Concurrency, Parallelism and Distributed Systems (CPDS)
Module I: Concurrency
Facultat d'Informàtica de Barcelona
Final Exam
November 13, 2019

Answer the questions concisely and precisely
Answer each problem in a separate page (remember to put your name)
Closed-book exam
Duration: 2 hour

Exercise 1 (5 Points) *Money Client Server.*
First of all, let us consider the *client side*:

```
CLIENT = (wallet.authorise -> service.request ->
              (service.reply -> CLIENT |service.abort -> CLIENT)
          ).
```

do the following:

- (1 Points) Give a picture of the LTS corresponding to `CLIENT`.
- (1/2 Points) Make a picture of

  ```
  client:CLIENT/{ service/{client,server}.service}
  ```

The *server side* is given by:

```
SERVER = (service.request -> wallet.invoice
          -> (wallet.confirm -> service.reply ->SERVER
             |wallet.default -> service.abort ->SERVER)
          ).
```

- (1 Points) Give the LTS corresponding to `SERVER`.

The client/sserver is given by:

```
||SES =(client:CLIENT||server:SERVER)/{ service/{client,server}.service}.
```

- (1 Points) Give the LTS corresponding to `||SES`.

Finally, let us ask for properties:

- Given the safety property:

  ```
  property HONEST
         = (server.wallet.confirm -> service.reply -> HONEST
            |server.wallet.default-> service. abort->HONEST).
  ```

  - (1 Points) Discuss if `SES` verifies `HONEST`. If is not the case, give a shortest trace to the ERROR state.

- (1/2 Points) Design the liveness property `LIVE_SERVICE` such that: *It should always be the case that the service either eventually replies or aborts.*

Exercise 2 (2 Points) *Java.*
Given the followng bounded buffer:

```
BOUNDED_BUFF(N=5) = BUFF[0],
BUFF[i:0..N] = (when(i>0) get->BUFF[i-1] |when(i<N) put -> BUFF[i+1]).
```

Implement a Java class `Buffer` in the following two ways:

- (1 Point) First, using the standart coding by `wait`, `notifylAll`.
- (1 Point) Second, using a busy waiting approach.

Exercise 3 (2 Points) *Two Words Translator.*
Please write a module contaning a simple translation program that gets a word in Spanish and prints an English translation The process should run in a loop, waiting for words to translate.
At the beginning just translates the words `casa` and `blanca`.
Following a trace of a possible execution:

```
28> Pid = spawn(fun translate:loop/0).
<0.156.0>
29> Pid ! "casa".
house
"casa"
30> Pid ! "blanca".
white
"blanca"
31> Pid ! "rosada".
I do not understand
"rosada"
```

Exercise 4 (1 Points) *Java/Erlang.*
Please describe shorty the main differences between Java and Erlang as programming languages.