© 2013 FIZ Karlsruhe

io-port 05804106

de Dios, Javier; Montenegro, Manuel; Peña, Ricardo

Certified absence of dangling pointers in a language with explicit deallocation.

Méry, Dominique (ed.) et al., Integrated formal methods. 8th international conference, IFM 2010, Nancy, France, October 11–14, 2010. Proceedings. Berlin: Springer (ISBN 978-3-642-16264-0/pbk). Lecture Notes in Computer Science 6396, 305-319 (2010).

Summary: Safe is a first-order eager functional language with facilities for programmer controlled destruction of data structures. It provides also regions, i.e. disjoint parts of the heap, where the program allocates data structures, so that the runtime system does not need a garbage collector. A region is a collection of cells, each one big enough to allocate a data constructor. Deallocating cells or regions may create dangling pointers. The language is aimed at inferring and certifying memory safety properties in a Proof Carrying Code like environment. Some of its analyses have been presented elsewhere. The one relevant to this paper is a type system and a type inference algorithm guaranteeing that well-typed programs will be free of dangling pointers at runtime. Here we present how to generate formal certificates about the absence of dangling pointers property inferred by the compiler. The certificates are Isabelle/HOL proof scripts which can be proof-checked by this tool when loaded with a database of previously proved theorems. The key idea is proving an Isabelle/HOL theorem for each syntactic construction of the language, relating the static types inferred by the compiler to the dynamic properties about the heap that will be satisfied at runtime.

Keywords: Memory management; type-based analysis; formal certificates; proof assistants
doi:10.1007/978-3-642-16265-7_22