Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 98 (2004) 3–4
www.elsevier.com/locate/entcs

Verifying Programs that Manipulate Pointers (Invited Talk)

Anders Møller[1]

BRICS, University of Aarhus, Denmark

Abstract

This talk gives an overview of various approaches for verifying the correctness of programs that manipulate data structures using pointers.

Keywords: separation logic, parametric shape analysis, pointer assertion logic

Outline of the Talk

Reasoning about the correctness of programs that manipulate data structures using pointers is notoriously difficult: Destructive updating through pointers can make complex structures, the heap has an unbounded size, and data-structure invariants typically only hold at the beginning and end of operations. Correctness properties include general requirements, such as, absence of null pointer dereferences, dangling pointers, and leaking memory, but also more specialized requirements, such as, partial correctness of procedures.

This talk will describe three approaches towards verifying programs that manipulate pointers. First, we look into the ideas behind *separation logic* by Reynolds, O'Hearn, and others [2]. This is an extension of Hoare logic introducing a "separating conjunction" operator that asserts that its sub-formulas hold for disjoint parts of the heap. Then, we consider *parametric shape analysis* by Sagiv, Reps, and Wilhelm [3]. This approach is based on data-flow

[1] Email: amoeller@brics.dk

doi:10.1016/j.entcs.2003.10.002

analysis and three-valued logic. The heap is modeled as a logical structure, and properties are expressed in first-order logic with transitive closure. Finally, we describe *pointer assertion logic* by Møller and Schwartzbach [1]. Based on a decidability result for monadic second-order logic on finite trees, this technique models heap structures using "graph types" and encodes program code as formulas through the use of Hoare logic.

These approaches share the common goal of modeling the complex infinite-state systems that arise when pointers are used in programs; however, they have different domains of applicability, different degrees of automation, and different scalability properties.

Slides are available at http://www.brics.dk/~amoeller/talks/infinity.pdf.

References

[1] Møller, A. and M. I. Schwartzbach, *The pointer assertion logic engine*, in: *Proc. ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI '01*, 2001, pp. 221–231.

[2] Reynolds, J. C., *Separation logic: A logic for shared mutable data structures*, in: *Proc. 17th Annual IEEE Symposium on Logic in Computer Science, LICS'02*, 2002, pp. 55–74.

[3] Sagiv, S., T. W. Reps and R. Wilhelm, *Parametric shape analysis via 3-valued logic*, ACM Transactions on Programming Languages and Systems **24** (2002), pp. 217–298.