**6SENG006C Concurrent Programming**

**FSP Process Analysis & Design Form**

|  |  |
| --- | --- |
| **Name** | W. Thaveesha Dheerasekera |
| **Student ID** | W1767369/ 2019483 |
| **Date** | 09/01/2023 |

**1. FSP Process Attributes**

|  |  |
| --- | --- |
| **Attribute** | **Value** |
| **Name** | PRINTER |
| **Description** | This is the basic procedure that simulates how a printer behaves in a printing system. This printer can be used to print up to 3 papers, then must refill. |
| **Alphabet** | {student.{acquire, print.paper, release}, technician.{acquire, refill, release}, waiting} |
| **Number of States** | 13 |
| **Deadlocks (yes/no)** | No |
| **Deadlock Trace(s)**  **(If applicable)** | Not Applicable |

**2. FSP Process Code**

|  |
| --- |
| **FSP Process:** |
| // CONSTANTS  const MAX\_PAPER = 3  range PAPER\_RANGE = 0 .. MAX\_PAPER  set PRINTER\_ACTIONS = { student.acquire , student.print.paper , student.release, technician.acquire , technician.refill , technician.release }  set Students = { a, b }  // PRINTER FSM  PRINTER ( N = MAX\_PAPER ) = PRINT[ N ],  PRINT[ papercount : PAPER\_RANGE ] = (  when( papercount > 0 ) student.acquire -> student.print.paper -> student.release -> PRINT[ papercount - 1 ] |  when( papercount == 0 ) technician.acquire -> technician.refill -> technician.release -> PRINT[ 3 ] | waiting -> WAIT  ), WAIT = ( waiting -> WAIT ). |

**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** |
| student.acquire | Student taking control of the printer | Synchronous |
| student.print.paper | Student printing a paper | Synchronous |
| student.release | Student leaving the printer after the completion | Synchronous |
| technician.acquire | Technician taking control of the printer | Synchronous |
| technician.refill | Technician refilling papers | Synchronous |
| technician.release | Technician leaving after refilling papers | Synchronous |
| waiting | Printer waiting for the next operation | Asynchronous |

**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.

|  |
| --- |
| Diagram  Description automatically generated |

**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 ready to be used |
| 1, 4, 7 | The printer has been taken over by a student |
| 2, 5, 8 | The student has a printing job going on |
| 3, 6, 9 | The control has been released by the student and the printer is now available for usage |
| 10 | The printer is waiting for a job input |
| 11 | A technician has taken control of the printer |
| 12 | Technician has refilled the paper and printer is now available for usage |

**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.

|  |
| --- |
|  |