Complexity of Presburger Arithmetic with Fixed Quantifier Dimension

Uwe Schöning
Universität Ulm
Abt. Theoretische Informatik
D-89069 Ulm

Abstract. It is shown that the decision problem for formulas in Presburger arithmetic with quantifier prefix $[\exists_1 \forall_2 \ldots \exists_m \forall^3]$ (for m odd) and $[\exists_1 \forall_2 \ldots \forall_m \exists^3]$ (for m even) is complete for the class Σ_m^P of the polynomial-time hierarchy. Furthermore, the prefix type $[\exists\forall\exists\exists]$ is complete for Σ_2^P, and the prefix type $[\exists\forall]$ is complete for NP. This improves results (and solves a problem left open) by Grädel [7].

1 Introduction

We assume familiarity with the standard classes in complexity theory, like P and NP, the classes Σ_m^P of polynomial-time hierarchy, and the notion of polynomial-time reduction (see [6, 9]).

Let 3-CNF (3-DNF) denote the set of propositional formulas in conjunctive (disjunctive) normal form where each clause consists of 3 literals. A literal is a Boolean variable or a negated Boolean variable.

The decision problem B_m consists of Boolean formulas of the form $F(X^1, \ldots, X^m)$ where each X^i is a separate sequence of Boolean variables, $X^i = (x_1^i, \ldots, x_{n_i}^i)$, such that

$$\exists X^1 \, \forall X^2 \ldots Q_m X^m \; F(X^1, X^2, \ldots, X^m)$$

is true. It is known that for each $m \geq 1$, B_m is complete for Σ_m^P (cf. [12, 13]).

Presburger arithmetic is the first-order theory of the natural numbers with addition. Let PA denote the set of formulas which are true in this interpretation. It is known that PA has double-exponential complexity on alternating Turing machines (cf. [4, 3, 1]) whereas the complexity of PA_m, the set of true Presburger formulas with m quantifier alternations, is roughly one exponential step lower than the general case (cf. [10, 5, 7]).

Formulas with fixed dimension are obtained by fixing the quantifier prefix (and therefore also the number of alternations) to a certain type which we will denote by

$[Q_1 Q_2 \dots Q_m]$, $Q_i \in \{\exists, \forall\}$. Grädel has obtained in his dissertation [7] several Σ_m^P-completeness results for decision problems of the form $[Q_1 Q_2 \dots Q_k] \cap PA$, $k > m$. More precisely, he shows that

$$[\exists_1 \forall_2 \dots \exists_m \exists^2 \forall^3] \cap PA \text{ is complete for } \Sigma_m^P \text{ (if } m \text{ is odd),}$$

$$[\exists_1 \forall_2 \dots \forall_m \forall^2 \exists^3] \cap PA \text{ is complete for } \Sigma_m^P \text{ (if } m \text{ is even).}$$

For the case $m = 1$ he can obtain a stronger result:

$$[\exists \forall \forall] \cap PA \text{ is complete for NP.}$$

By swapping universal and existential quantifiers one can, of course, obtain a dual Π_m^P-completeness result. Furthermore, the same complexity status as listed above have all such prefix classes which extend the above ones by finitely many quantifiers and do not increase the number of alternations. Therefore, the complexity status of all but finitely many prefix types is resolved in terms of a completeness result in the polynomial-time hierarchy. Some prefix types remain open, especially Grädel poses the open problem what the status of $[\exists \forall] \cap PA$ is.

In this paper we will stengthen the above completeness results by including more prefix types, and we resolve thereby the complexity status of $[\exists \forall] \cap PA$; it is NP-complete.

2 Main Result

Theorem. For each $m \geq 1$, the language $[\exists \forall \dots \exists_m \forall^3] \cap PA$ (for m odd) and $[\exists \forall \dots \forall_m \exists^3] \cap PA$ (for m even) is Σ_m^P-complete.

Proof: Membership in Σ_m^P is shown in [7] (relying on results in [8, 11, 10]).

For the following we assume that m is odd. In this case the problem $B_m \cap$ 3-CNF is complete for Σ_m^P (see [12, 13]). (For the case of m even, we need to consider the problem $B_m \cap$ 3-DNF instead. The proof in this case is virtually the same.)

Let $F = F(X^1, X^2, \dots, X^m)$ be a formula in 3-CNF where the X^i are sequences of Boolean variables, $X^i = (x_1^i, \dots, x_k^i)$. We assume without loss of generality that each variable sequence X^i consists of the same number of variables, namely k.

Let

$$p_1, p_2, \dots, p_k$$

be the sequence of the first k primes. It is important to notice that this sequence can be constructed in polynomial-time, relative to the size of F.

For the intended reduction, we want to map the formula

$$\exists X^1 \, \forall X^2 \dots \exists X^m \; F(X^1, X^2, \dots, X^m)$$

to a formula in Presburger arithmetic of the following form

$$\exists z_1\, \forall z_2 \ldots \exists z_m\; G(z_1, z_2, \ldots, z_m)$$

where the z_i are variables that represent natural numbers (and encode the assignments X^i) and G is a Presburger formula intended to check whether these assignments make F true. A Boolean assignment $(x_1, \ldots, x_k) \in \{0,1\}^k$ will be represented by a number z that satisfies the set of modular equations

$$\begin{gathered} z \equiv x_1 \pmod{p_1} \\ z \equiv x_2 \pmod{p_2} \\ \vdots \\ z \equiv x_k \pmod{p_k} \end{gathered}$$

The existence of such a $z < \prod_{j=1}^{k} p_j$ is guaranteed by the Chinese remainder theorem.

We need to construct a Presburger formula $A(z)$ that evaluates to true if and only if the number z correctly represents a Boolean assignment, in the sense above. We need to express that for $j = 1, \ldots, k$ it holds that $(z \bmod p_j) \in \{0,1\}$. Therefore $A(z)$ has the following, tentative form

$$\bigwedge_{j=1}^{k} [\,(z \bmod p_j)\ \in\ \{0,1\}\,]$$

Equivalently,

$$\bigwedge_{j=1}^{k} \bigwedge_{k=2}^{p_j-1} [\, z \not\equiv k \pmod{p_j}\,]$$

The expression in brackets can be rewritten as a formula in Presburger arithmetic:

$$\forall u\, (p_j \cdot u + k \neq z)$$

where the notation $p_j \cdot u$ is an abbreviation for $u + u + \cdots + u$ (p_j times). The universal quantifier can be pulled in front, so that the formula for $A(z)$ gets the final form

$$\forall u \left[\bigwedge_{j=1}^{k} \bigwedge_{k=2}^{p_j-1} (p_j \cdot u + k \neq z) \right]$$

The intended formula

$$\exists z_1\, \forall z_2 \ldots Q_m z_m\; G(z_1, z_2, \ldots, z_m)$$

is indeed equivalent to the following form

$$\begin{array}{llllll} \exists z_1 & (A(z_1) & \wedge & & & \\ & \forall z_2 & (A(z_2) & \rightarrow & & \\ & & \exists z_3 & (A(z_3) & \wedge & \\ & & & & \ddots & \\ & & & & & H(z_1, z_2, \ldots, z_m) \ldots)) \end{array}$$

We have seen that $A(z_i)$ can be expressed by one universal quantifier. This enables us in this case to merge quantifiers of the same type into one quantifier. In particular, the universal quantifier in $A(z_1)$ can be melted together with "$\forall z_2$" since they are connected conjunctively. Similarly, the existential quantifier that we need to express "$A(z_2) \to \ldots$" can be melted together with "$\exists z_3$", and so on. Altogether, we get a quantifier prefix (before the beginning of the formula H) of the form $[\exists_1 \forall_2 \ldots \exists_m \forall]$.

The Presburger formula H is intended to express the fact that F is satisfied by the assignments $(X^1, \ldots, X^m)$. This formula consists of a conjunction of formulas,

$$H = \bigwedge_{i=1}^{n} C_i$$

where C_i expresses in Presburger arithmetic that the ith clause in F is satisfied. As a concrete example, let the literals of this clause be x_3^1, $\neg x_1^2$, x_2^2. We can then, tentatively, express C_i as

$$\neg[\,(z_1 \equiv 0 \pmod{p_3}) \wedge (z_2 \equiv 1 \pmod{p_1}) \wedge (z_2 \equiv 0 \pmod{p_2})\,]$$

Furthermore, we can combine subformulas which start with the same z_i. By the Chinese remainder theorem, there is a number $a < p_1 \cdot p_2$ (which can be efficiently computed, see [2] page 824) such that the above formula is equivalent to

$$\neg[\,(z_1 \equiv 0 \pmod{p_1}) \wedge (z_2 \equiv a \pmod{p_1 \cdot p_2})\,]$$

This can be expressed in Presburger arithmetic as

$$\begin{aligned} & \neg\,[\,\exists u(p_1 \cdot u = z_1) \wedge \exists v(p_1 p_2 \cdot v + a = z_2)\,] \\ \equiv \quad & \forall u \forall v\,[\,(p_1 \cdot u \neq z_1) \vee (p_1 p_2 \cdot v + a \neq z_2)\,] \end{aligned}$$

Like in this specific example we can, in general, express C_i by a formula with one, two or three universal quantifiers, depending on the number of different X^i's (respectively z_i's) in that clause.

These up the 3 universal quantifiers per clause can be moved in front of the whole conjunction such that H gets the following form

$$\forall u \forall v \forall w\,[\,\bigwedge_{i=1}^{n} (\ldots)\,]$$

Now the whole resulting formula has the following structure:

$$\exists z_1 \forall z_2 \ldots \exists z_m\,(A(z_m) \wedge \forall u \forall v \forall w\,[\ldots])$$

Again we can melt together two quantifiers, namely the universal quantifier in $A(z_m)$ and "$\forall u$". So the final form of the Presburger formula has the quantifier prefix $[\exists_1 \forall_2 \ldots \exists_m \forall^3]$. Finally, we remark that the reduction can be carried out in polynomial time. □

Inspecting the proof, by the fact that the number of different X^i's determines the last block of universal quantifiers, we get the following corollary.

Corollary. The decision problem $[\exists\forall\exists\exists] \cap PA$ is Σ_2^P-complete.

The decision problem $[\exists\forall] \cap PA$ is NP-complete.

Remark: The NP-completeness of $[\exists\forall] \cap PA$ could also be obtained by a reduction from some other NP-complete problem; for this, Grädel (personal communication) has proposed problem [AN2] from [6].

References

[1] L. Berman. The complexity of logical theories. *Theoretical Computer Science* 11 (1980) 71–77.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest. *Introduction to Algorithms.* MIT Press, 1990.

[3] J. Ferrante, C.W. Rackoff. *The Computational Complexity of Logical Theories.* Lecture Notes in Mathematics 718, Springer, 1979.

[4] M.J. Fischer, M.O. Rabin. Super-exponential complexity of Presburger arithmetic. *SIAM-AMS Proceedings* 7 (1974) 27–41.

[5] M. Fürer. The complexity of Presburger arithmetic with bounded quantifier alternation depth. *Theoretical Computer Science* 18 (1982) 105–111.

[6] M.R. Garey, D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[7] E. Grädel. *The Complexity of Subclasses of Logical Theories.* Dissertation, Universität Basel, 1987.

[8] H. Lenstra. Integer programming with a fixed number of variables. *Math. of Operations Research* 8 (1983) 538–548.

[9] C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[10] C.R. Reddy, D.W. Loveland. Presburger arithmetic with bounded quantifier alternation. *Proc. of the 10th ACM Symposium on Theory of Computing* 1978, 320–325.

[11] B. Scarpellini. Complexity of subcases of Presburger arithmetic. *Transactions of the AMS* 284 (1984) 203–218.

[12] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science* 3 (1977) 1–22.

[13] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science* 3 (1977) 23–33.