6SENG002W Concurrent Programming

FSP Process Analysis & Design Form

1. FSP Process Attributes

Attribute	Value
Name	PRINTER
Description	Represents the printer machine that can print documents.
Alphabet	{ student_1.acquire, student_1.notEnoughPaper, student_1.outOfPaper, student_1.paperNotEmpty, student_1.print[0..3], student_1.release, student_1.servicePrinter, student_2.acquire, student_2.notEnoughPaper, student_2.outOfPaper, student_2.paperNotEmpty, student_2.print[0..3], student_2.release, student_2.servicePrinter, tech.acquire, tech.notEnoughPaper, tech.outOfPaper, tech.paperNotEmpty, tech.print[0..3], tech.release, tech.servicePrinter }
Number of States	26
Deadlocks (yes/no)	No deadlocks / errors
Deadlock Trace(s)	None

2. FSP Process Code

FSP Process:

```
PRINTER( S = 3 ) = PRINTER[ S ],
PRINTER[ i : PAPR_LVL ]
=  if ( i != 0 )
   then ( acquire ->( print[ doc : DOC_ID ] -> release -> PRINTER[ i - 1 ]
                    | paperNotEmpty -> release -> PRINTER[ i ] ) )

   else ( acquire -> ( outOfPaper -> servicePrinter -> release -> PRINTER[ i + 3 ]
                    | notEnoughPaper -> release -> PRINTER[ i ] )
   ) .
```

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
acquire	Acquires the printer lock to ensure mutual exclusion	Synchronous
release	Releases the printer lock so other processes can claim lock	Synchronous
print[DOC_ID]	Prints the document of a student process	Synchronous
notEnoughPaper	When paper level is zero and a student tries to print it will say "notEnoughPaper"	Synchronous
outOfPaper	When paper level is zero and the technician acquires lock it will then proceed to service the printer	Synchronous
servicePrinter	The technician then services the printer	Synchronous
paperNotEmpty	When paper level is not zero and technician acquires lock it will say "paperNotEmpty" as it doesn't need to refill the papers	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.
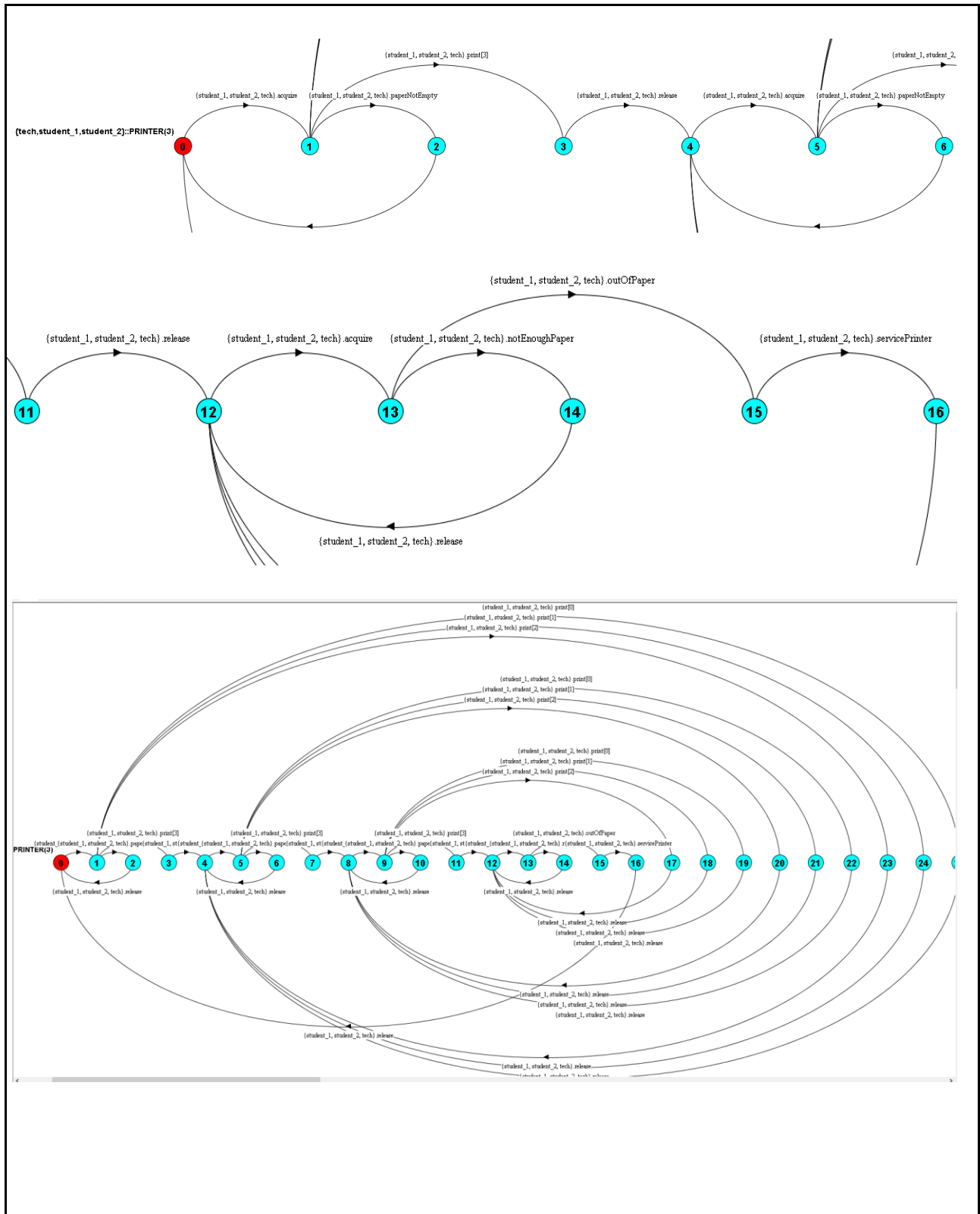
5. LTS States

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.)

States	Represents
0, 4, 8, 12	States where the students or technician can acquire the lock for the printer
2, 3, 6, 7, 10, 11, 14, 16 and (17-25)	States where the students or technician releases the lock for the printer
1, 5,	States where there's an option for students to print or for technician to refill however there's enough paper
13, 14	States where students chose to print however there's not enough paper so they will release the lock
13, 15	States where there's not enough paper so technician services the printer then releases lock

6. Trace Tree for FSP Process

The trace tree for the process. Use the conventions given in the lecture notes.

PRINTER

acquire

release   print[DOC_ID]   paperNotEmpty   release

PRINTER[i + 3]

notEnoughPaper   outOfPaper

PRINTER[i]

release   servicePrinter

PRINTER[i]   release

PRINTER[i + 3]