NATURAL PROOFS FOR STRUCTURE, DATA, AND SEPARATION

XIAOKANG QIU PRANAV GARG
ANDREI STEFANESCU P. MADHUSUDAN

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

NSF SITE VISIT, PHILADELPHIA, PA
LOGICAL REASONING FOR HEAP-MANIPULATING PROGRAMS

Expressive Logics:
- Separation logics
- HOL
- Matching logic, etc.

Decidable Logics:
- Strand
- LISBQ
- CSL, etc.

Expressive Logics:
- Expressive
- Decidable

Automatic

Expressiveness

Natural Proofs

Keep expressiveness

Give up decidability
Sound but incomplete
Preserve automaticity
NATURAL PROOFS: IN A NUTSHELL

- Handle a logic that is very **expressive**
  (inevitably undecidable)

- Retain automaticity at the same level as decidable logics

- Identify a class of **natural proofs** $N$ such that
  - $N$ includes natural proof tactics used in human proofs
  - Many correct programs can be proved using a proof in class $N$
  - The class $N$ is effectively searchable
    (checking if there is a proof in $N$ is efficiently decidable)
NATURAL PROOFS FOR PROGRAM VERIFICATION

**DRYAD**
(a dialect of Separation Logic)

Program

Program + **DRYAD** annotation (with recursive definitions)

Translate

Program + **FOL** annotation (with recursive definitions)

Encode

Verification Condition

Don’t know

If verification fails:

- SMT Solver (unsatisfiable)

If verification succeeds:

- SMT Solver (satisfiable)

**Encode**

Formula in a dec. logic over sets:
*Does a natural proof exist?*

User-provided axioms

Tactic 1:

- unfold recursive defs

Tactic 2:

- formula abstraction

N
106 Dryad-annotated programs

- Inductive data structures (e.g., red black trees, binomial heaps): standard operations (recursive & iterative)
- Glib library: operating singly/doubly-linked lists
- OpenBSD library: operating queues
- Linux kernel: VMA operations for virtual memory management
- ExpressOS: operations memory isolation
- full-functional, partial correctness (e.g., data structure invariants, expected set-of-keys, …)

All the VCs were automatically proved by our tool!

- 69 verified within 1 sec
- 96 verified within 10 sec
- 104 verified within 100 sec

“RBT-delete_iter” spent 225 secs, “binomial-heap-merge_rec” spent 153 secs

Most efficient terminating automatic mechanism that can prove such a wide variety of data-structure manipulating programs full-functionally correct

In the future: Natural proofs for synthesizing heap manipulations?
THANK YOU!