A Verified Garbage Collector for Gallina

Shengyi Wang, Anshuman Mohan, Aquinas Hobor

National University of Singapore

APLAS NIER
November 15, 2019

Broad Problem

Verify graph-manipulating programs
written in executable C
with machine-checked correctness proofs

Broad Problem

Verify graph-manipulating programs !!
written in executable C
with machine-checked correctness proofs

Broad Problem

Verify graph-manipulating programs !!
written in executable C !!
with machine-checked correctness proofs

Broad Problem

Verify graph-manipulating programs !!
written in executable C !!
with machine-checked correctness proofs !!

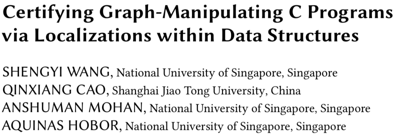
Broad Problem

Verify graph-manipulating programs !!
written in executable C !!
with machine-checked correctness proofs !!

Ubiquitous in critical areas!

Broad Solution

Verified
Software
Toolchain



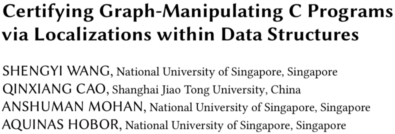
Certifying Graph-Manipulating C Programs via Localizations within Data Structures

SHENGYI WANG, National University of Singapore, Singapore
QINXIANG CAO, Shanghai Jiao Tong University, China
ANSHUMAN MOHAN, National University of Singapore, Singapore
AQUINAS HOBOR, National University of Singapore, Singapore

VST + CompCert + 40000 LOC library

Broad Solution

Verified
Software
Toolchain



Certifying Graph-Manipulating C Programs via Localizations within Data Structures

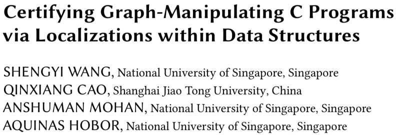
SHENGYI WANG, National University of Singapore, Singapore
QINXIANG CAO, Shanghai Jiao Tong University, China
ANSHUMAN MOHAN, National University of Singapore, Singapore
AQUINAS HOBOR, National University of Singapore, Singapore

VST + CompCert + 40000 LOC library

Powerful enough to verify real code
against strong specifications
expressed with mathematical graphs

Broad Solution

Verified
Software
Toolchain



Certifying Graph-Manipulating C Programs via Localizations within Data Structures

SHENGYI WANG, National University of Singapore, Singapore
QINXIANG CAO, Shanghai Jiao Tong University, China
ANSHUMAN MOHAN, National University of Singapore, Singapore
AQUINAS HOBOR, National University of Singapore, Singapore

VST + CompCert + 40000 LOC library

Powerful enough to verify real code
against strong specifications
expressed with mathematical graphs

[Wang *et. al.*, PACMPL OOPSLA 2019]

This Talk

Gallina ⇝ CompCert C ⇝ Assembly

This Talk

Certi Coq

Gallina ⇝ CompCert C ⇝ Assembly

This Talk

Gallina ⇝ CompCert C ⇝ Assembly

Gallina assumes infinite memory
but CompCert C has a finite heap

Solution: garbage collect the CompCert C code

This Talk

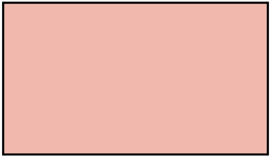
Gallina ⇝ CompCert C ⇝ Assembly

Gallina assumes infinite memory
but CompCert C has a finite heap

Solution: garbage collect the CompCert C code
New problem: verify the garbage collector

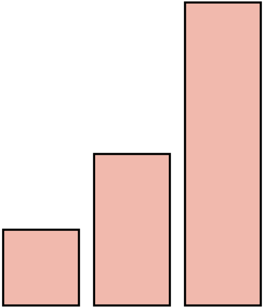
Garbage Collection: an Introduction

GC has jurisdiction over the heap



Garbage Collection: an Introduction

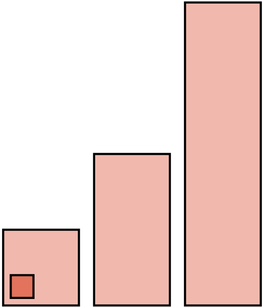
GC has jurisdiction over the heap



Garbage Collection: an Introduction

GC has jurisdiction over the heap

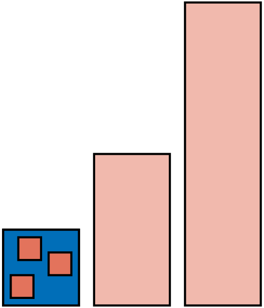
Mutator `mallocs` in special subheap



Garbage Collection: an Introduction

GC has jurisdiction over the heap

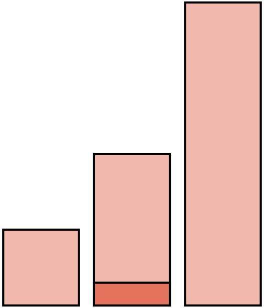
- Mutator `mallocs` in special subheap
  - If subheap is full



Garbage Collection: an Introduction

GC has jurisdiction over the heap

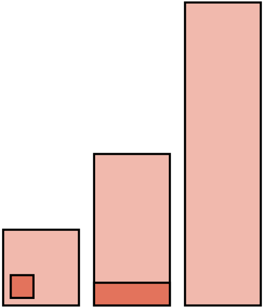
- Mutator `mallocs` in special subheap
  - If subheap is full call GC



Garbage Collection: an Introduction

GC has jurisdiction over the heap

- Mutator `mallocs` in special subheap
  - If subheap is full call GC and try again

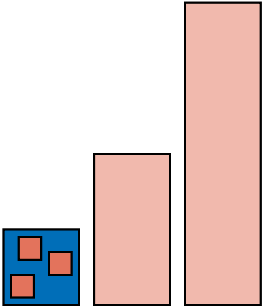


Intuitive Specification

Primum non nocere: first, do no harm

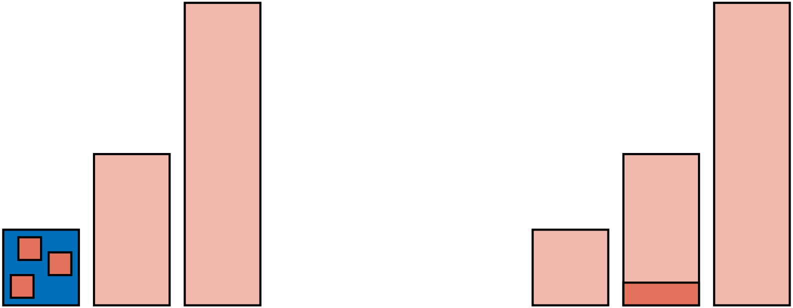
Intuitive Specification

Primum non nocere: first, do no harm



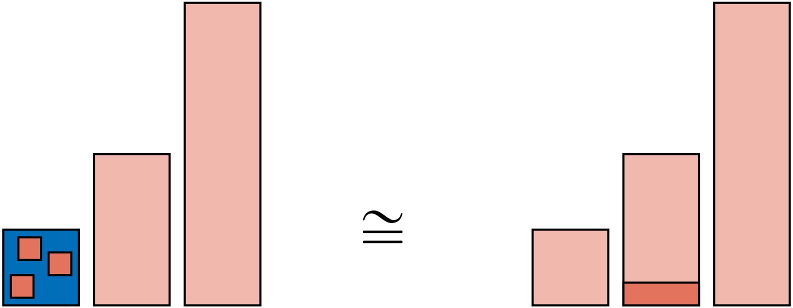
Intuitive Specification

Primum non nocere: first, do no harm



Intuitive Specification

Primum non nocere: first, do no harm



Our Garbage Collector

- 12 generations, doubling in size
- Functional mutator: no back pointers

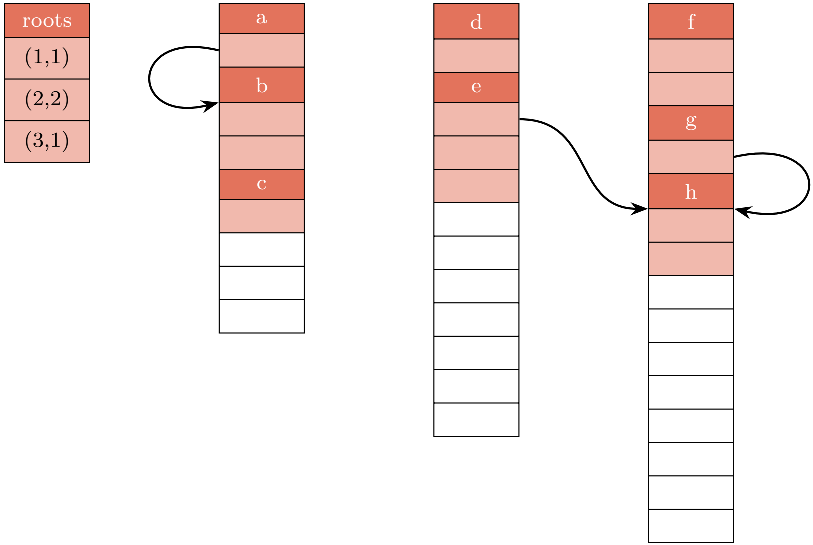
Our Garbage Collector

- 12 generations, doubling in size
- Functional mutator: no back pointers
- Cheney's mark-and-copy collects gen to next
- Potentially triggers cascade of pairwise collections

Our Garbage Collector

- 12 generations, doubling in size
- Functional mutator: no back pointers
- Cheney's mark-and-copy collects gen to next
- Potentially triggers cascade of pairwise collections
- Two key functions:
  `forward` copies individual objects
  `do_scan` repairs copied objects

Overview of `forward` and `do_scan`



roots

(1,1)

(2,2)

(3,1)

a

b

c

d

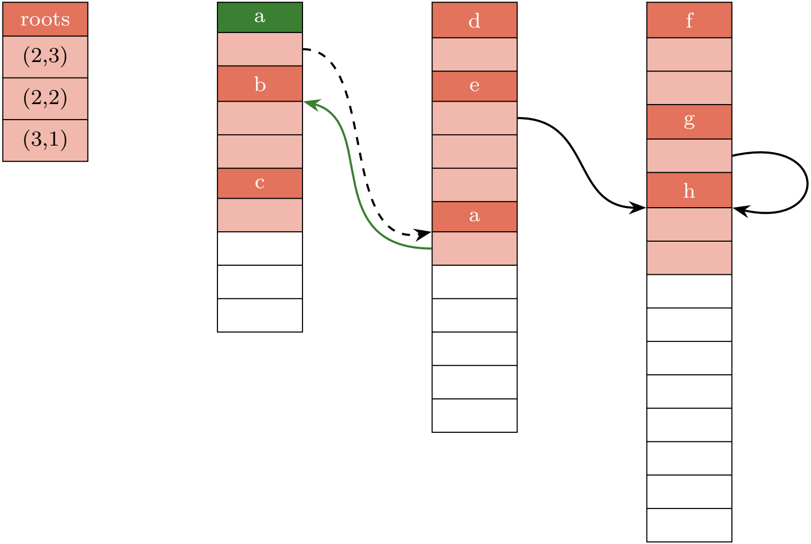
e

f

g

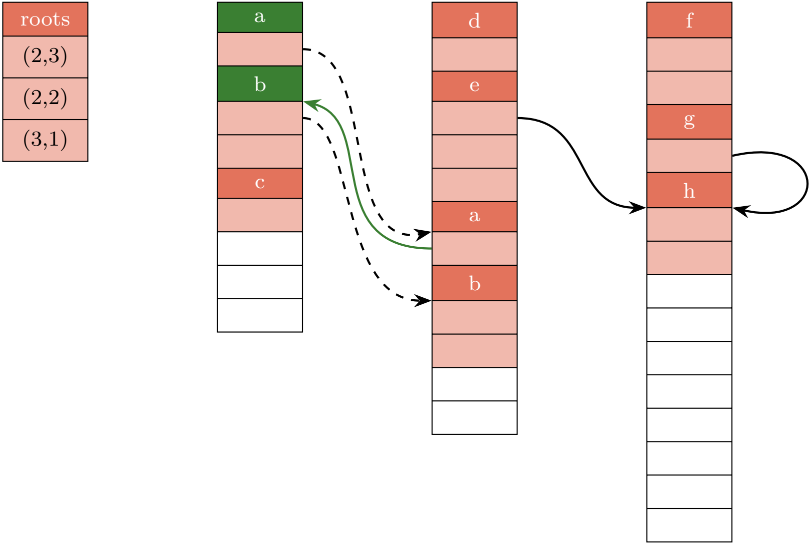
h

Overview of `forward` and `do_scan`



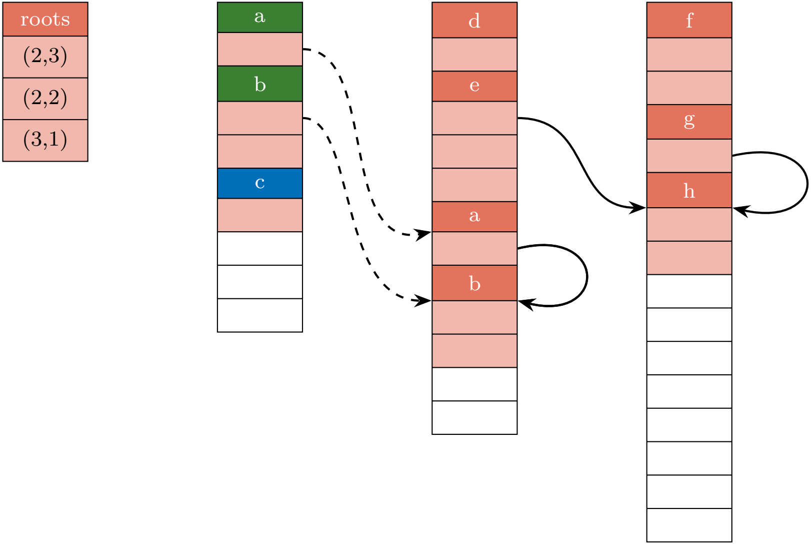
`forward` ✓

Overview of `forward` and `do_scan`



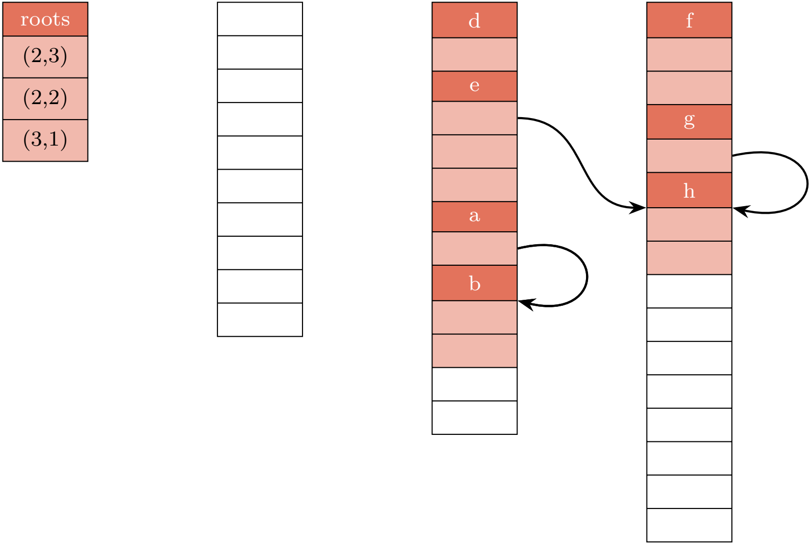
`forward` ✓

Overview of `forward` and `do_scan`



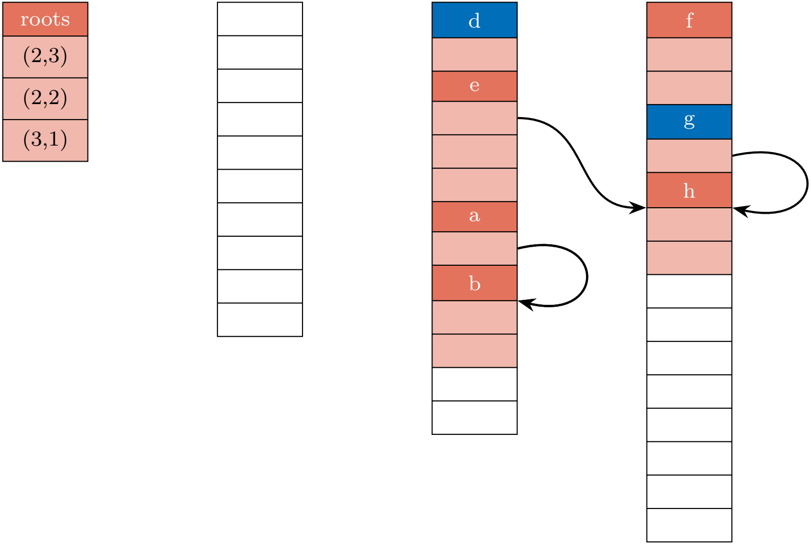
`forward` ✓ `do_scan` ✓

Overview of `forward` and `do_scan`



`forward` ✓ `do_scan` ✓ `do_gen` ✓

Overview of `forward` and `do_scan`



roots

(2,3)

(2,2)

(3,1)

d

e

a

b

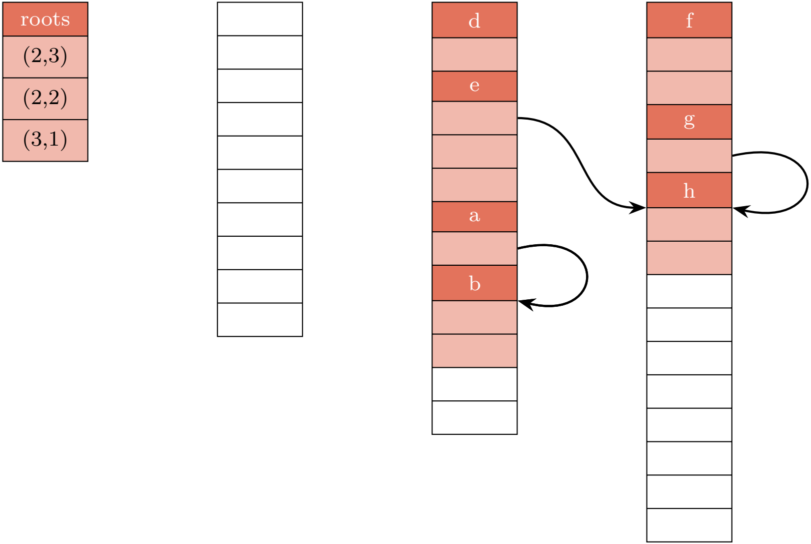
f

g

h

`forward` ✓ `do_scan` ✓ `do_gen` ✓

Overview of `forward` and `do_scan`



`forward` ✓ `do_scan` ✓ `do_gen` ✓