Sound, precise, and fast abstract interpretation with tristate numbers

HARISHANKAR VISHWANATHAN, *Rutgers University, USA*
MATAN SHACHNAI, *Rutgers University, USA*
SRINIVAS NARAYANA, *Rutgers University, USA*
SANTOSH NAGARAKATTE, *Rutgers University, USA*

CGO '22: Proceedings of the 20th IEEE/ACM International Symposium on Code Generation and Optimization, April 22, Pages 254-265, https://doi.org/10.5281/zenodo.5703630

Presented by Maxim Petrov. May 16, 2023

Meta Data & Stats

Conference:	CGO
Track:	Program Analysis and Optimization
Year:	2022
Number of Authors:	4
Citations:	0
Pages (PDF):	20
Figures:	5
References:	67
Formals:	3 definitions, 28 lemmas & theorems

What is the Study About?

Formal specification of *tnum* (tristate numbers) abstraction domain for user code static analysis verification based on *BPF static analyzer*. Optimality and soundness of the abstract arithmetic operators for *tnum* domain. Novel algorithm for multiplication of tnums.

Keywords & Terms

Berkeley Packet Filter (BPF) - virtual machine for packet-filtering in Linux kernel (https://lwn.net/Articles/599755/; https://www.tcpdump.org/papers/bpf-usenix93.pdf; https://www.kernel.org/doc/Documentation/networking/filter.txt)

Extended Berkeley Packet Filter (eBPF) - universal in-kernel virtual machine, that has hooks all over the kernel (https://lwn.net/Articles/740157/;)

Abstract Interpretation - partial execution of a computer program which gains information about its semantics (e.g., control-flow, data-flow) without performing all the calculations. (https://en.wikipedia.org/wiki/Abstract_interpretation;)

Tristate Numbers (*tnums*) - n-trits numbers, consisting of a ***trits*** with possible values `{1,0,μ}`, where `μ` denotes undefined bit (0 or 1).

Soundness

Precision

Bounded Verification

Stated Problem(s) & RQ

- Linux kernel provides no formal reasoning or proofs of soundness or precision of its bit-wise & arithmetic algorithms on *tnums*.
- Performance of known proven arithmetics algorithms is lower than kernel ones.

Contributions of the Study

- Provides the first proof of soundness and optimality of the kernel's algorithms for addition and subtraction.
- Novel multiplication algorithms which is provably sound.
- Contribution into Linux kernel.

Paper Structure

I. Introduction

II. Background

A. Primer on Abstract Interpretation

- *Abstraction and concretezation functions. Abstract operators. Galois connection. Optimality.*

B. The Tnum Abstract Domain

- *Abstract and concrete domains. Implementation of tnums in the Linux kernel. Galois connection. Abstract operators on tnums. Challenges.*

III. Soundness And Optimality Of Abstract Arithmetic Over Tnums

A. Automatic Bounded Verification Of Kernel Tnum Arithmetic

- *Soundness of 2-ary operators. Membership predicate. Quantifying over well-formed tnums. Putting it all together. Example: encoding abstract tnum addition. Observation from bounded verification.*

B. Soundness and Optimality Of Tnum Abstract Addition

- *An example. Full adder equation. Key proof technique.*

C. Sound and Efficient Tnum Abstract Multiplication

- *Our algorithm `our_mul` through an example.*

IV. Experimental Evaluation

- *Prior algorithms from abstract multiplication.*

A. Evaluation Of Precision Of `our_mul`

B. Performance Evaluation Of `our_mul`

V. Related Work

- *BPF safety. Abstract interpretation. Safety of static analyzers.*

VI. Conclusion

Acknowledgments

References

VII. Supplementary Materials

A. Proofs for Auxiliary Lemmas for Tnum Addition

B. Proof of our new algorithm for Tnum multiplication

Feedback

Positives

- widely used, practical topic
- formal approach with lots of formal definitions and proofs
- supplied practical results
- lots of references to related works

To be improved

- complex formal theory
- hardly applicable to general code static analysis domain