Prof. O. Nierstrasz
Jorge Ressia, Jan Kurš

Serie 4 - Safety Patterns

Exercise 1

Answer the following questions:

1. Why are immutable classes inherently safe?
2. What is "balking"?
3. When is partial synchronization better than full synchronization?
4. How does containment avoid the need for synchronization?
5. What liveness problems can full synchronization introduce?
6. When is it all right to declare only some methods as synchronized?

Exercise 2

What action trace violates the following safety property?

```
property PS = (a -> (b -> PS|a -> PS)|b -> a -> PS).
```

Exercise 3

Write a safety property LIFTCAPACITY:

```
LIFT[i:0..4] = (enter -> LIFT[i+1]
                |when(i>0) exit  -> LIFT[i-1]
                |when(i==0)exit  -> LIFT[0]
                ).
```

Hint: the action trace should be: enter, enter, enter, enter, ERROR.

Exercise 4

The dinning savages: A tribe of savages eats communal dinners from a large pot that can hold M servings of stewed missionary. When a savage wants to eat, he helps himself from the pot unless it is empty in which case he waits for the pot to be filled. If the pot is empty the cook refills the pot with M servings. The behavior of the savages and the cook are described by:

```
SAVAGE = (getserving -> SAVAGE).
COOK   = (fillpot -> COOK).
```

Model the behavior of the pot as an FSP process.