Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
Electronic Notes in Theoretical Computer Science 267 (2010) 59–72
www.elsevier.com/locate/entcs

Relational Abstract Domain of Weighted Hexagons

Jędrzej Fulara[1,2], Konrad Durnoga[3], Krzysztof Jakubczyk[1,4] and Aleksy Schubert[1,5]

Institute of Informatics
University of Warsaw
ul. Banacha 2
02-097 Warsaw, Poland

Abstract

We propose a new numerical abstract domain for static analysis by abstract interpretation, the domain of Weighted Hexagons. It is capable of expressing interval constraints and relational invariants of the form $x \leq a \cdot y$, where x and y are variables and a denotes a non-negative constant. This kind of domain is useful in analysis of safety for array accesses when multiplication is used (e.g. in guarding formulæ or in access expressions). We provide all standard abstract domain operations, including widening operator, as well as a graph-based algorithm for checking satisfiability and computing normal form for elements of the domain. All described operations are performed in $O(n^3)$ time. Expressiveness of this domain lies between the Pentagons by Logozzo and Fähndrich and the Two Variables Per Inequality by Simon, King and Howe.

Keywords: Numerical abstract domains, static analysis, abstract interpretation.

1 Introduction

The concrete semantics of a program yields possibly infinite computations, hence answering any non trivial questions may be infeasible. A simpler, yet not fully precise, model can be employed to reason about program properties. Abstract Interpretation [6] is a widely used technique that simplifies the process of computation so that its vital properties can be captured within finite resources (time, space etc.) of a computing machine. This technique has been successfully applied in various fields, including program verification [2], error discovery and debugging [3], specification

[1] This work was partly supported by Polish government grant N N206 493138.
[2] Email: fulara@mimuw.edu.pl
[3] Email: kdr@mimuw.edu.pl
[4] Email: kjk@mimuw.edu.pl
[5] Email: alx@mimuw.edu.pl

doi:10.1016/j.entcs.2010.09.006

```
Require: array input[1...m]
1: {input.len ≤ m}
2: output := array [1...2 * input.len]
3: {output.len ≤ 2 · input.len; ... ; m ≤ 1/2 · output.len}
4: for i := 1 to m do
5:    {i ≤ 1/2 · output.len; ...}
6:    output[2 * i - 1] := input[i]
7:    output[2 * i] := input[i]
8: end for
9: return output
```

Fig. 1. A simple code fragment along with invariants that allow proving correctness of the array accesses in lines 6 and 7.

generation or code optimisation during compilation, including program transformation [5].

The Abstract Interpretation Framework allows one to automatically infer invariants that describe some properties of the analysed program. To apply this framework we need to define an *abstract domain* — a representation of the invariants, and operations on them (union, intersection, widening, satisfiability test, etc.) [8]. In this paper, we present an abstract domain that can express relations between pairs of numerical variables x, y of the form $x \leq a \cdot y$, where a is a non-negative constant, as well as interval constraints $x \in [b, c]$. In a two-dimensional case such constraints describe a polygon with at most 6 edges and angles determined by the coefficients from the inequalities (Figure 2). This motivates the name *Weighted Hexagons*.

Example

Figure 1 presents a simple procedure that duplicates all entries of the given *input* array. Static analysis using the domain of Weighted Hexagons can automatically infer in line 5 an invariant $i \leq \frac{1}{2} \cdot output.len$.

1.1 Related Work

In the past, several numerical abstract domains were developed. The most basic one is the Domain of Intervals [7] that represents invariants of the form $x \in [a, b]$. It is efficient (linear time and memory), but does not handle relations between variables. The domain of Convex Polyhedra [10] is very precise — it represents invariants of the form $\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n \leq c$ where $v_1, \ldots, v_n$ are program variables and $c, \alpha_1, \ldots, \alpha_n$ are numerical constants. However the domain representation and operations are inefficient (exponential in the number of variables), and so not practical for many applications. The domain of Octagons [1,12] stores constraints $\pm x \pm y \leq c$ where x and y are program variables and c is a constant. All domain operations can be performed in $O(n^3)$ time using $O(n^2)$ of memory, where n denotes the number of variables. Using the domain of Pentagons [11] one can represent numerical intervals together with symbolic inequalities between variables of the form $x < y$. The complexity of domain operations is $O(n^2)$, but the authors do not present a satisfiability test (if a domain element describes a system of constraints with empty