Topics Since Midterm II

- Interpreters
- Streams
- Iterators and generators
- Logic
- Parallelism
- MapReduce
Interpreters
Project 4; Lab 6a, 6b; Discussion 5b, 6a

• **read** input
• translate the input into an expression (**parse**)
• **evaluate** the parsed expression
• **print** the result

```
def repl():
    while True:
        try:
            src = input('scm> ')
            expression = read_exp(tokenize(src))
            print(calc_eval(expression))
        except SomeErrors:
            ...
```
Streams
Discussion 7a

- lazy RList: they have a first and a rest
- the rest is only evaluated when needed

```python
class Stream(Rlist):
    def __init__(self, first, compute_rest=lambda: Stream.empty):
        self.first = first
        self._compute_rest = compute_rest
        self._rest = None

...
class Stream(Rlist):
    def __init__(self, first, compute_rest=lambda: Stream.empty):
        ...

@property
def rest(self):
    if self._compute_rest:
        self._rest = self._compute_rest()
        self._compute_rest = None
    return self._rest
def make_stream(<some arguments>):
    def compute_rest():
        return make_stream(<some updated arguments>)
    return Stream(first, compute_rest)

def integer_stream(first=1):
    def compute_rest():
        return integer_stream(first+1)
    return Stream(first, compute_rest)

look ma, no arguments!

construct a Stream

returns another Stream
Iterators and Generators
Lab 7a; Discussion 7a

• can represent infinite sequences in finite memory
• represent sequences with functions that compute the next values
• will only calculate values when they are needed

Must have the following methods:
__iter__: returns an iterator object
__next__: checks if there are any values left to compute and raises a StopIteration error if there aren’t; calculates the next value

Generators are special Python iterators. They use yield statements to report values.

But remember, you are allowed to have an Iterator that only has the __iter__ method as long as it returns an object that has a __next__ method.
Logic
Lab 7b; Discussion 8a

• Declarative Programming vs Imperative Programming
• expressions are facts or queries
• a simple fact declare a relation to be true
• a compound fact includes multiple relations

(fact <conclusion>
    <hypothesis 1>
    ...
    <hypothesis n>)

the conclusion is true if and only if all hypotheses are true
Parallelism
Discussion 7b

• multiple programs being run at the same time can yield results that would not happen if the programs were run in serial
• as long as programs don’t modify shared state, running programs in parallel is great!
• if programs do need to modify shared state, then **locks** and **semaphores** are used to indicate when access is permitted
• **race conditions**: when multiple threads concurrently access the same data and mutate it
MapReduce
Lab 8a

- a framework for concurrently processing huge amounts of data
- **map phase**: apply a mapper function to inputs, emitting a set of intermediate key-value pairs
- **reduce phase**: for each intermediate key, apply a reducer function over all the corresponding intermediate values