

## Problem Wk.4.4.3: TransducedSignal

Given a state machine and a signal, we can construct a new signal, which is the sequence of outputs generated by the machine when the original signal is used as an input. Unfortunately, this only defines a partial signal (for  $n \geq 0$ ), because state machines describe how to generate new values of the output given previous values of input and output.

We can define a class `TransducedSignal`, which is a subclass of `Signal`, that takes two arguments at initialization time: `s`, (the input signal) is an instance of `Signal` and `m` (the state machine which will take `s` as input) is an instance of `SM`.

All we need to is define its `sample` method, which returns the value of the transduced signal at step  $n$ . How can we compute this value? By running the state machine forward, starting in its initial state, for  $n$  steps, and seeing what the output is.

Yes, we know this is very inefficient. If that bothers you, extend the class so that it caches values and internal states of the state machine as necessary to avoid unnecessary repeated computation.

---

### Part 1: Implementation

Implement the `TransducedSignal` class.

You should write your code in Idle. In the design lab folder, create a file with

```
import lib601.sm as sm
import lib601.sig as sig
```

at the top. You can then place your definitions and test cases in the file. Evaluate them with Run Module. You may find this procedure useful in testing:

```
def samplesInRange(sig, lo, hi):
    return [sig.sample(i) for i in range(lo,hi)]
```

You may want to use the `transduce` method of the `sm.SM` class, which takes a list of input values,  $x[0], \dots, x[n]$ , and generates a list of output values,  $y[0], \dots, y[n]$ .

```
class TransducedSignal(sig.Signal):
    def __init__(self, s, m):
        self.s = s
        self.m = m

    def sample(self, n):
        if n < 0:
            return 0
        else:
            # Your code here
```

---

## Part 2: Application

Use your implementation to compute the value of my bank account over time if: I deposit \$100 at time 0, time 20, and time 50, and there is 1 percent management fee per time step.

- Start by constructing the input signal.
- Then, construct an `LTISM` representing this bank account (using your implementation of `LTISM` from the previous problem).
- Next, construct a new signal, which is the result of transducing the input signal through the bank account state machine.
- Finally, look at the specified values of the resulting signal.

If you need help remembering the names of methods and classes that we're using, look at the [software documentation](#).

Enter numbers with at least 3 digits beyond the decimal point.

1. Value at time 10:
2. Value at time 20:
3. Value at time 30:
4. Value at time 40:
5. Value at time 50:
6. Value at time 60:

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>.