6SENG002W Concurrent Programming

FSP Process Analysis & Design Form

Name	Pasan Kottearachchi
Student ID	W1761885
Date	28/12/2022

1. FSP Process Attributes

Attribute	Value
Name	PRINTER
Description	This is a shared printer that is used by the students for printing documents and the technician to refill the papers
Alphabet	{stud_1, stud_2, tech}.{documentPrint[1..3], stud.{acquire, release}, technician.{acquire, refillPrinterSheets, release}}
Number of States	18
Deadlocks (yes/no)	No
Deadlock Trace(s) (if applicable)	N/A

2. FSP Process Code

FSP Process:

```
//Define the CONSTANTS of the system
const MAX_NO_OF_PAPER = 3
range PAPER_RANGE = 1 .. MAX_NO_OF_PAPER

//Initialise the process of printer
PRINTER(SHEET_COUNT = MAX_NO_OF_PAPER) = PRINT_PAPER[SHEET_COUNT],
PRINT_PAPER[sheet : 0 .. SHEET_COUNT ] = (
 when(sheet > 0) stud.acquire -> documentPrint[PAPER_RANGE] -> stud.release ->
PRINT_PAPER[sheet - 1]
|
 when (sheet == 0) technician.acquire -> technician.refillPrinterSheets -> technician.release ->
PRINT_PAPER[MAX_NO_OF_PAPER]
) .
```

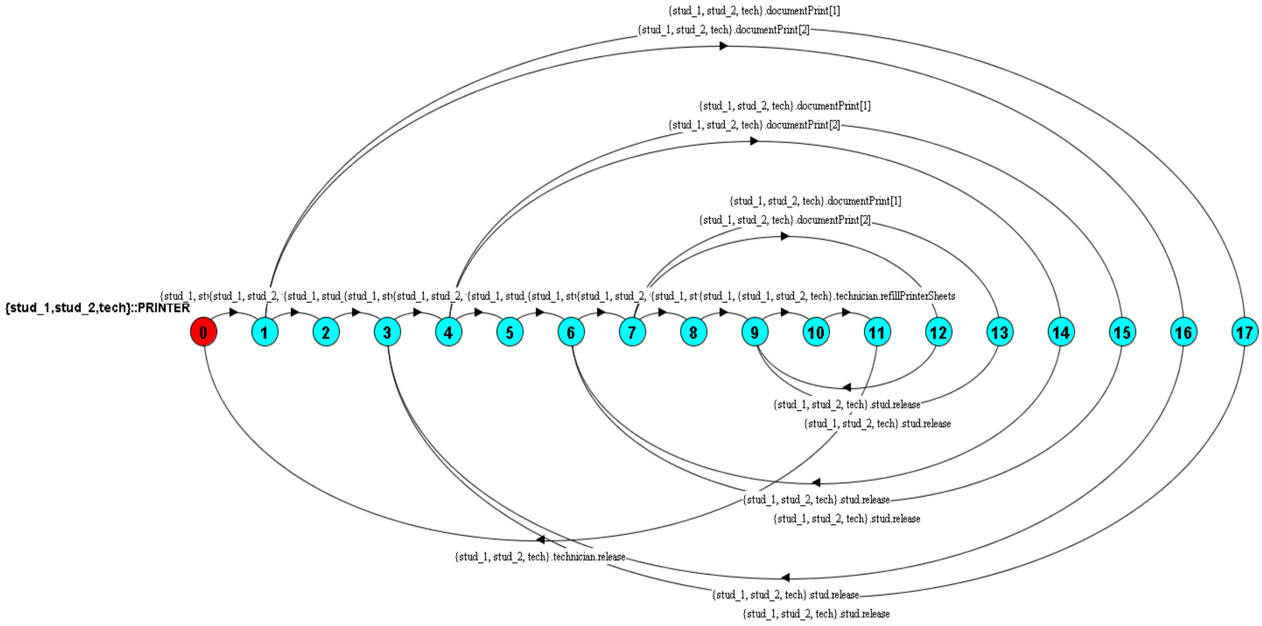
3. Actions Description

A description of what each of the FSP process' actions represents, i.e. is modelling. In addition, indicate if the action is intended to be synchronised (shared) with another process or asynchronous (not shared). (Add rows as necessary.)

Actions	Represents	Synchronous or Asynchronous
stud.acquire	Student acquire to control of the printing process.	Synchronous
documentPrint[1]	Print the document 1	Synchronous
documentPrint[2]	Print the document 2	Synchronous
documentPrint[3]	Print the document 3	Synchronous
stud.release	Student release the printer after printing a document.	Synchronous
technician.acquire	Technician acquire to control of the printing process.	Synchronous
technician.refillPrinterSheets	Technician refilling the sheets.	Synchronous
technician.release	Technician release the printer after refilling the sheets.	Synchronous

4. FSM/LTS Diagrams of FSP Process

Note that if there are too many states, more than 64, then the LTSA tool will not be able to draw the diagram. In this case draw small diagrams of the most important parts of the complete diagram.


{stud_1,stud_2,tech}::PRINTER
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
{stud_1, stud_2, tech}.documentPrint[1]
{stud_1, stud_2, tech}.documentPrint[2]
{stud_1, stud_2, tech}.documentPrint[1]
{stud_1, stud_2, tech}.documentPrint[2]
{stud_1, stud_2, tech}.documentPrint[1]
{stud_1, stud_2, tech}.documentPrint[2]
{stud_1, st{stud_1, stud_2, {stud_1, stud {stud_1, st{stud_1, stud_2, {stud_1, stud {stud_1, st{stud_1, stud_2, {stud_1, st{stud_1, {stud_1, stud_2, tech}.technician.refillPrinterSheets
{stud_1, stud_2, tech}.stud.release
{stud_1, stud_2, tech}.stud.release
{stud_1, stud_2, tech}.stud.release
{stud_1, stud_2, tech}.stud.release
{stud_1, stud_2, tech}.technician.release
{stud_1, stud_2, tech}.stud.release
{stud_1, stud_2, tech}.stud.release

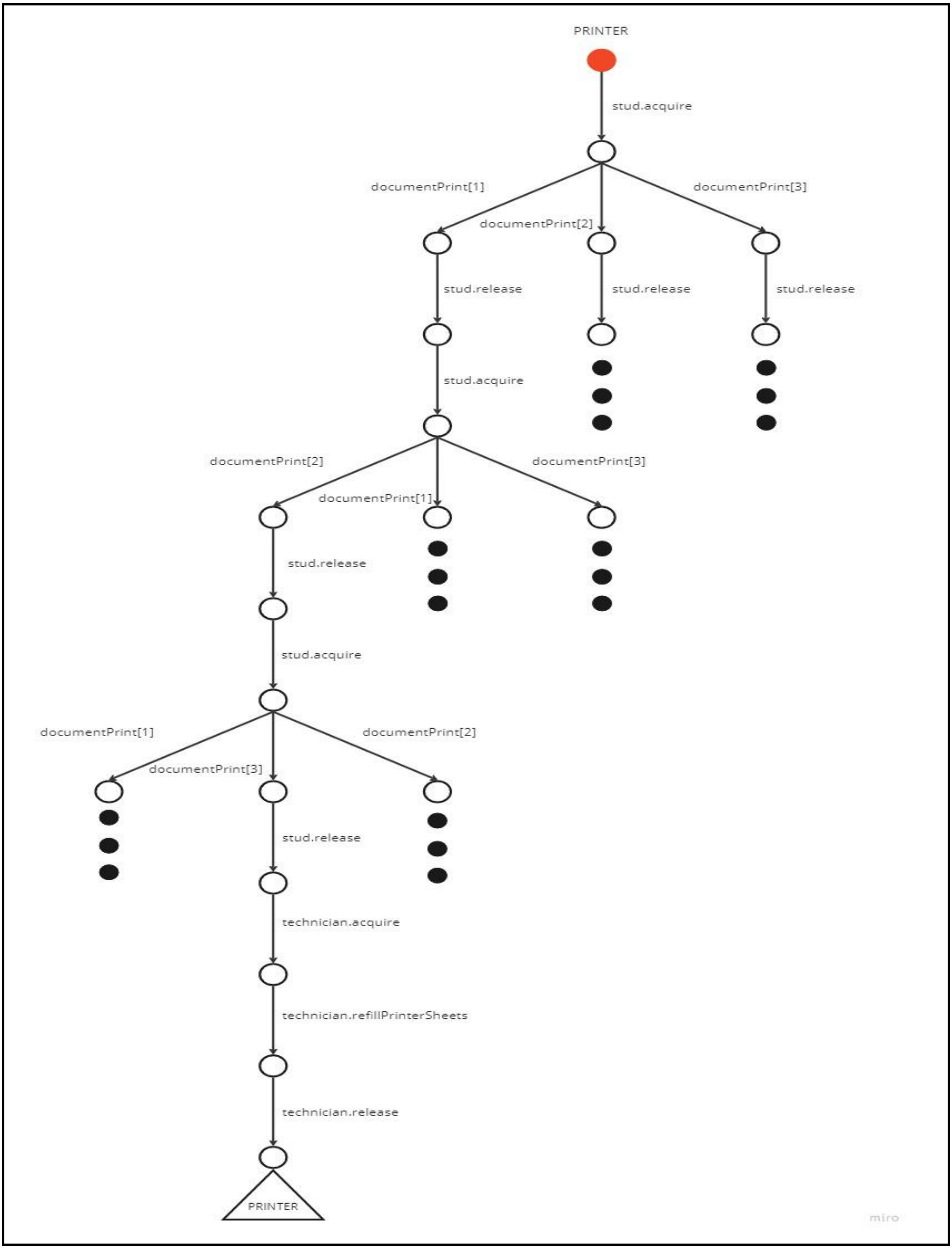
5. LTS States

A description of what each of the FSP process' states represents, i.e. is modelling. If there are a large number of states then you can group similar states together &/or only include the most important ones. For example, identify any states related to mutual exclusion (ME) & the associated critical section (CS), e.g. waiting to enter the CS state, in the CS state(s), left the CS state. (Add rows as necessary.)

State	Represents
0	Printer is acquired to used student.
1, 4, 7	Student can acquired to control the printer.
2, 5, 8, 12, 13, 14, 15, 16, 17	Document can has been printed.
3, 6, 9	Student released the printer after printing
10	Technician can acquired to refill the sheets.
11	Technician refilled the sheets and ready to used

6. Trace Tree for FSP Process

The trace tree for the process. Use the conventions given in the lecture notes and add explanatory notes if necessary.


PRINTER
stud.acquire
documentPrint[1]
documentPrint[2]
documentPrint[3]
stud.release
stud.release
stud.release
stud.acquire
documentPrint[2]
documentPrint[1]
documentPrint[3]
stud.release
stud.acquire
documentPrint[1]
documentPrint[3]
documentPrint[2]
stud.release
technician.acquire
technician.refillPrinterSheets
technician.release
PRINTER
miro