CS 210

Lab assignment 2
Jens Blanck
25/2/18

To run this lab you will need the LTSA tool. Download ltsa.jar from Blackboard and run that.

Consider the following variation of the game Nim. From a pile of pebbles players may take 1, 4 or 6 pebbles each time it is their turn. The winner is the player that takes the last pebble. The game starts with a pile of n pebbles.

1. The following FSP gives a simple Counter.

   ```
   const N = 10
   range R = 0..N

   Count = Count[N],
   Count[i:R] = (one -> Count[i-1]).
   ```

   Enter the above into LTSA and compile and check that the generated LTS is what you expect.

2. When the above is compiled there is a warning that `Count.0` is defined to be `ERROR`. Adjust the counter by adding a guard, using the `when` keyword, so that the compiler does not issue this warning.

3. Modify the above counter to give a `Pile` process that allows valid moves in the Nim game specified above.

4. Create a `Game` process such that two players `a` and `b` can perform the moves allowed by `Pile`. [Hint: Use prefixes and the `::` operator.]

5. Write a process `Alternate` that controls a game so that player `a` and `b` performs alternate moves. [Here you may find `set` definitions useful as a shorthand for several actions.]

6. Compose `Game` with the `Alternate` process so that the composition correctly specifies the above game.

7. Assume that n is 10. How many states does the resulting LTS have? The LTS can also be minimized, how many states does it have now?

8. Does the LTS have states where both `a` and `b` have valid moves?

9. Identify states where `a` or `b` have a winning strategy, i.e., states where for example `a` can win regardless of what `b` chooses to do. What number of pebbles do these states correspond to?