

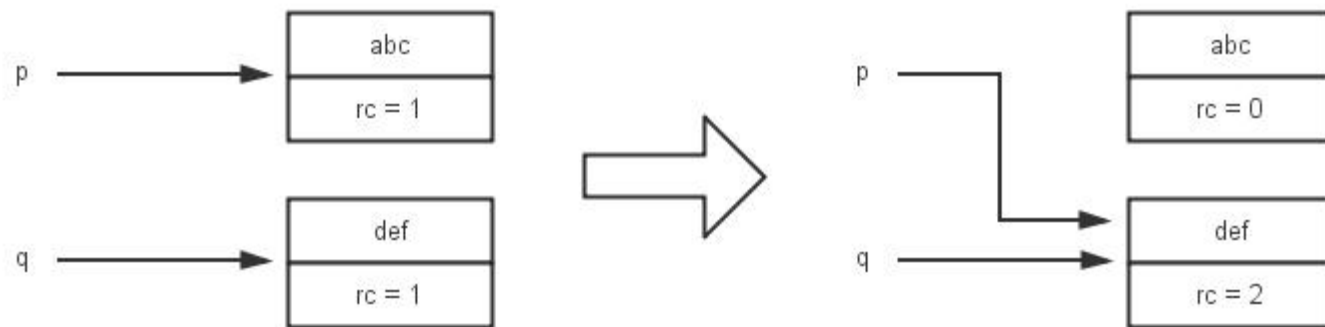
# 内存垃圾收集器

- 内存垃圾回收器([Garbage Collector](#), 简称GC), 是一种自动内存管理机制
- 第一次出现于1958年, 由John McCarthy首先实现, 使之作为Lisp实现的一部分
- 包括Java、Python、Perl、Modula-3、Prolog、ML和Smalltalk等语言的运行时系统都含有GC

# GC的典型技术

- **无需**标记和“stop-the-world”回收
  - 引用计数
- **需要**标记和“stop-the-world”回收
  - 标记和清除
  - 标记紧缩
  - 拷贝回收
  - 分代回收

# 引用计数

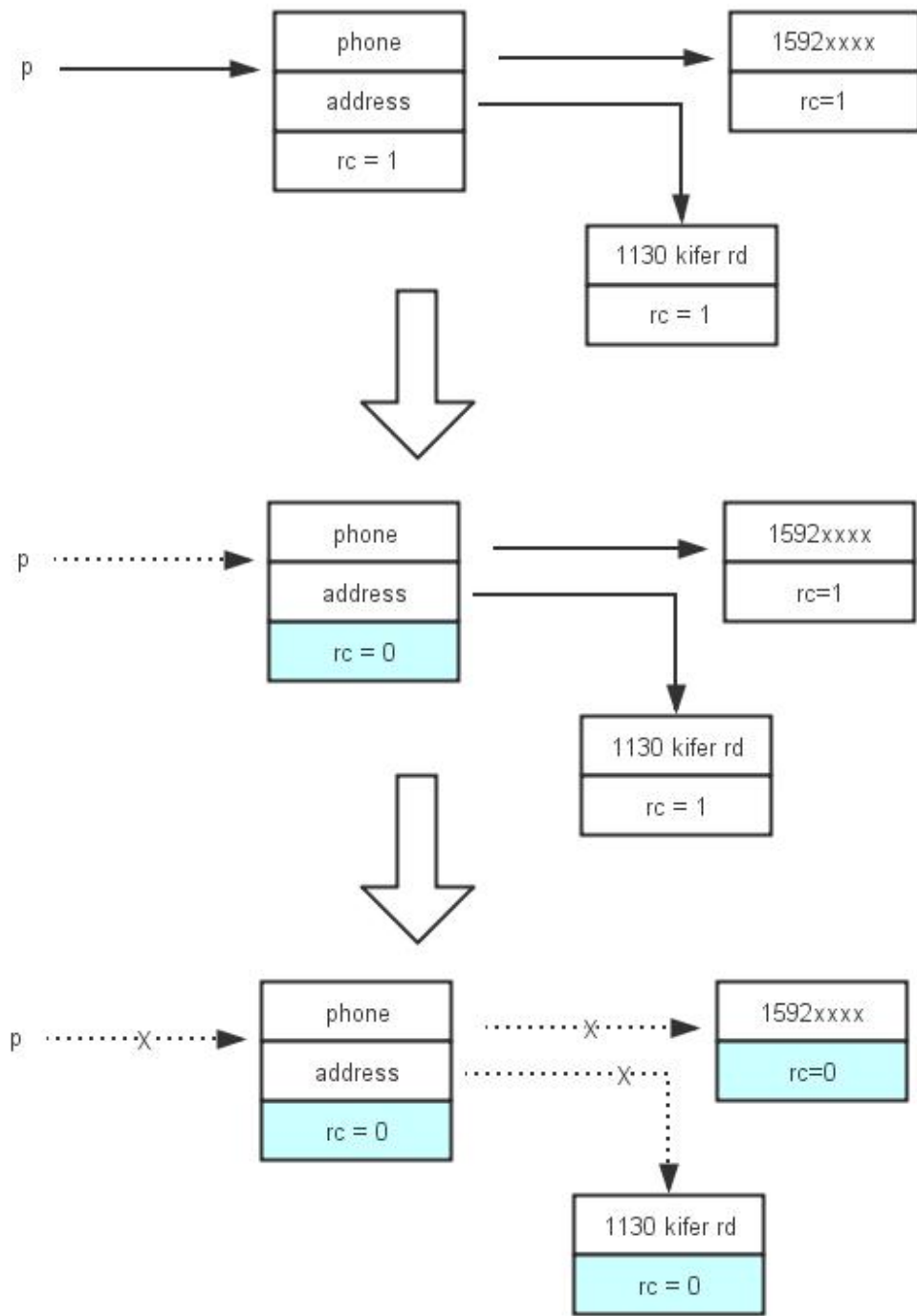


- p指针指向q的对象时，p原指向的对象计数减1，q的计数加1
- 计数减为0的对象，系统随时可以回收

# 引用计数的局限和问题

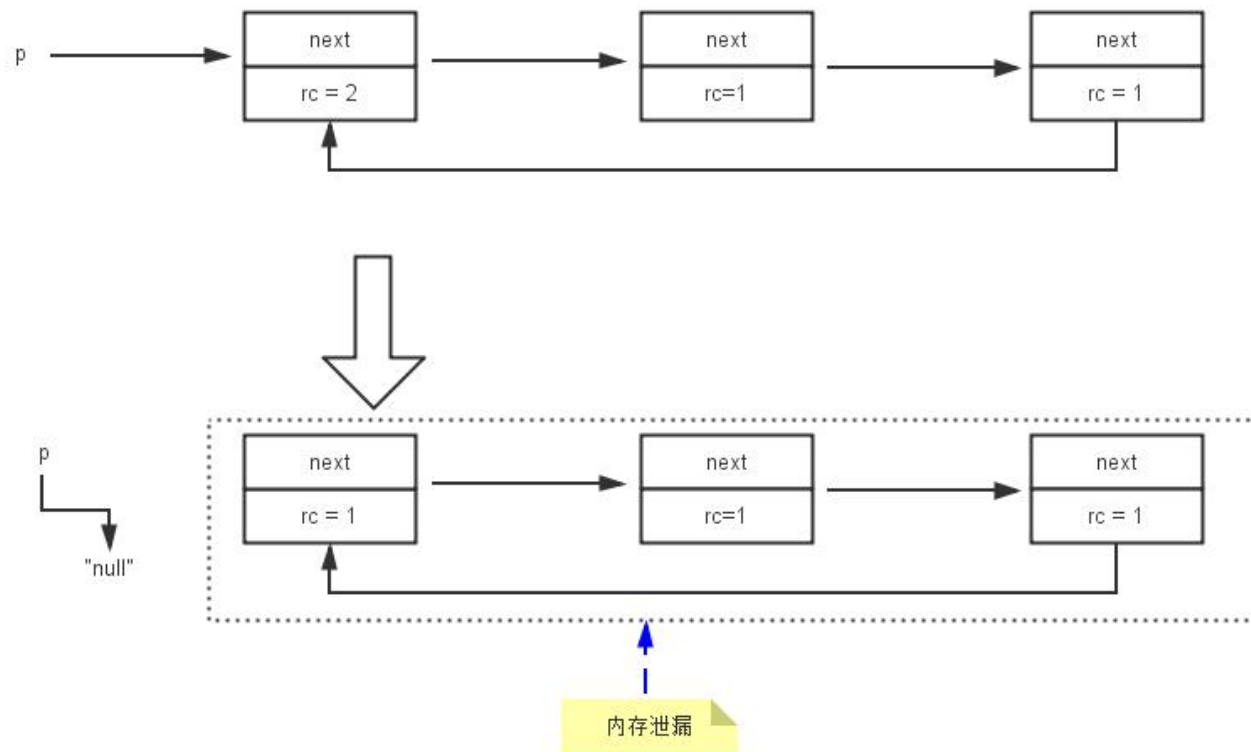
- 效率问题

- 每次改变对象指针均需要递归改写计数（出现引用为0的情况）

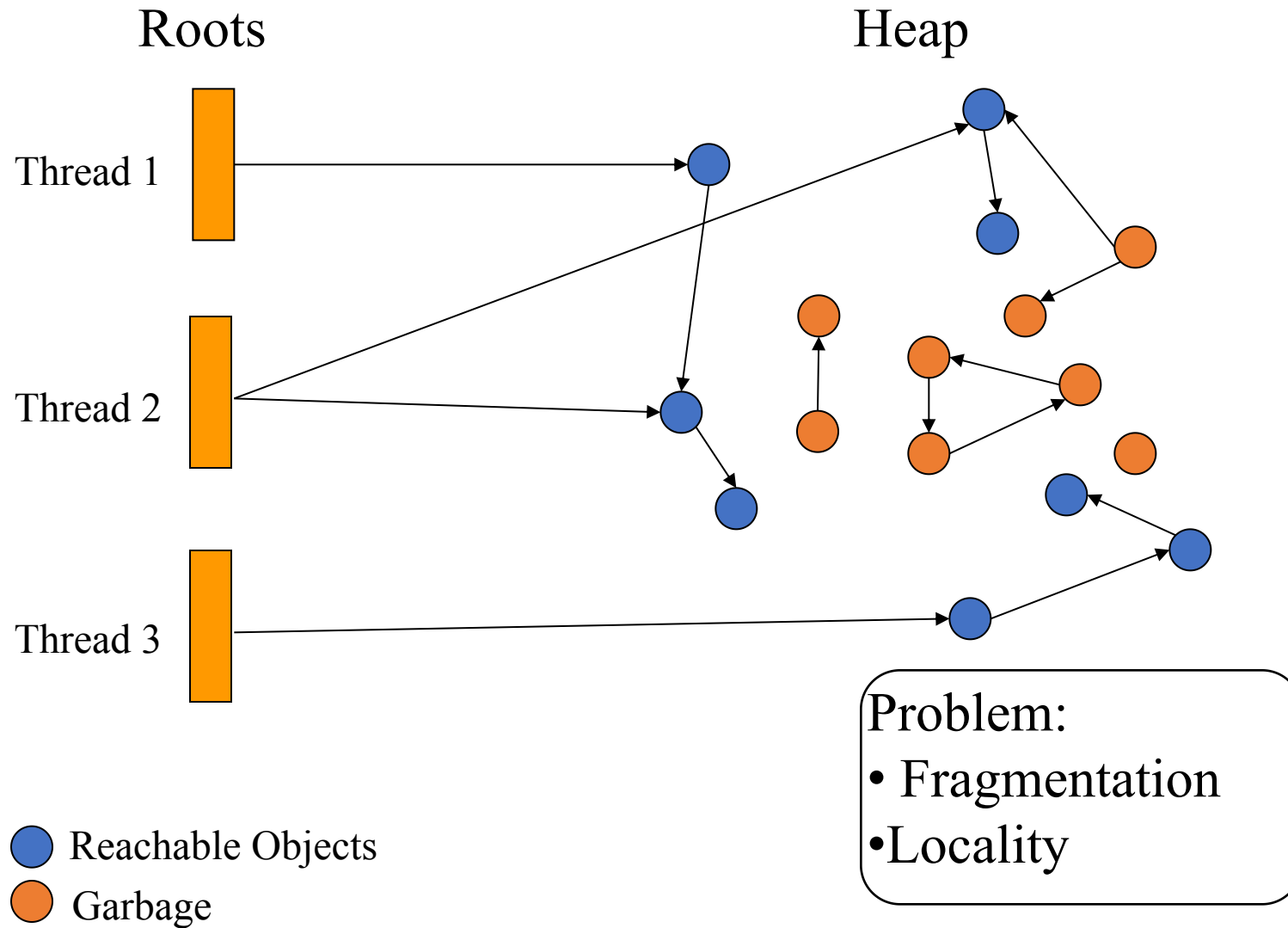


# 引用环问题

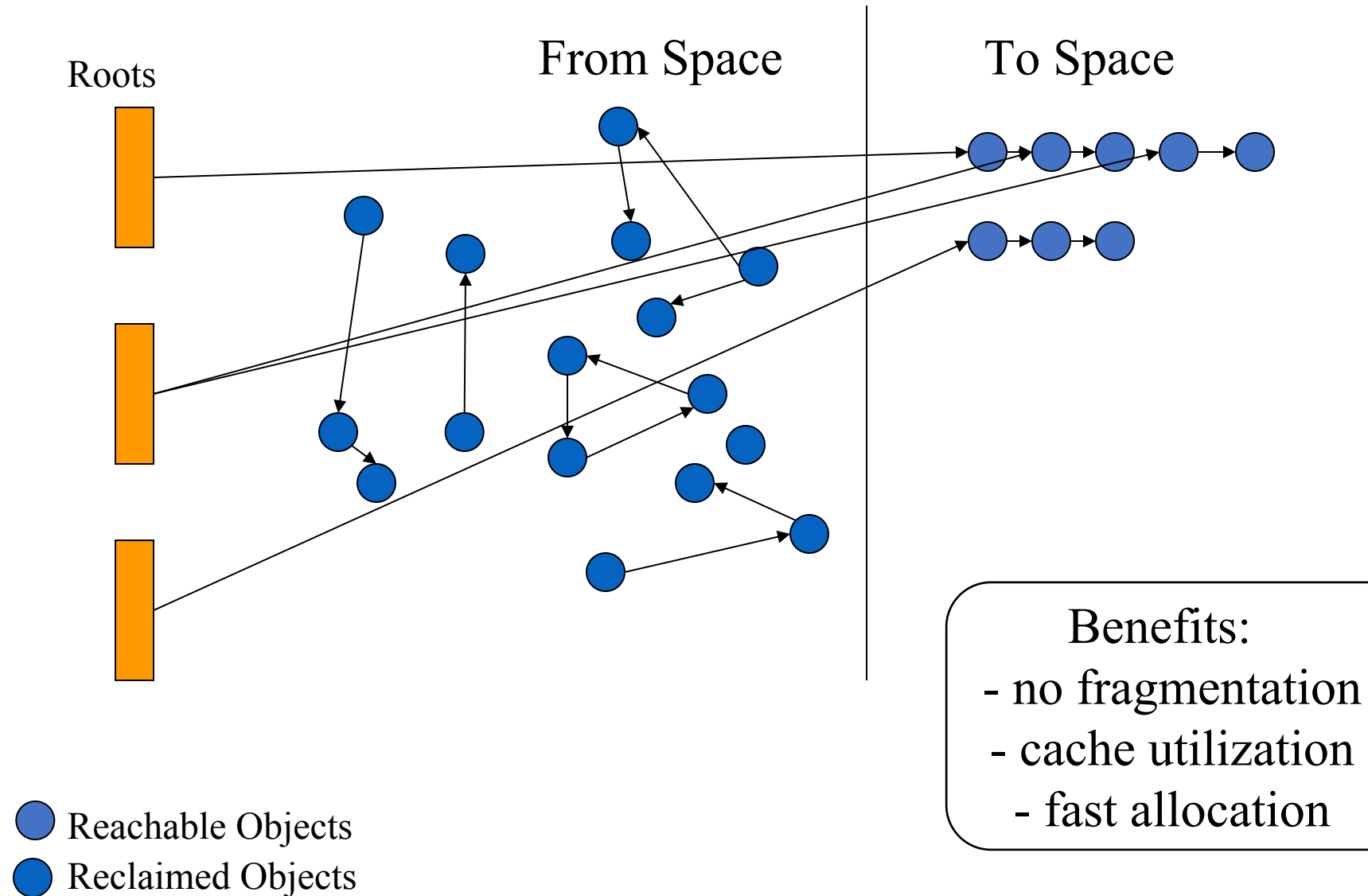
- 导致内存泄漏
- 通过“弱引用”解决
  - SWIFT
  - 华为方舟



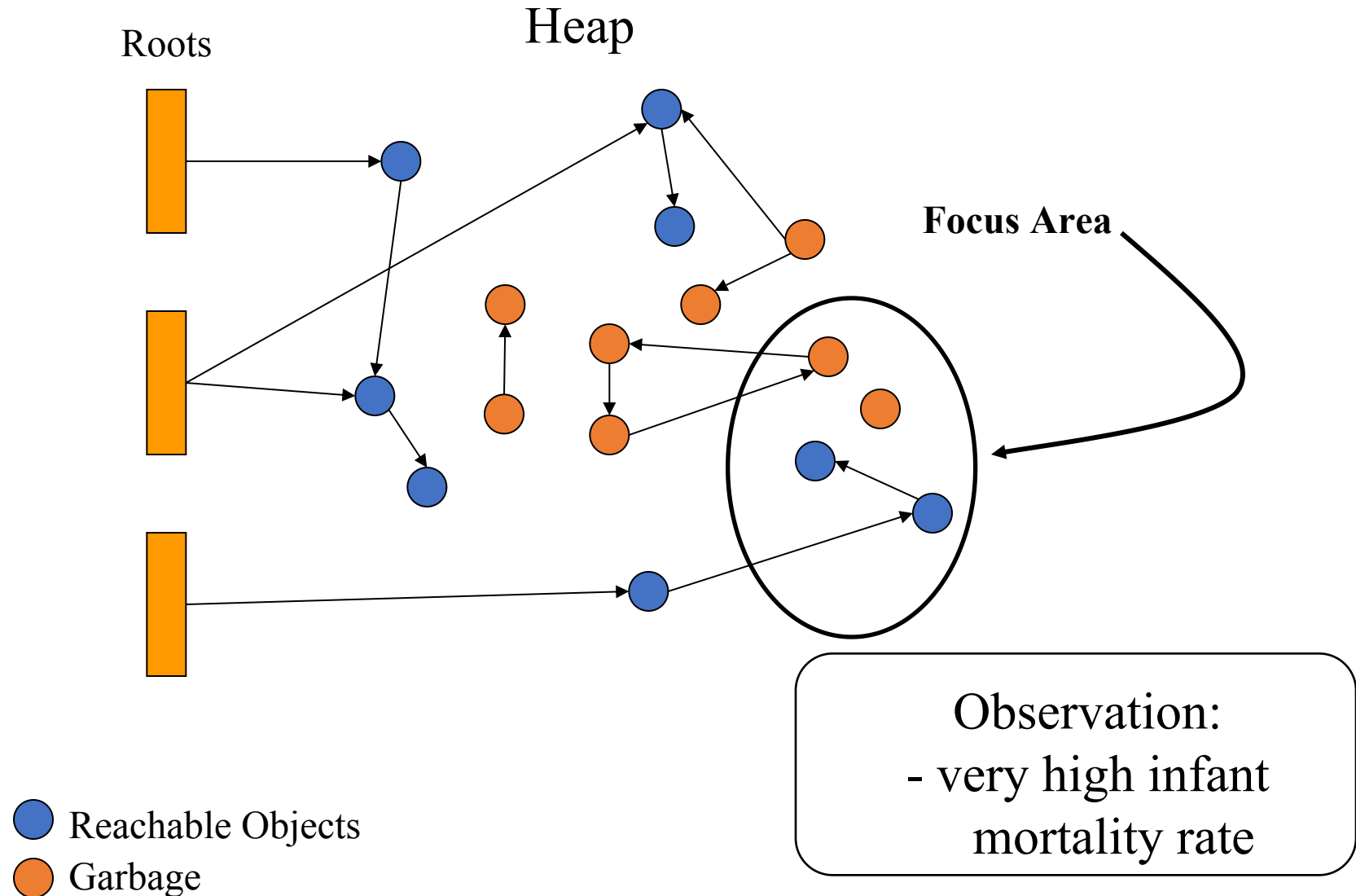
# How Mark/Sweep Works



# How Copying Works



# How Generational Collection works





# Generational Continued

