



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

Memory management in Rust

Stack Allocation

Workings of the Stack

Advantages & Limitations

Heap Allocation

Workings of the Heap

Box<T> in Rust

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Use cases

Comparison: Heap vs. Stack

Reference Counting

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

- **Use cases** are a way of describing the requirements of a system in terms of the goals that users want to achieve when using the system.
- They are a way of capturing the functional requirements of a system in a structured and organized manner.
- Use cases are typically represented by a list of numbered items, each describing a specific goal or function that the system must support.
- Each use case is typically written in a structured format, including a title, a description of the goal, and a list of steps or actions that the user must perform to achieve the goal.
- Use cases are often used to communicate the requirements of a system to stakeholders, including developers, testers, and users.
- They provide a clear and concise way of describing the functional requirements of a system, making it easier for everyone involved in the development process to understand what the system is supposed to do.
- Use cases are also used to guide the development of a system, as they provide a clear set of requirements that the system must meet.
- By using use cases, developers can ensure that the system they are building meets the needs of the users and that it is easy to use.
- Use cases are a key part of the requirements engineering process, and they are essential for the successful development of any system.