

6SENG006W Concurrent Programming

FSP Process Composition Analysis & Design Form

Name	P. Howells
Student ID	
Date	18/10/22

1. FSP Composition Process Attributes

Attribute	Value
Name	VendingSystem
Description	Models a drinks vending machine, a coffee customer & a tea customer.
Alphabet (Use LTSA's compressed notation , if alphabet is large.)	{ cc.blue, cc.choose_coffee, cc.coffee, cc.drink_coffee, cc.pickup_coffee, cc.red, cc.tea, tc.blue, tc.choose_tea, tc.coffee, tc.drink_tea, tc.pickup_tea, tc.red, tc.tea }
Sub-processes (List them.)	DRINKS, COFFEE_CUSTOMER, TEA_CUSTOMER
Number of States	25 (0 .. 24)
Deadlocks (yes/no)	No deadlocks/errors
Deadlock Trace(s) (If applicable)	None

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:

```
set VMActions = { red, coffee, blue, tea }

set Customers = { cc, tc }

// The 2 customers
|| CUSTOMERS = ( cc:COFFEE_CUSTOMER || tc:TEA_CUSTOMER ) .

// Drinks Vend Machine & Customers

|| VendingSystem = ( Customers :: DRINKS || CUSTOMERS ) .
```

3. Combined Sub-processes

(Add rows as necessary.)

Process	Description
DRINKS	Represents a simple drinks vending machine offering tea & coffee.
COFFEE_CUSTOM	Represents a customer who wants a cup of coffee from the machine.
TEA_CUSTOMER	Represents a customer who wants a cup of tea from the machine.
CUSTOMERS	Represents the 2 customers

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
{cc, tc}::DRINKS	cc.blue, cc.coffee, cc.red, cc.tea, tc.blue, tc.coffee, tc.red, tc.tea
cc:COFFEE_CUSTOMER	cc.blue, cc.choose_coffee, cc.coffee, cc.drink_coffee, cc.pickup_coffee, cc.red, cc.tea
tc:TEA_CUSTOMER	tc.blue, tc.choose_tea, tc.coffee, tc.drink_tea, tc.pickup_tea, tc.red, tc.tea

Synchronous Actions	Synchronising Processes (List)
cc.coffee, cc.red,	DRINKS, COFFEE_CUSTOMER
tc.red, tc.tea	DRINKS, TEA_CUSTOMER
n/a	COFFEE_CUSTOMER, TEA_CUSTOMER

Blocked Synchronous Actions	Blocking Processes	Blocked Processes
cc.red, cc.tea	COFFEE_CUSTOMER	DRINKS
tc.blue, tc.coffee	TEA_CUSTOMER	DRINKS

Sub-Process	Asynchronous Actions (List)
DRINKS	None
COFFEE_CUSTOMER	cc.choose_coffee, cc.pickup_coffee, cc.drink_coffee
TEA_CUSTOMER	tc.choose_tea, tc.pickup_tea, tc.drink_tea

5. Parallel Composition Structure Diagram

The structure diagram for the parallel composition.

