(Linear) Algebra for Program Analysis

[Abstract]

Markus Müller-Olm *
FernUniversität Hagen
LG Praktische Informatik V
58084 Hagen, Germany
mmo@ls5.cs.uni-dortmund.de

Helmut Seidl
Universität Trier
FB 4-Informatik
54286 Trier, Germany
seidl@uni-trier.de

May 19, 2003

In this talk I reported on ongoing research in which we use techniques from algebra and linear algebra to construct highly precise analysis routines for imperative programs or program fragments. We are interested in programs that work on variables taking values in some fixed ring or field $\mathbb{F}$, e.g., the integers or the rationals. Our analyses precisely interpret assignment statements with affine or polynomial right hand side and treat other assignment statements as well as guarded branching statements conservatively as non-deterministic statements.

More specifically, we consider affine and polynomial programs (see below) as abstractions of programs. Our analyses are precise for affine and polynomial programs, respectively, and may be applied to more general kind of programs by abstracting these to an affine or polynomial program before (or as part of) the analysis. In *affine programs* assignments of the form $x_j := a_0 + \sum_{i=1}^{n} a_i x_i$ are allowed, where $a_0, \ldots, a_n \in \mathbb{F}$ are constants and $x_1, \ldots, x_n$ are the program variables. In order to allow to safely abstract assignments of a different form, we also allow non-deterministic assignments $x :=?$ in affine programs. Branching is non-deterministic. *Polynomial programs* generalize affine programs. Here, deterministic assignments of the form $x_j := p$, where $p \in \mathbb{F}[x_1, \ldots, x_n]$ is a multi-variate polynomial in the program variables, are allowed.

Our results may be summarized as follows:

1. We have an interprocedural analysis that determines for every program point of an affine program all valid affine relations[1] [3].

2. We have generalized this analysis to polynomial relations of bounded degree[2] and to affine programs with local variables [3].

*On leave from Universität Dortmund.

[1] An *affine relation* is a condition of the form $a_0 + \sum_{i=1}^{n} a_i x_i = 0$, where $a_0, \ldots, a_n \in \mathbb{F}$ are constants from the underlying field and $x_1, \ldots, x_n$ are the program variables. A relation is *valid* at a program point, if it holds whenever control reaches that program point.

[2] A *polynomial relation* is a condition of the form $p(x_1, \ldots, x_n) = 0$ where $p(x_1, \ldots, x_n)$ is

3. For polynomial programs, we have constructed an intraprocedural analysis that decides validity of polynomial relations [2, 1]. We can also compute all valid polynomial relations of bounded degree intraprocedurally in polynomial programs.

The running time of analysis 1 and 2 is linear in the size of the program and polynomial in the number of program variables. For analysis 3 we have a termination guarantee but do not know an upper complexity bound. These analyses have many potential applications, because analysis questions can often be coded as affine or polynomial relations easily. Some obvious examples are:

- x is a constant of value $a \in \mathbb{F}$ at a program point p iff the affine relation $x - a = 0$ is valid at p;
- x and y have always the same value at p iff the affine relation $x - y = 0$ is valid at p; and
- x only takes values in the set $\{a_1, \ldots, a_k\} \subseteq \mathbb{F}$ iff the polynomial relation $(x - a_1) \cdot \ldots \cdot (x - a_k)$ is valid.

Our analyses may also be used for program verification purposes: they compute the strongest invariant in affine and polynomial programs that can be stated as a conjunction of affine or (bounded degree) polynomial identities.

Preliminary results of this line of research can be found in the references.

References

[1] M. Müller-Olm. Variations on Constants. Habilitationsschrift, Fachbereich Informatik, Universität Dortmund, 2002.

[2] M. Müller-Olm and H. Seidl. Polynomial constants are decidable. In M. Hermenegildo and G. Puebla, editors, *SAS 2002 (Static Analysis of Systems)*, volume 2477 of *Lecture Notes in Computer Science*, pages 4–19. Springer, 2002.

[3] M. Müller-Olm and H. Seidl. Computing interprocedurally valid relations in affine programs. Technical report, Universität Trier, Fachbereich 4-Informatik, 2003.

a multi-variate polynomial in $x_1, \ldots, x_n$ over $\mathbb{F}$. A polynomial relations is bounded by $d \in \mathbb{N}$ if the sum of exponents of every monomial in p is less than or equal d.