



Garbage Collection & Reference Counting

Clemens Ruck & Alex Egger

Summer Term 2017

Overview

Methods of Memory Management

- 1 Shortcomings of Manual Memory Management
- 2 Garbage Collection

Memory Management in Rust

- 1 Rust's approach
- 2 Stack Allocation
- 3 Heap Allocation with `Box<T>`
- 4 Comparison of Stack and Heap allocations

Reference Counting

- 1 Concept
- 2 Problems
- 3 `Box<T>` in Rust

Common shortcomings of manual memory management

Garbage Collection

Example - Mark & Sweep

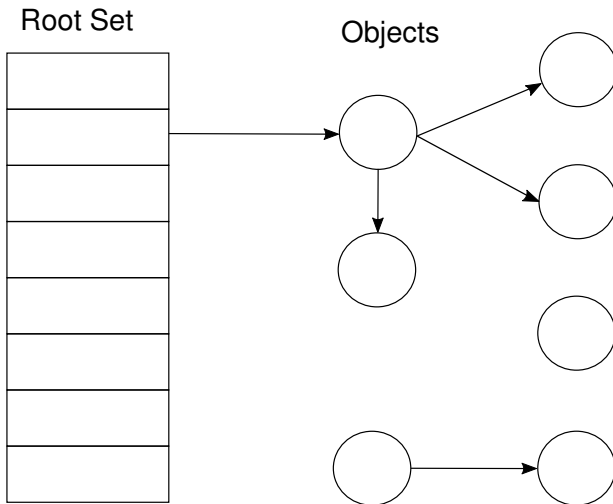


Figure: A graph-representation of alive objects.

Example - Mark & Sweep

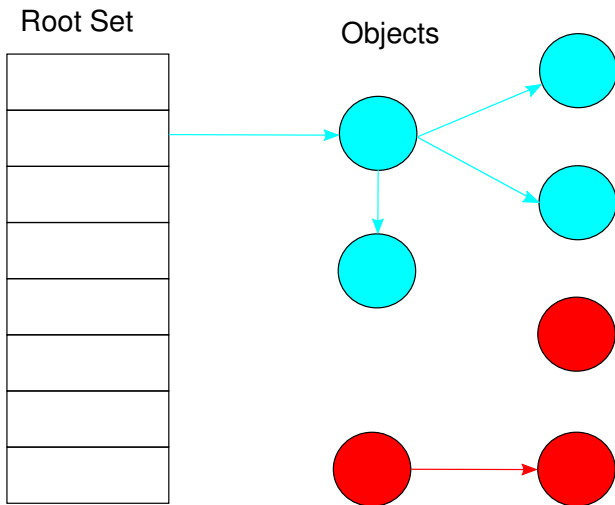


Figure: The 'Mark' stage of the algorithm.

Example - Mark & Sweep

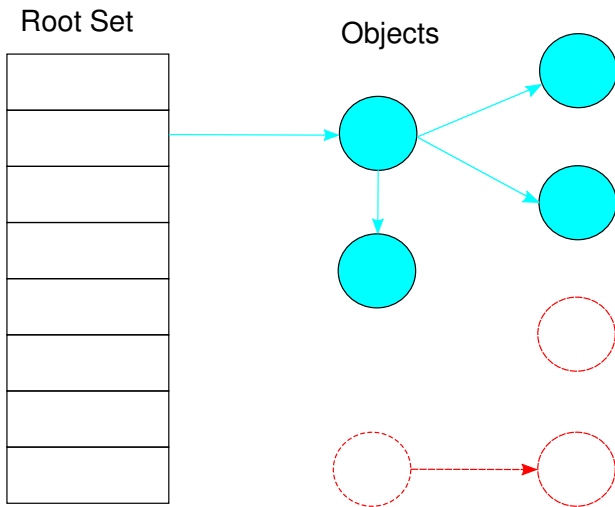


Figure: The 'Sweep' stage of the algorithm.

Memory management in Rust

Stack Allocation

Local Variables
Return Address
Parameters
⋮

Workings of the Stack

Advantages & Limitations

Heap Allocation

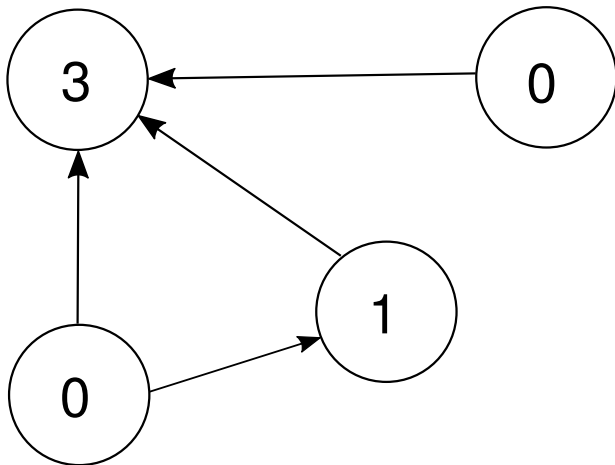
Workings of the Heap

Box<T> in Rust

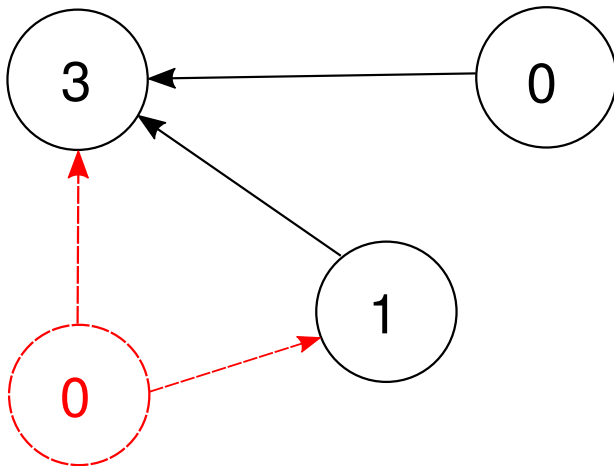
Use cases

Comparison: Heap vs. Stack

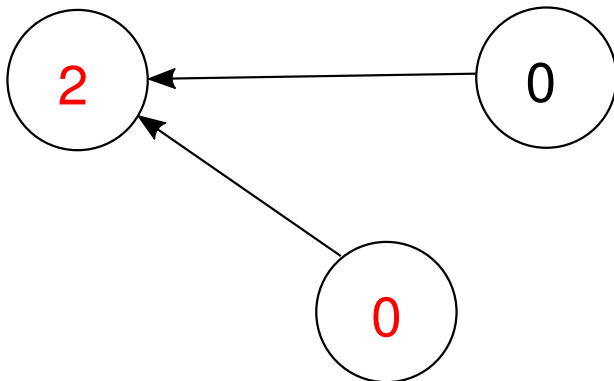
Reference Counting - Example



Reference Counting - Example



Reference Counting - Example



Limitations

Rc<T> (and Weak<T>) in Rust

Example



Limitations of $R_c\langle T \rangle$

Multi-threaded Reference Counting with Arc<T>

Example



The 'unsafe' keyword

Use cases