Duck, Gregory J.; **Jaffar, Joxan**; **Yap, Roland H. C.**
Shape neutral analysis of graph-based data-structures. (English) Zbl 1451.68174
Theory Pract. Log. Program. 18, No. 3-4, 470-483 (2018).

Summary: Malformed data-structures can lead to runtime errors such as arbitrary memory access or corruption. Despite this, reasoning over data-structure properties for low-level heap manipulating programs remains challenging. In this paper we present a constraint-based program analysis that checks data-structure integrity, w.r.t. given target data-structure properties, as the heap is manipulated by the program. Our approach is to automatically generate a solver for properties using the type definitions from the target program. The generated solver is implemented using a Constraint Handling Rules (CHR) extension of built-in heap, integer and equality solvers. A key property of our program analysis is that the target data-structure properties are *shape neutral*, i.e., the analysis does not check for properties relating to a given data-structure graph *shape*, such as doubly-linked-lists versus trees. Nevertheless, the analysis can detect errors in a wide range of data-structure manipulating programs, including those that use lists, trees, DAGs, graphs, etc. We present an implementation that uses the Satisfiability Modulo Constraint Handling Rules (SMCHR) system. Experimental results show that our approach works well for real-world C programs.

MSC:

68Q60 Specification and verification (program logics, model checking, etc.)
68N17 Logic programming
68P05 Data structures

Keywords:

constraint handling rules; satisfiability modulo constraint handling rules; satisfiability modulo theories; program analysis; data structures; memory errors

Software:

LLBMC; Predator; SLAyer; Smallfoot; SMCHR; VeriFast

Full Text: DOI

References:

[1] Berdine, J.; Calcagno, C.; Cook, B.; Distefano, D.; O'Hearn, P.; Wies, T.; Yang, H., Computer Aided Verification, Shape Analysis for Composite Data Structures, (2007), Springer

[2] Berdine, J.; Calcagno, C.; O'Hearn, P., Formal Methods for Components and Objects, SmallFoot: Modular Automatic Assertion Checking with Separation Logic, (2005), Springer

[3] Berdine, J.; Cook, B.; Ishtiaq, S., Computer Aided Verification, SLAyer: Memory Safety for Systems-level Code, (2011), Springer

[4] Boehm, H.; Weiser, M., Garbage Collection in an Uncooperative Environment, Software Practice and Experience, 18, 9, (1988)

[5] Distefano, D.; O'Hearn, P.; Yang, H., Tools and Algorithms for the Construction and Analysis of Systems, A Local Shape Analysis Based on Separation Logic, (2006), Springer

[6] Duck, G., SMCHR: Satisfiability Modulo Constraint Handling Rules, Theory and Practice of Logic Programming, 12, 4-5, 601-618, (2012) · Zbl 1260.68059

[7] Duck, G., International Joint Conference on Artificial Intelligence, Satisfiability Modulo Constraint Handling Rules (Extended Abstract), (2013), AAAI

[8] Duck, G.; Jaffar, J.; Koh, N., Constraint Programming, Constraint-based Program Reasoning with Heaps and Separation, (2013), Springer

[9] Duck, G.; Yap, R., Compiler Construction, Heap Bounds Protection with Low Fat Pointers, (2016), ACM

[10] Duck, G.; Yap, R.; Cavallaro, L., Network and Distributed System Security Symposium, Stack Bounds Protection with Low Fat Pointers, (2017), The Internet Society

[11] Dudka, K.; Peringer, P.; Vojnar, T., Computer Aided Verification, Predator: A Practical Tool for Checking Manipulation of

Dynamic Data Structures Using Separation Logic, (2011), Springer

[12] Dudka, K.; Peringer, P.; Vojnar, T., Static Analysis, Byte-Precise Verification of Low-Level List Manipulation, (2013), Springer

[13] Frühwirth, T., Theory and Practice of Constraint Handling Rules, Journal of Logic Programming, 37, (1998) · Zbl 0920.68029

[14] Frühwirth, T., Constraint Handling Rules, (2009), Cambridge University Press · Zbl 1182.68039

[15] Hinze, R.; Paterson, R., Finger Trees: A Simple General-purpose Data Structure, Journal of Functional Programming, 16, 2, (2006) · Zbl 1088.68041

[16] Jacobs, B.; Smans, J.; Philippaerts, P.; Vogels, F.; Penninckx, W.; Piessens, F., NASA Formal methods, VeriFast: a Powerful, Sound, Predictable, Fast Verifier for C and Java, (2011), Springer

[17] Kroening, D.; Tautschnig, M., Tools and Algorithms for the Construction and Analysis of Systems, CBMC: C Bounded Model Checker, (2014), Springer

[18] (2018)

[19] Matthews, J.; Moore, J.; Ray, S.; Vroon, D., Logic for Programming, Artificial Intelligence, and Reasoning, Verification Condition Generation Via Theorem Proving, (2006), Springer

[20] Merz, F.; Falke, S.; Sinz, C., Verified Software: Theories, Tools, Experiments, LLBMC: Bounded Model Checking of C and C++ Programs Using a Compiler IR, (2012), Springer

[21] Reynolds, J., Logic in Computer Science, Separation Logic: A Logic for Shared Mutable Data Objects, (2002), IEEE