composition with is a composition of all previous states that you have created including (previous composition of all 3 philosophers), cookies, Cola, Servant, Scola, SCOOKIE. The final composition should not have any mutual exclusion.

Question 2)

You must define a state for **DOOR** and define its actions based on the actions described in the question

You must create another state for the director and define their actions

The state for the CONTROL contains 5 guarded actions (0 ...4) with different conditions, Very similar to (assignment 1, Q10).

In the end, there is a final composition which is a composition of all previously defined states (we defined 3 states in this question), there is no mutual exclusion in this composition however you must add labelling for this composition, the actions for the east door and west door must be shown in the state Door). Please watch tutorials to understand how to add labelling to the composition.

Question 3)

You must create a state for car-park control which should have 4 guarded actions. With two different conditions (similar to assignment 1 Q10). LECTURE note 7, slide 5 but you must write a safety property. Please read the hint for Q1 and understand how to write a safety property.

You must create a state for arrivals

And another state for departures.

Then you must create a composition for car-park would be the composition of all 3 states that you have created so far.

Then you must create a property called overflown with 4 guarded actions this property does not have any if statements however it defines what two actions of arriving and depart would do.

property OVERFLOW (N=4) =

Then you must create a composition called check car park and this would be the composition of your previously defined property and previously defined composition.

At the end define a final composition which would be the composition of car-park while depart has lesser priority.

Question 4)

There are a total of 4 different states you must define each state appropriately.

In the end, there is a composition which is a composition of 4 philosophers and all philosophers are **mutually excluded** to the appropriate fork.

Question 5)

There is no LTSA code for this question, however, you must write an FSP (not an FSP code for LTSA)

Question 6)

LTSA code for this question is very similar to Assignment 1 question 10, however, there are a total of 11 guarded actions (010)

Question 7)

There is no LTSA coding for this question, however, you must first draw LTS by hand then you must use your hand-drawn diagram and write a safety property and this property to find which actions would lead to ERROR and specify that in your new handwritten FSP.

Question 8)

Part a)

You must create 3 different states for each smoker.

You must create a state for tobacco

Another state for paper

Another state for the match

You must also create 3 different states for each agent that will be delivering each item

You must create another state or rule.

You must create 1 composition for smokers (this should be the composition of 3 different smokers)

1 composition for all resources (this should be the composition of 3 different resources that we previously defined and mutually excluded to 4 different actions)

Another composition for the agent rule (this should be the composition of all three agents and the state rule and mutually excluded to 4 different actions).

In the end, there must be a final composition that would be composed (composition) of all three previously defined compositions.

Part b)

Add safely actions to your code on part a)