

# Principles of Garbage Collection



## Manual vs. automatic memory management

**Soundness:** never reclaim memory still in use

**Utility:** collect as much garbage as possible

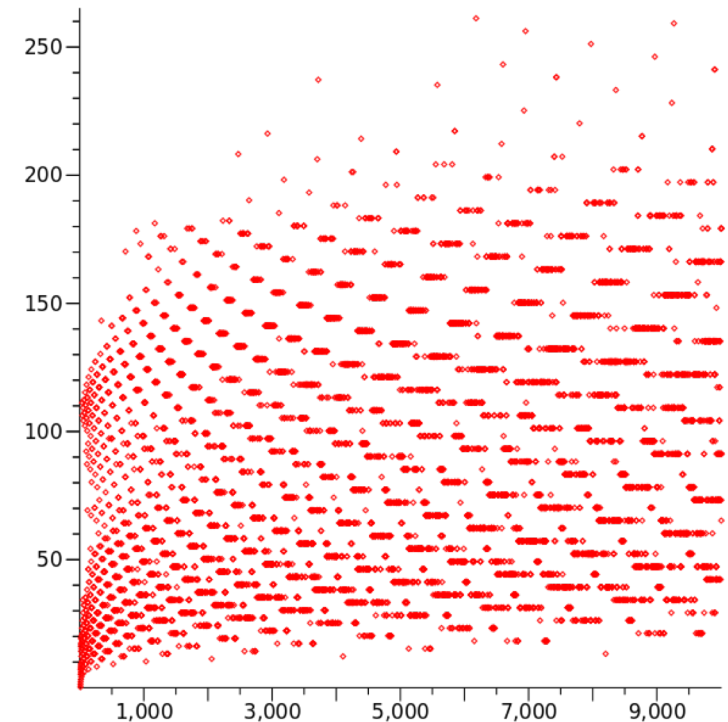
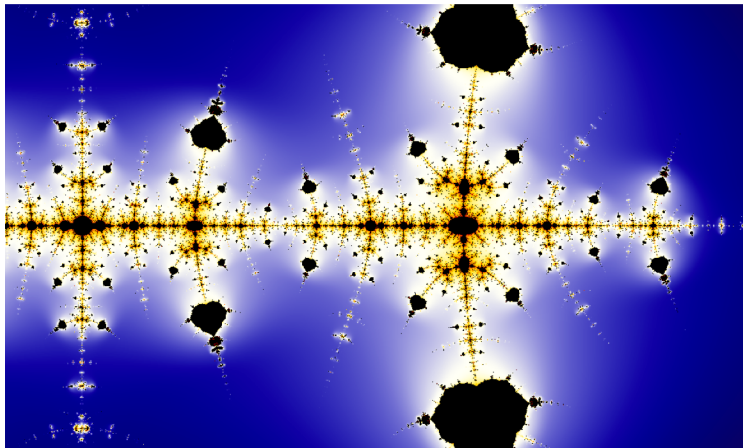
**Efficiency:** run fast enough to not add much run-time overhead

And now as I said concerning faith—faith is not to have a perfect knowledge of things; therefore if ye have faith ye hope for **things which are not seen, which are true.**

Alma 32:21

# The Collatz conjecture

```
def collatz( n ):  
    while n > 1:  
        if mod( n, 2 ) == 0:  
            n = n/2  
        else:  
            n = 3*n + 1
```



# Most garbage collectors:

- Identify reclaimable space
- Collect or reclaim (fast!)

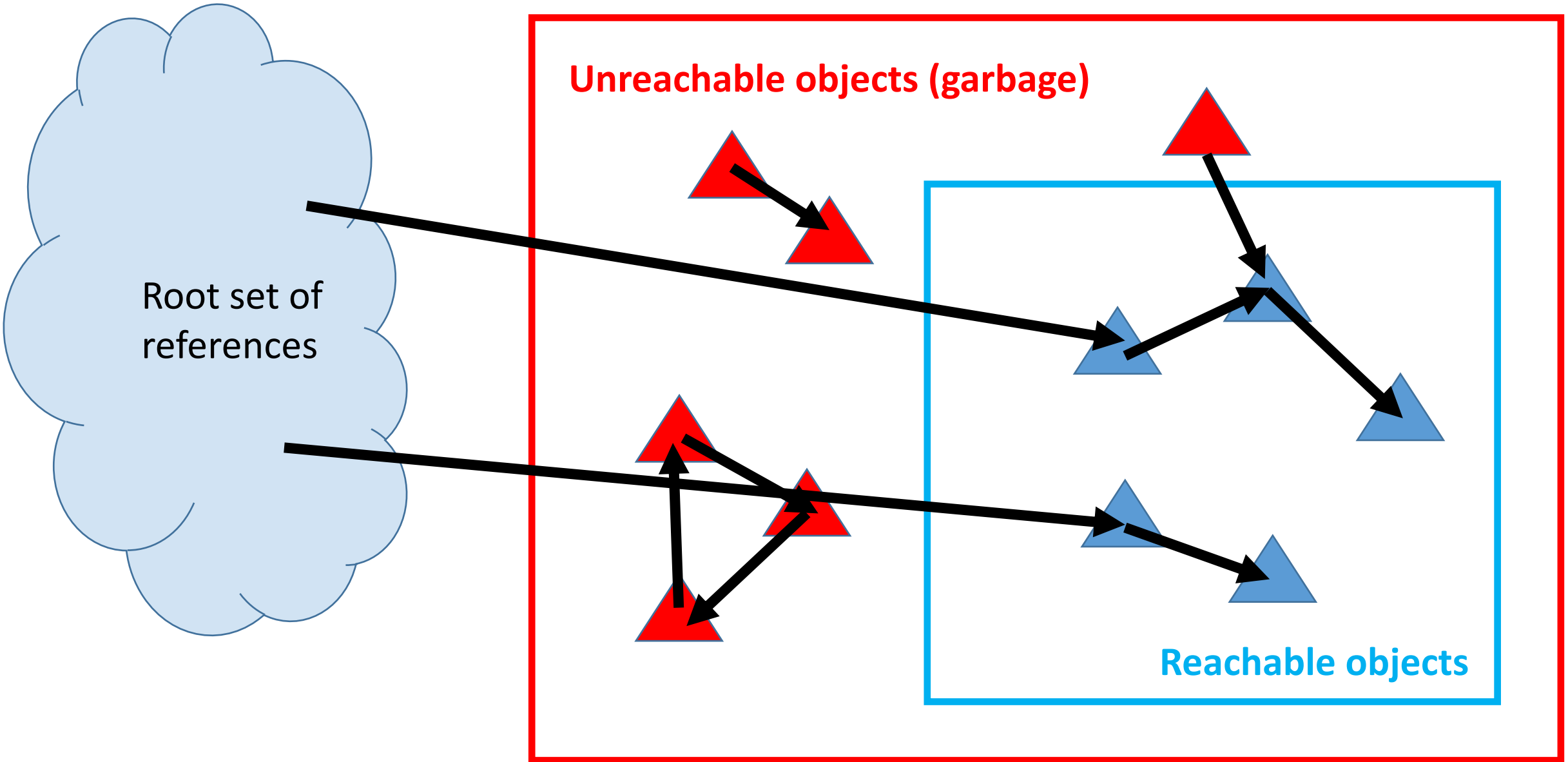
Key term: “root set” – the named entities in a program

# Reachability

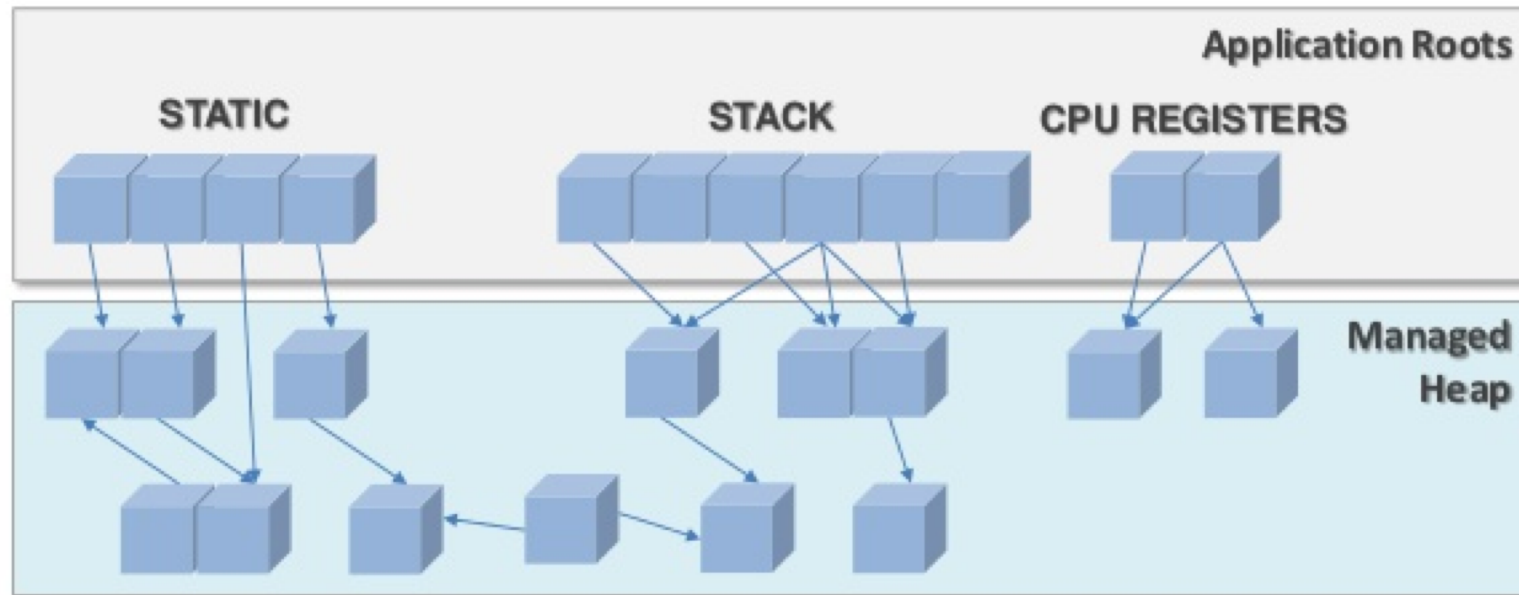
Start at the root set

If object  $o$  is reachable, then all objects that  $o$  references are also reachable

And so on, recursively



# The root set in a C / .Net program



What is the equivalent in our interpreter?