

6SENG006W Concurrent Programming

FSP Process Composition Analysis & Design Form

Name	Sam Clark
Student ID	W1854525
Date	05/01/2024

1. FSP Composition Process Attributes

Attribute	Value
Name	TicketMachine
Description	Represents a ticket machine where the passenger can print tickets, paper technician refills paper and toner technician refills toner.
Sub-processes (List them.)	PAPER_TECH, TONER_TECH, PASSENGER, PAPER_MACHINE, TONER_MACHINE, TICKET_MACHINE
Number of States	3104
Deadlocks (yes/no)	No
Deadlock Trace(s) (If applicable)	N/A

2. FSP "main" Program Code

The code for the parallel composition of all of the sub-processes and the definitions of any constants, ranges & process labelling sets used. (Do not include the code for the individual sub-processes.)

FSP Program:

```
const MAX_PAPER = 5
const MAX_TONER = 5

range PAPER_CAPACITY = 0 .. MAX_PAPER
range TONER_CAPACITY = 0 .. MAX_TONER

const MIN_PRINT = 1
const MAX_PRINT = 8

range TICKETS_TO_PRINT = MIN_PRINT .. MAX_PRINT

PAPER_TECH(INITIAL_PAPER = 3) = Paper[INITIAL_PAPER] ,
TONER_TECH(INITIAL_TONER = MAX_TONER) = Toner[INITIAL_TONER] ,
PAPER_MACHINE(INITIAL_PAPER = 3) = Paper_Machine[INITIAL_PAPER] ,
TONER_MACHINE(INITIAL_TONER = 5) = Toner_Machine[INITIAL_TONER] ,
||TICKET_MACHINE(INITIAL_PAPER = 3, INITIAL_TONER = 5) = (PAPER_MACHINE ||
TONER_MACHINE).

||PURCHASE_TICKET_SYSTEM = (PASSENGER || PAPER_TECH || TONER_TECH ||
TICKET_MACHINE) .
```

3. Combined Sub-processes

(Add rows as necessary.)

Process	Description
PAPER_TECH	Represents a paper technician that refills the paper when it runs out
TONER_TECH	Represents a toner technician that refills the toner when it runs out
PASSENGER	Represents a passenger using the ticket machine to print tickets
PAPER_MACHINE	Represents the paper section of the ticket machine, combined with toner machine

TONER_MACHINE	Represents the toner section of the ticket machine, combined with paper machine
TICKET_MACHINE	Combination of the paper machine and toner machine sections that make up the ticket machine

4. Analysis of Combined Process Actions

- **Alphabets** of the combined processes, including the final process labelling.
- **Synchronous** actions are performed by at least two sub-process in the combination.
- **Blocked Synchronous** actions cannot be performed, because at least one of the sub-processes can never perform them, because they were added to their alphabet using alphabet extension.
- **Asynchronous** actions are performed independently by a single sub-process.

Group actions together if appropriate, e.g. if they include indexes in[0], in[1], ..., in[5] as in[1..5]. Add rows as necessary.

Processes	Alphabet (Use LTSA's compressed notation , if alphabet is large.)
PASSENGER	ps.acquire, ps.printTicket, ps.release, ps.finish, remainingTickets[1..8]
PAPER_TECH	p.acquire, p.release, p.wait, p.refillPaper
TONER_TECH	t.acquire, t.release, t.wait, t.refillToner
TICKET_MACHINE	checkPaper[0..5], checkToner[0..5], p.acquire, p.refillPaper, p.release, p.wait, ps.acquire, ps.printTicket, ps.release, t.acquire, t.refillToner, t.release, t.wait

Synchronous Actions	Synchronised by Sub-Processes (List)
p.acquire, p.release, p.refillPaper	PAPER_TECH, TICKET_MACHINE
Ps.acquire, ps.printTicket, ps.release	PASSENGER, TICKET_MACHINE
t.acquire, t.release, t.refillToner	TONER_TECH, TICKET_MACHINE

Blocked Synchronous Actions	Blocking Processes	Blocked Processes
t.acquire, t.release, t.wait, t.refillToner	TONER_TECH	PASSENGER, PAPER_TECH
p.acquire, p.release, p.wait, p.refillPaper	PAPER_TECH	PASSENGER, TONER_TECH
ps.acquire, ps.printTicket, ps.release	PASSENGER	PAPER_TECH, TONER_TECH

Sub-Processes	Asynchronous Actions (List)
PAPER_TECH	p.acquire, p.refillPaper, p.release, p.wait
TONER_TECH	t.acquire, t.refillToner, t.release, t.wait
PASSENGER	ps.acquire, ps.printTicket, ps.finish, ps.wait, ps.release

5. Parallel Composition Structure Diagram

The structure diagram for the parallel composition.

