

Problem Wk.3.1.1: Simulating Cascade

You are given the following state machines:

```
class Delay(sm.SM):
    def __init__(self, v0):
        self.startState = v0
    def getNextValues(self, state, inp):
        # Output is old state
        return (inp, state)

class Increment(sm.SM):
    startState = 0
    def __init__(self, incr):
        self.incr = incr
    def getNextValues(self, state, inp):
        return (state, inp + self.incr)
```

Consider the following cascaded compositions of state machines and fill in the the behavior tables.

```
1. sm1 = Delay(1)
   sm2 = Delay(2)
   c = sm.Cascade(sm1, sm2)
   c.transduce([3,5,7,9])
```

	t=0	t=1	t=2	t=3	t=4
sm1 input	3	5	7	9	
sm1 state	1				
sm1 output					
sm2 input					
sm2 state	2				
sm2 output					

```
2. sm1 = Delay(1)
   sm2 = Increment(3)
   c = sm.Cascade(sm1, sm2)
   c.transduce([3,5,7,9])
```

	t=0	t=1	t=2	t=3	t=4
sm1 input					
sm1 state					
sm1 output					
sm2 input					
sm2 state					
sm2 output					

MIT OpenCourseWare
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.