Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 331 (2017) 57–70

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Sparsity Preserving Algorithms for Octagons

Jacques-Henri Jourdan

MPI-SWS, Inria Paris

Abstract

Known algorithms for manipulating octagons do not preserve their sparsity, leading typically to quadratic or cubic time and space complexities even if no relation among variables is known when they are all bounded. In this paper, we present new algorithms, which use and return octagons represented as weakly closed difference bound matrices, preserve the sparsity of their input and have better performance in the case their inputs are sparse. We prove that these algorithms are as precise as the known ones.

Keywords: Numerical abstract domain, Octagon abstract domain, static analysis, static analyzer

1 Introduction

In order to capture numerical properties of programs, static analyzers use *numerical abstract domains*. The choice of a numerical abstract domain in a static analyzer is a compromise between *precision*, the ability of capturing complex numerical properties, and performance. Non-relational abstract domains, such as intervals [6], are very efficient but relatively imprecise: they cannot represent relations between program variables. On the other hand, in order to capture numerical relations between program variables, one can express them as linear inequalities. This class of relational numerical abstract domain is composed of *linear abstract domains*. A linear abstract domain corresponds to a different precision vs. performance trade-off: they range from the less precise, efficient ones such as zones [13], pentagons [12] or octagons [13,14] to the more precise, costly ones, such as subpolyhedra [11], octahedra [5], two variables per inequalities [16], zonotopes [15] or general polyhedra [8].

In particular, the Octagon abstract domain [13,14] accurately represents many of the variable relationships appearing in a program, while being still reasonably fast (all the operations have quadratic or cubic complexity on the number of variables). It is very popular in the static analysis community, which explains why algorithmic improvements [3,1,17] and precision improving variants [4] are regularly published.

[1] This work was supported by Agence Nationale de la Recherche, grant ANR-11-INSE-003.

http://dx.doi.org/10.1016/j.entcs.2017.02.004

As reported by the designers of Astrée [7], its quadratic or cubic performances still make it unusable as-is with a reasonable number of variables. Indeed, the data structures typically used to represent octagonal abstract values, i.e., strongly closed difference bound matrices, have a quadratic size in the number of variables for which an upper or lower bound is known. A common solution is the use of *variable packing* [13, §8.4.2], where the Octagon abstract domain is only used on small packs of variables. The downside of packing is that no relation is stored between variables that are not in the same pack. A variant of packing has been introduced to mitigate the imprecision [2], but loss in precision can still occur.

The problem of the performance of octagons has already been studied: in particular, Singh et al. [17] proposed an implementation of the Octagon abstract domain optimized in the case its representation is sparse. But they do not address the fact that it is dense as soon as interval bounds are known for many variables, and we anticipate that, for this reason, the sparsity is very low in their implementation.

Instead, in this paper, we propose to use new algorithms for the Octagon abstract domain: these algorithms work on a sparse representation for octagons, so that the cost of the analysis of two independent sets of variables is the sum of the costs of the analyses of the two sets of variables, taken independently. Our algorithms have the same precision as the traditional ones. Our main idea is the following: in order to ensure an optimal precision of all the operations, the data structures representing octagons, difference bound matrices, are usually kept *strongly closed*: that is, algorithms make sure that any returned difference bound matrix is a best abstraction. However, most often, strongly closed difference bound matrices are dense because of the necessary *strengthening* step. In this paper, we propose to weaken the maintained invariant on difference bound matrices and to keep them *weakly closed* hence skipping the strengthening step. Weakly closed difference bound matrices are not necessarily dense, so that we can use sparse data structures to represent them. We prove that some algorithms can be kept unchanged to work on weakly closed difference bound matrices without losing any precision and give new algorithms for the other operations.

We begin by preliminary definitions in §2. In §3, we describe and prove the soundness and relative precision of our new algorithms. We conclude in §4.

2 Definitions

Let $\mathbb{V}_+$ be a finite set of variables. We call a *regular environment* a function from $\mathbb{V}_+$ to $\mathbb{R}$. A regular environment represents the numerical state of a program. The role of the Octagon abstract domain is to approximate sets of regular environments ρ. To that end, the abstract domain of octagons stores a set of inequalities of the following form:

$$\pm\rho(u) \pm \rho(v) \leq Cst_{uv} \qquad u, v \in \mathbb{V}_+ \tag{1}$$

This corresponds to giving bounds to sums and differences of values of ρ. Moreover, if we use twice the same variable with the same sign, we see that, using such constraints, we can express interval constraints over values of an environment [13].