Eraser: A Dynamic Data Race Detector for Multithreaded Programs
Review

- Introduce tool to help eliminate data races
  - Threads must use mutex locks
  - Finds bugs during dynamic executions
  - Checks shared variables on heap or global vars
  - Locks at “right” granularity

- Assumption?
  - Correct program holds a specific lock when accessing shared data
How?

- Lockset algorithm:
  - Determines association of locks and data
  - Tracks lock set $C(v)$ that *could* be associated with shared variable
  - If lock set becomes empty, sync error reported
Example #1

**Code:**

```plaintext
lock(A)
x = x + 1
unlock(A)

lock(B)
x = x + 1
unlock(B)
```
Example #2

Thread 1:  
lock(A)  
if x > 5:  
    z = x + 1  
else:  
    z = 0  
unlock(A)

Thread 2:  
x = 10
Alternative Methods

- Monitors
  - Object-oriented
  - Methods perform locks on instance variables

- Static Analysis
  - Advantages?
  - Disadvantages?