AMERICAN MATHEMATICAL SOCIETY
MathSciNet
Mathematical Reviews


Citations | From References: 1 | From Reviews: 0

**MR3734216** 03F30 03D15 52B55 68Q17
**Nguyen, Danny** (1-UCLA); **Pak, Igor** (1-UCLA)

## ★Short Presburger arithmetic is hard. (English summary)

*58th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2017*, 37–48, *IEEE Computer Soc.*, *Los Alamitos*, *CA*, 2017.

Presburger arithmetic allows integer variables, integer constants, Boolean operations (&, $\wedge$, $\neg$), quantifiers ($\exists$, $\forall$), equations and inequalities ($=$, $<$, $>$, $\leq$, $\geq$), addition and subtraction ($+$, $-$) and multiplication by integer *constants*. It does not allow multiplication of *variables* (if we allow multiplication of variables, we get Peano arithmetic). Geometrically, a quantifier-free formula of Presburger arithmetic describes the set of integer points in a Boolean combination of rational polyhedra (that is, in the set obtained from finitely many rational polyhedra by taking unions, intersections and complements). Similarly, a formula of Presburger arithmetic with existential quantifiers only describes the set of integer points obtained from the set of integer points in a Boolean combination of polyhedra by a projection along some coordinates.

Unlike Peano arithmetic, Presburger arithmetic is decidable. Here the authors zoom in on the computational complexity of Presburger arithmetic, once the combinatorial complexity of the formula is bounded in advance. If we fix the number of variables, the validity of a formula with no quantifier alternations (that is, of the type $\exists x_1 \dots \exists x_k \Phi(x_1, \dots, x_k)$ or of the type $\forall x_1 \dots \forall x_k \Phi(x_1, \dots, x_k)$) can be established in polynomial time by Lenstra's integer programming algorithm [see H. W. Lenstra Jr., Math. Oper. Res. **8** (1983), no. 4, 538–548; MR0727410]. For a fixed number of variables, formulas with one quantifier alternation ($\exists x_1 \dots \exists x_k \forall y_1 \dots \forall y_m \Phi(x_1, \dots, x_k, y_1, \dots, y_m)$) can also be solved in polynomial time, as shown by R. Kannan [in *Polyhedral combinatorics (Morristown, NJ, 1989)*, 39–47, DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 1, Amer. Math. Soc., Providence, RI, 1990; MR1105115]. The decision procedure can be characterized as a polynomial time algorithm for parametric integer programming.

Suppose now that we fix the number of variables *and* the number of Boolean operations in advance (and hence get what is called a *short* formula of Presburger arithmetic). Thus the only parameters of the formula are the numerical values of the constants in the formula. The authors show that deciding validity becomes NP-complete if one allows two quantifier alternations. Remarkably, they present an example of a formula

$$\exists z \in \mathbf{Z}\ \forall y \in \mathbf{Z}^2\ \exists x \in \mathbf{Z}^2\ \Phi(x, y, z)$$

with an NP-complete decision problem, even though $\Phi$ contains at most 10 inequalities. Another remarkable example is an NP-complete decision problem for a formula of the type

$$\exists z \in \mathbf{Z}\ \forall y \in \mathbf{Z}^2\ \exists x \in \mathbf{Z}^6 :\ Ax + By + Cz \leq b$$

with at most 24 inequalities.

As the number of quantifier alternations is allowed to increase, the computational complexity in the polynomial hierarchy also moves up. The authors also describe the computational complexity of corresponding counting problems.

The proof is very clever; it uses the continued fraction expansion of a rational number to encode a growing family of intervals, with the help of which the authors build an NP-complete problem.

{For the collection containing this paper see MR3734212}

*Alexander I. Barvinok*

© *Copyright American Mathematical Society 2018*