Pentagons: A Weakly Relational Abstract Domain for the Efficient Validation of Array Accesses

Francesco Logozzo & Manuel Fähndrich
Microsoft Research
{ logozzo, maf }@microsoft.com

ABSTRACT

We introduce Pentagons (Pntg), a weakly relational numerical abstract domain useful for the validation of array accesses in byte-code and intermediate languages (IL). This abstract domain captures properties of the form of $\mathtt{x} \in [a, b] \wedge \mathtt{x} < \mathtt{y}$. It is more precise than the well known Interval domain, but it is less precise than the Octagon domain.

The goal of Pntg is to be a lightweight numerical domain useful for adaptive static analysis, where Pntg is used to quickly prove the safety of most array accesses, restricting the use of more precise (but also more expensive) domains to only a small fraction of the code.

We implemented the Pntg abstract domain in Clousot, a generic abstract interpreter for .NET assemblies. Using it, we were able to validate 83% of array accesses in the core runtime library mscorlib.dll in less than 8 minutes.

Keywords

Abstract Domains, Abstract Interpretation, Bounds checking, Numerical Domains, Static Analysis, .NET Framework

1. INTRODUCTION

The goal of an abstract interpretation-based static analysis is to statically infer properties of the execution of a program that can be used to ascertain the absence of certain runtime failures. Traditionally, such tools focus on proving the absence of out-of bound memory accesses, divisions by zero, overflows, or null dereferences.

The heart of an abstract interpreter is the abstract domain, which captures the properties of interest for the analysis. In particular, several *numerical* abstract domains have been developed, *e.g.*, [6, 9, 11], that are useful to check properties such as out of bounds and division by zero, but also aliasing [12], parametric predicate abstraction [3] and resource usage [10].

In this paper we present Pentagons, Pntg, a new numerical abstract domain designed and implemented as part of Clousot, a generic static analyzer based on abstract interpretation of MSIL. We intend Clousot to be used by developers during coding and testing phases. It should therefore be scalable, yet sufficiently precise. To achieve this aim, Clousot is designed to adaptively choose the necessary precision of the abstract domain, as opposed to fixing it *before* the analysis (*e.g.*, [8]). Thus, Clousot must be able to discharge most of the "easy checks" very quickly, hence focusing the analysis only on those pieces of code that require a more precise abstract domain or fixpoint strategy.

Clousot uses the abstract domain of Pntg to quickly analyze .NET assemblies and discharge most of the proof obligations from the successive phases of the analysis. As an example let us consider the code in Fig. 1, taken from the basic component library of .NET. Clousot, instantiated with the abstract domain Pntg, automatically discovers the following invariant at program point $(*)$:

$$0 \leq \mathtt{num} < \mathtt{array.Length} \wedge 0 \leq \mathtt{num2} < \mathtt{array.Length}$$

This is sufficient to prove that $0 \leq \mathtt{index} < \mathtt{array.Length}$, *i.e.*, the array is never accessed outside of its bounds.

The elements of Pntg are of the form $\mathtt{x} \in [a, b] \wedge \mathtt{x} < \mathtt{y}$, where x and y are program variables and a, b are rationals. Such elements allow expressing (most) bounds of program variables, and in particular those of array indices: intervals $[a, b]$ take care of the numerical part (*e.g.*, to check array underflows $0 \leq a$), and disequalities $\mathtt{x} < \mathtt{y}$ handle the symbolic reasoning (*e.g.*, to check array overflows $\mathtt{x} < \mathtt{arr.Length}$).

Pntg is therefore an abstract domain more precise than the widespread Intervals, Intv [4], as it adds symbolic reasoning, but it is less precise than Octagons, Oct [9], as it cannot for instance capture equalities such as $\mathtt{x} + \mathtt{y} == 22$. We found that Pntg is precise enough to validate 83% of the array bound accesses (lower and upper) in mscorlib.dll, the main library in the .NET platform, in less than 8 minutes. Similar results are obtained for the other assemblies of the .NET framework. Thus, Pntg fits well with the programming style adopted in this library. Nevertheless, it is not the ultimate abstract domain for bounds analysis. In fact, when used on part of Clousot's implementation, it validates only 65.6% of the accesses.

2. NUMERICAL ABSTRACT DOMAINS

Abstract interpretation is a theory of approximations, [4]. It captures the intuition that semantics are more or less precise depending on the observation level. The observation level is formalized by the notion of an abstract domain. An abstract domain $\bar{\mathsf{D}}$ is a complete lattice $\langle \mathsf{E}, \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$,

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil
Copyright 2008 ACM 978-1-59593-753-7/08/0003 ...$5.00.