Theoretical Computer Science 56 (1988) 289-301
North-Holland

SUBCLASSES OF PRESBURGER ARITHMETIC AND THE POLYNOMIAL-TIME HIERARCHY

Erich GRÄDEL

Mathematisches Institut der Universität Basel, CH-4051 Basel, Switzerland

Communicated by E. Engeler
Received January 1987

Abstract. We investigate the complexity of subclasses of Presburger arithmetic, i.e., the first-order theory of natural numbers with addition. The subclasses are defined by restricting the quantifier prefix to finite lists $Q_1 \dots Q_s$. For all $m \geq 2$ we find formula classes, defined by prefixes with $m+1$ alternations and $m+5$ quantifiers, which are Σ_m^p- respectively Π_m^p-complete. For $m = 1$, the class of $\exists\forall\forall$-formulas is shown to be NP-complete. For $m = 0$ and for all natural numbers t, the class of $\exists^t$-formulas is known to be in P. Thus we have a nice characterisation of the polynomial-time hierarchy by classes of Presburger formulas. Finally, the NP-completeness of the $\exists\forall\forall$-class is used to prove that for certain formulas there exist no equivalent quantifier-free formulas of polynomial length.

1. Introduction

Most interesting logical theories are either undecidable or have very high complexity. In fact, all theories which have a model with at least one nontrivial relation are hard for PSPACE; this is an easy consequence of a theorem due to Stockmeyer [15]. Most theories even have an exponential lower bound. It thus makes sense to ask whether one can define—by syntactical restrictions—formula classes in such theories which are on the one hand powerful enough to formulate interesting statements but are on the other hand substantially easier to decide. The most natural way to define such formula classes is by restrictions on the quantifier prefix, i.e., by bounds on the *quantifier alternations* or on the total number of quantifiers (the *dimension*) of the formula. Another possibility is to restrict the syntactical structure of the quantifier-free part of the formula.

The limitation to a simpler quantifier structure often reflects the restriction to problems which are mathematically more interesting and significant. Restricting, e.g., arithmetic to existential formulas (i.e., formulas with only $\exists$-quantifiers) gives Hilberts 10th problem; existential formulas of Presburger arithmetic correspond to the question whether a system of linear diophantine inequalities has a solution; the word problem for finitely presented groups is expressed by universal formulas in the theory of groups.

Extensive research has investigated the decision problem and complexity of subclasses of the predicate calculus (see, e.g., the survey article by Börger [2]). The methods developed there turn out to be very useful in the investigations of the

corresponding problem for specific logical theories. In a subsequent paper, a general method for treating such questions, based on bounded versions of the domino-problem, will be presented.

Here we shall consider the complexity of subclasses of Presburger arithmetic (PA). PA is known to have double-exponential complexity by the well-known results due to Fischer/Rabin, Ferrante/Rackoff and Berman [6, 5, 1].

The subclasses PA_m will denote PA restricted to sentences with at most m quantifier alternations or more precisely, with at most m blocks of adjacent quantifiers of the same kind. These classes were investigated by Reddy/Loveland [12]:

$$PA_m \in \text{DSPACE}(2^{cn^{m+4}}) \quad \text{for a constant } c$$

and Fürer [7]:

$$PA_m \notin \text{NTIME}(2^{(n/m)^{cm}}) \quad \text{for a positive constant } c.$$

That is, the complexity of classes with bounded quantifier alternations is roughly one exponential step lower than in the general case.

For very small m, however, the lower bound does not hold. Deciding, e.g., an existential formula reduces to determining whether a system of linear diophantine inequalities is solvable. This is an NP-complete problem, as was shown by Von Zur Gathen and Sieveking [9]. Thus PA_1 is (NP$\cup$Co-NP)-complete. In [7] Fürer put up the question, for which m PA_m actually has an exponential lower bound. We will come back to this in the subsequent paper mentioned above and show that this already happens for $m=2$.

In this paper we shall investigate the complexity of formula classes of PA defined by *fixed* prefixes $Q_1, \ldots, Q_s$ and prove the following results:

(1) For all $m \geq 2$ there are formula classes defined by prefixes with $m+1$ quantifier alternations and $m+5$ quantifiers, which are complete in Σ_m^p and Π_m^p respectively. (In fact, almost all prefixes with $m+1$ alternations have this property.)

(2) For $m=1$ we show that the class of $\exists\forall\forall$-formulas of PA is NP-complete.

Together with the result, due to Lenstra [11] and Scarpellini [13], that, for fixed dimension t, the class of $\exists^t$-formulas is in P, this yields a nice characterisation of the whole polynomial-time hierarchy by classes of Presburger formulas.

In the last section, the NP-completeness of the $\exists\forall\forall$-class will be used to derive a lower-bound result on the length of formulas: if there is a set in NP with nonpolynomial circuit complexity (this is a generalised version of Cook's hypothesis P$\neq$NP), then there are formulas of the form $\exists y\, H(x_1, x_2, x_3, y)$ for which there are no equivalent quantifier-free formulas of polynomial length.

2. Preliminaries

Presburger arithmetic (PA) is the first-order theory of the natural numbers with addition. (By abuse of notation, PA will denote both the model $(\mathbf{N}, +, \leq)$ and the set of sentences which are true in this model.) We shall use the relation $\leq$ (which

is of course definable in PA) as a primitive symbol. Furthermore, $\dot{-}$ will be a function symbol (not a relation) and we shall—in general—allow multiplication with integer constants. Thus, terms have the form $\alpha_1 x_1 + \cdots + \alpha_n x_n + \beta$, where the α_i and β are integers in binary notation; atoms are $(t_1 = t_2)$ or $(t_1 \leq t_2)$ (where t_1, t_2 are terms).

Whereas the exact formalisation of the theory has no influence on the complexity of the theory as a whole, it might be important for certain subclasses because multiplication with integer constants helps to save quantifiers. It will, however, turn out that all upper and lower bounds proved in this paper will hold for the "strong" formalisation described above as well as for "weaker" versions without scalar multiplication.

If $Q_1, \ldots, Q_s$ is any quantifier prefix ($Q_i = \exists$ or $Q_i = \forall$), the subclass $[Q_1 \ldots Q_s] \cap$ PA denotes the set of sentences $Q_1 x_1 \ldots Q_s x_s F(x_1, \ldots, x_s)$ (F quantifier-free), which are true in PA.

PA$\models G$ means that G is true in PA.

The complexity of such subclasses of PA will be described in terms of Stockmeyer's polynomial-time hierarchy [15], which is a subrecursive analogue to the arithmetical hierarchy.

Definition. The *polynomial-time hierarchy* consists of the classes $\Sigma_m^{\mathrm{P}}, \Pi_m^{\mathrm{P}}$ $(m \geq 0)$, where $\Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}} = \mathrm{P}$, and, for $m \geq 1$, a set of words L over an alphabet Γ belongs to Σ_m^{P} iff there is a polynomial $p(n)$ and a set $L' \in \mathrm{P}$ such that, for all $x \in \Gamma^*$,

$$x \in L \text{ iff } (\exists y_1)(\forall y_2)(\exists y_3)\ldots(Q_m y_m)[(x, y_1, y_2, \ldots, y_m) \in L'],$$

where the quantifiers alternate (so Q_m is $\exists$ ($\forall$) if m is odd (even)), and $y_1, \ldots, y_m$ range over all words in Γ^* of length not exceeding $p(|x|)$. Π_m^{P} contains the sets whose complements are in Σ_m^{P}. (Of course, Π_m^{P} can also be defined similarly to Σ_m^{P}, but the string of quantifiers will begin with an $\forall$ instead of a $\exists$.)

Observe that $\Sigma_1^{\mathrm{P}} = \mathrm{NP}$ and $\Pi_1^{\mathrm{P}} = \text{Co-NP}$.

Another possible definition is that Σ_m^{P} (Π_m^{P}) is the class of sets which can be decided by an alternating Turing machine, beginning in an existential (universal) state, in polynomial time with m alternations (the equivalence of the two definitions is proved in [3]).

It is not known whether the obvious inclusions $\Sigma_m^{\mathrm{P}} \cup \Pi_m^{\mathrm{P}} \subseteq \Sigma_{m+1}^{\mathrm{P}} \cap \Pi_{m+1}^{\mathrm{P}}$ are proper; if $\Sigma_m^{\mathrm{P}} = \Sigma_{m+1}^{\mathrm{P}}$ for any m, then $\Sigma_m^{\mathrm{P}} = \Sigma_k^{\mathrm{P}}$ for all $k > m$. If we could separate any two levels of the hierarchy, we thus would prove that $\mathrm{P} \neq \mathrm{NP}$. Obviously, the whole hierarchy is contained in PSPACE, but it is not known whether $\mathrm{PH} := \bigcup_{m \geq 0} \Sigma_m^{\mathrm{P}}$ is equal to PSPACE. If it is, then the polynomial-time hierarchy is finite (see [15, 16] for further results on the polynomial-time hierarchy).

3. Upper bounds

In this section we shall prove upper bounds on the complexity of formula classes defined by finite prefixes $Q_1, \ldots, Q_s$.

If the prefix contains only existential or only universal quantifiers, the problem is known to be in P.

Theorem 3.1 (Lenstra, Scarpellini [11, 13]). *For all fixed dimensions* $t \in \mathbf{N}$, $[\exists^t] \cap \mathrm{PA}$ *and* $[\forall^t] \cap \mathrm{PA}$ *are in* P.

Lenstra proved in [11] that a system of linear diophantine inequalities of fixed dimension t can be solved in polynomial time. In fact, he exhibited an algorithm which, given a $m \times t$-matrix A with integral coefficients and a vector $b \in \mathbf{Z}^m$, finds—in polynomial time—a vector $x \in \mathbf{N}^t$ satisfying the system $Ax \leqslant b$ or decides that no such vector exists. For sentences $\exists x_1 \ldots \exists x_t\, F(x_1, \ldots, x_t)$, where the quantifier-free part F is in disjunctive normal form (DNF), this immediately yields a decision procedure. In [13] Scarpellini reduced the problem for arbitrary formulas F to this case.

We now consider an arbitrary, but fixed prefix $Q_1 \ldots Q_s$. For this purpose we analyse the quantifier elimination procedure for PA as given in [12]: the language $\mathscr{L}$ of PA is embedded in a language $\mathscr{L}'$, which contains the additional relations $(\alpha \mid t)$ and $(\alpha \nmid t)$ (for integers α and terms t) which express that α divides t and that α does not divide t respectively.

Let now $\exists x\, G(x)$ be any formula from $\mathscr{L}'$ with G quantifier-free. By elementary transformations and by moving all negations signs into the atoms (using that $\neg(t_1 \leqslant t_2) \equiv (t_2 + 1 \leqslant t_1)$) we get a formula $\exists x\, G'(x)$ with $G'(x)$ consisting of conjunctions and disjunctions (no negations) of atoms of the following four classes:

$$\text{(A)}\quad \alpha x \leqslant t, \qquad \text{(C)}\quad \beta \mid (\alpha x + t),$$

$$\text{(B)}\quad t \leqslant \alpha x, \qquad \text{(D)}\quad \beta \nmid (\alpha x + t),$$

where $\alpha, \beta \in \mathbf{N}$, t a term not containing x.

We define $G_\infty(x)$ as the formula derived from $G'(x)$ by replacing all atoms of type (A) by FALSE, and all atoms of type (B) by TRUE. $G(t+j/\alpha x)$ means that in all atoms of G (if necessary) both sides have been multiplied by α and then αx has been replaced by the term $(t+j)$ (where j is a natural number). Finally, σ will denote the least common multiple of all β's occurring in atoms of types (C) and (D).

The elimination procedure is now given by the following lemma.

Lemma 3.2

$$\mathrm{PA} \models \exists x\, G(x) \text{ iff } \mathrm{PA} \models \bigvee_{j=0}^{\sigma-1} G_\infty(j) \vee \bigvee_{\text{(A)}} \bigvee_{j=0}^{\alpha\sigma-1} (G'(t+j/\alpha x) \wedge (\alpha \mid t+j))$$

(*where* $\bigvee_{(\mathrm{A})}$ *means that the disjunction is taken over all atoms of type* (A).)

An arbitrary formula $Q_1 x_1 \ldots Q_s x_s\, F(x_1, \ldots, x_s)$ of Presburger arithmetic can be decided by successive application of the following procedure: eliminate the innermost quantifier with help of Lemma 3.2 (if it is an $\forall$, replace it first by $\neg\exists\neg$);

then, by elementary transformations, bring the formula on the right-hand side in Lemma 3.2 into the desired form described above in order to eliminate the next quantifier. This procedure is a slight modification of Cooper's version of the Presburger algorithm. The proof of its correctness (i.e., the proof of Lemma 3.2), is found in [4].

The quantifier-elimination procedure itself is not a good decision procedure because the resultant formula becomes too long. But we can, as in [12], prove upper bounds on the absolute values of the constants that occur in the formula after elimination of the quantifiers $Q_1, \ldots, Q_s$. We shall then be able to infer a bound w such that the quantifiers $Q_1, \ldots, Q_s$ need not range over the whole of N but only over a finite subset limited by w. The bound w will depend on

- n: the *length* of the formula,
- m: the number of *quantifier alternations*, and
- s: the *dimension* of the formula (i.e., the total number of quantifiers).

We shall use essentially the same arguments as in [12], but also take into account s.

Let C_r be a bound for the size of the coefficients of the variables and of the β in the atoms of types (C) and (D) after r quantifier eliminations. Obviously, $C_0 \leqslant 2^n$; by the elimination of a quantifier, a coefficient is at most multiplied by another coefficient (by the substitution $G(t+j/\alpha x)$) and added to another coefficient (by collecting terms). Thus

$$C_r \leqslant C_{r-1}^2 + C_{r-1} \leqslant 2C_{r-1}^2.$$

This proves the following lemma.

Lemma 3.3. $C_r \leqslant 2^{2^r-1} \cdot 2^{2^r n} \leqslant 2^{2^r(n+1)}$.

It is the parameter σ, the least common multiple of the β from the divisibility and nondivisibility relations, that takes the main responsibility for the length of the resulting formula after the elimination of the quantifiers. Let the σ after r eliminations be limited by S_r. As we begin the procedure with a Presburger formula (without $|$ and $\nmid$), we have $S_0 = 1$. Eliminating a quantifier we introduce, for each α from an atom $\alpha x \leqslant t$, a number of new divisibility relations $\alpha \mid t+j$. This means that $S_r \leqslant S_{r-1}^{q_{r-1}}$, where q_{r-1} is the number of different α. It remains to determine q_r, which not only depends on the number of the eliminated quantifiers, but as well on their alternations.

We call two inequalities of forms $(\alpha x \leqslant t)$ or $(t \leqslant \alpha x)$ *coefficient-distinct* if at least one variable has different coefficients in the two inequalities. Let q_j^i be the maximal number of coefficient-distinct inequalities after elimination of a string of quantifiers with i alternations and j quantifiers after the last alternation.

Lemma 3.4. $q_r^m \leqslant n^{(r+1)^{m+1}}$.

Proof. $q_0^0 \leqslant n$ and $q_1^0 \leqslant n^2$ because the disjunction $\bigvee_{(A)}$ runs over at most n inequalities; each clause has the same number of coefficient-distinct inequalities as the original formula.

In order to eliminate a string of $\exists$-quantifiers, after every elimination we commute the disjunction signs with the next $\exists$ and eliminate $\exists$ in every disjunction separately. Each disjunction will have the same number of coefficient-distinct inequalities as before the last elimination. That gives $q_r^0 \leq n^{r+1}$ and, more generally, $q_r^i \leq (q_0^i)^{r+1}$. (For a string of $\forall$-quantifiers $\forall\forall\ldots\forall$ this also holds because it corresponds to a string $\neg\exists\exists\ldots\exists\neg$.)

Whenever a quantifier alternation occurs, the negation sign (after replacing $\forall$ by $\neg\exists\neg$) that must be imported into the formula has the effect that the next $\exists$ faces a conjunction of formulas. Thus we have the same situation as in the beginning but with a larger formula as the scope of the $\exists$. Therefore,

$$q_0^1 \leq q_r^0 \leq n^{r+1}, \qquad q_1^1 \leq (q_0^1)^{r+1}$$

and, by repetition,

$$q_0^m \leq q_r^{m-1} \leq (q_0^{m-1})^{r+1} \ldots \leq n^{(r+1)^m}, \qquad q_r^m \leq n^{(r+1)^{m+1}}. \qquad \square$$

We can now find the bound S_r.

Lemma 3.5. *$S_r \leq 2^{(cn)^{(r+2)^{m+1}}}$ for a constant $c > 0$.*

Proof

$$S_r \leq C_{r-1}^{q_{r-1}} \cdot S_{r-1} = \prod_{i=0}^{r-1} C_i^{q_i} \leq C_{r-1}^{(r-1)q_{r-1}} \leq C_{r-1}^{(r-1)q_{r-1}^m}$$

if the eliminated string has m quantifier alternations. With help of Lemmas 3.3 and 3.4 it follows that

$$S_r \leq 2^{2^{r-1}(n+1)(r-1)n^{r^{m+1}}}$$

which (together with $r \leq n$) proves the lemma. $\square$

Finally, we need a bound for the additive constants after r eliminations. We denote this bound by K_r.

Lemma 3.6. *$K_r \leq 2^{cn^{(r+3)^{m+1}}}$ for a constant $c > 0$.*

Proof. Obviously, the lemma is true for $K_0 \leq 2^n$. While a quantifier is removed, an additive constant may be multiplied by a coefficient and added to an additive constant from the substituting term $t+j$ (where $j < C_{r-1}S_{r-1}$). We assume the lemma to be true for $r-1$ and conclude $K_r \leq (2C_{r-1}+1)K_{r-1}$ because $S_{r-1} \leq K_{r-1}$; so

$$K_r \leq K_{r-1}^2 \leq 2^{2cn^{(r+2)^{m+1}}} \leq 2^{cn^{(r+3)^{m+1}}}. \qquad \square$$

With almost verbatim the same arguments (which are therefore omitted) as in [12] we can now infer the following theorem.

Theorem 3.7. *Let G be a sentence $Q_1x_1 \ldots Q_sx_s\ F(x_1, \ldots, x_s)$ of length n with m quantifier changes. Then*

$$\mathrm{PA} \models G \text{ iff } \mathrm{PA} \models (Q_1x_1 \leq w)(Q_2x_2 \leq w) \ldots (Q_sx_s \leq w)\ F(x_1, \ldots, x_s)$$

where $w = 2^{cn(s+3)^{m+2}}$.

This immediately yields a decision procedure for arbitrary subclasses of Presburger arithmetic, which are defined by quantifier prefixes: Given a sentence G as in Theorem 3.7, compute w from n, m, s and check for all s-tuples of natural numbers $a_1, \ldots, a_s$ with $a_i \leq w$, whether $G(a_1, \ldots, a_s)$ is true. An alternating Turing machine can do this within time $(s \cdot \log w)^k$ (for a $k \in \mathbf{N}$) and with m alternations.

We can thus conclude from Theorem 3.7 the following already known facts [5, 12, 1]:

$$\mathrm{PA} \in \mathrm{ATIME}(2^{2^{cn}}, n),$$
$$\mathrm{PA}_m \in \mathrm{ATIME}(2^{dn^{m+2}}, m) \qquad \text{for constants } c \text{ and } d.$$

On the other hand, we can now prove upper bounds for those formula classes which are defined by finite prefixes as shown in the next theorem.

Theorem 3.8. *Let $Q_1 \ldots Q_s$ be a quantifier prefix with m alternations.*

(a) *If $Q_1 = \exists$, then $[Q_1 \ldots Q_s] \cap \mathrm{PA} \in \Sigma^{\mathrm{p}}_{m-1}$.*

(b) *If $Q_1 = \forall$, then $[Q_1 \ldots Q_s] \cap \mathrm{PA} \in \Pi^{\mathrm{p}}_{m-1}$.*

Proof. For fixed s and m, there is a polynomial $p(n)$ such that $w \leq 2^{p(n)}$ (see Theorem 3.7). The length of the binary representation of any natural number $a \leq w$ is then bounded by $p(n)$. Thus $[Q_1 \ldots Q_s] \cap \mathrm{PA}$ is the set of sentences $Q_1x_1, \ldots, Q_sx_s\ F(x_1, \ldots, x_s)$ such that

$$\mathrm{PA} \models (Q_1|x_1| \leq p(n)) \ldots (Q_s|x_s| \leq p(n))\ F(x_1, \ldots, x_s).$$

We now cut the prefix in two, just right of the last Q_k different from Q_s, i.e., $Q_1 \ldots Q_s = Q_1 \ldots Q_k | Q_s \ldots Q_s$. The string left of the | has now $m-1$ quantifier alternations. For any choice of $a_1, \ldots, a_k$ (with $a_i \leq w$), the Lenstra-Scarpellini-algorithm decides

$$Q_sx_{k+1} \ldots Q_sx_s\ F(a_1, a_2, \ldots, a_k, x_{k+1}, \ldots, x_s)$$

in polynomial time (see Theorem 3.1).

This means that $[Q_1 \ldots Q_s] \cap \mathrm{PA}$ is defined by $m-1$ alternations of polynomially bounded quantifiers over a set in P, and hence (recall the definitions) is in $\Sigma^{\mathrm{p}}_{m-1}$ (if $Q_1 = \exists$) or in Π^{p}_{m-1} (if $Q_1 = \forall$). □

4. Completeness results

This section will show that the upper bounds proved in Section 3 are optimal for most prefixes, even for "weak" versions of PA without $\leqslant$ and without scalar multiplication. For all complexity classes Σ_m^{P}, Π_m^{P} with $m \geqslant 2$ we shall find formula classes of Presburger arithmetic defined by prefixes with $m+1$ alternations and $m+5$ quantifiers, which are *complete* in Σ_m^{P} respectively Π_m^{P}.

For NP and Co-NP we even find somewhat stronger results as is shown now.

Theorem 4.1

(a) $[\exists\forall\forall] \cap \mathrm{PA}$ *is* NP-*complete.*

(b) $[\forall\exists\exists] \cap \mathrm{PA}$ *is* (Co-NP)-*complete.*

Proof. The second claim follows from the first, and Theorem 3.8 yields that $[\exists\forall\forall] \cap$ PA is in NP.

NP-completeness will be shown by reducing 3SAT (i.e., the set of satisfiable Boolean formulas in CNF with at most three litterals in each conjunction clause) to $[\exists\forall\forall] \cap \mathrm{PA}$.

Let $\varepsilon = \varepsilon_1, \ldots, \varepsilon_n \in \{0,1\}^n$. We encode ε by the natural number $a_\varepsilon = p_1^{\varepsilon_1} p_2^{\varepsilon_2} \ldots p_n^{\varepsilon_n} q_1^{\delta_1} q_2^{\delta_2} \ldots q_n^{\delta_n}$ where $\delta_i = 1 - \varepsilon_i$ $(1 \leqslant i \leqslant n)$ and $p_1, \ldots, p_n, q_1, \ldots, q_n$ are the first $2n$ primes. Note that these can be constructed in polynomial time.

We first define a formula $A_n(x)$ expressing that

$$\text{for all } i \leqslant n \quad p_i \mid x \text{ iff } q_i \nmid x,$$

i.e., that x encodes some $\varepsilon \in \{0,1\}^n$ (namely the only ε such that a_ε divides x). $A_n(x)$ is the conjunction of the two formulas

$$\forall u \bigwedge_{i=1}^{n} p_i q_i u \neq x,$$

$$\forall u \forall v \bigwedge_{i=1}^{n} \left\{ \bigwedge_{m=1}^{p_i - 1} p_i u \neq x + m \vee \bigwedge_{m=1}^{q_i - 1} q_i v \neq x + m \right\}.$$

The first expresses that p_i and q_i do not both divide x, the second that either p_i or q_i divide x (using the fact that if p does not divide $x+m$ for all $m < p$, then p divides x). Note that pu is just a shorthand for $u + u + \cdots + u$ (p times). Thus $A_n(x)$ is a $\forall\forall$-formula which can be constructed in polynomial time and has the desired properties.

Let now F be a Boolean formula in 3-CNF

$$F(X_1, \ldots, X_n) \equiv \bigwedge_i C_i \equiv \bigwedge_i Y_{i_1} \vee Y_{i_2} \vee Y_{i_3}$$

where $Y_{i_j} \in \{X_1, \ldots, X_n, \neg X_1, \ldots, \neg X_n\}$. We now map each clause $Y_{i_1} \vee Y_{i_2} \vee Y_{i_3}$ in F to a nondivisibility relation $s_i \nmid x$ $(s_i \in \mathbb{N})$. First we substitute the literals Y_{i_j} by relations $r_{i_j} \nmid x$, namely:

$$X_k \mapsto (q_k \nmid x), \qquad \neg X_k \mapsto (p_k \nmid x)$$

(p_k, q_k as above). This extends to a substitution of the clause C_i:

$$(Y_{i_1} \vee Y_{i_2} \vee Y_{i_3}) \mapsto (r_{i_1} \nmid x \vee r_{i_2} \nmid x \vee r_{i_3} \nmid x) \mapsto (r_{i_1} r_{i_2} r_{i_3} \nmid x).$$

Denote $r_{i_1} r_{i_2} r_{i_3} \nmid x$ by s_i. $F(X_1, \ldots, X_n)$ is satisfiable if and only if there is an a_ε which is not divided by any of the s_i. This means

$$F(X_1, \ldots, X_n) \in 3\text{SAT} \text{ iff } \text{PA} \models G, \quad \text{where } G \equiv \exists x \{A_n(x) \wedge \forall u \bigwedge_i s_i u \neq x\}.$$

As G is an $\exists\forall\forall$-sentence which can be constructed in polynomial time, the theorem is proved. □

Corollary 4.2. *For all* $s \geq 1$, $t \geq 2$,
(a) $[\exists^s \forall^t] \cap \text{PA}$ *is* NP-*complete*;
(b) $[\forall^s \exists^t] \cap \text{PA}$ *is* Co-NP-*complete.*

We now extend this idea to arbitrary levels of the polynomial-time hierarchy. We shall reduce certain sets of quantified Boolean formulas to appropriate classes in PA.

Definition. Let $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^m$ be vectors of variables (i.e., $\mathbf{X}^i = X_1^i, X_2^i, \ldots, X_{n_i}^i$), and let F be Boolean formulas. Then

$$B_m := \{F \mid (\exists \mathbf{X}^1)(\forall \mathbf{X}^2) \ldots (Q\mathbf{X}^m)[F(\mathbf{X}^1, \ldots, \mathbf{X}^m) \text{ is TRUE}]\}.$$

Similarly,

$$\bar{B}_m := \{F \mid (\forall \mathbf{X}^1)(\exists \mathbf{X}^2) \ldots (Q\mathbf{X}^m)[F(\mathbf{X}^1, \ldots, \mathbf{X}^m) \text{ is TRUE}]\}.$$

By restricting F to 3-CNF and 3-DNF respectively we define the sets $B_m \cap$ 3-CNF and $B_m \cap$ 3-DNF (and similarly for $\bar{B}_m$).

In [16, 15] Wrathall and Stockmeyer show that B_m and $B_m \cap$ 3-CNF (if m is odd) respectively $B_m \cap$ 3-DNF (if m is even) are Σ_m^{P}-complete. Corresponding results hold for $\bar{B}_m$ and Π_m^{P}.

We shall reduce these sets to formula classes of PA to prove the last theorem of this section.

Theorem 4.3. *If m is odd, then*
(a) $[\exists_1 \forall_2 \ldots \exists_m \exists^2 \forall^3] \cap \text{PA}$ *is* Σ_m^{P}-*complete*;
(b) $[\forall_1 \exists_2 \ldots \forall_m \forall^2 \exists^3] \cap \text{PA}$ *is* Π_m^{P}-*complete.*
If m is even, then
(c) $[\exists_1 \forall_2 \ldots \forall_m \forall^2 \exists^3] \cap \text{PA}$ *is* Σ_m^{P}-*complete*;
(d) $[\forall_1 \exists_2 \ldots \exists_m \exists^2 \forall^3] \cap \text{PA}$ *is* Π_m^{P}-*complete* (*via polynomial reduction*).

Proof. First of all, note that all these prefixes have $m+1$ quantifier alternations; the formula classes they define are therefore contained in Σ_m^P respectively Π_m^P (see Theorem 3.8).

We shall explicitly construct a polynomial reduction from $B_m \cap 3\text{-CNF}$ (for odd m) to $[\exists_1\forall_2\ldots\exists_m\exists^2\forall^3] \cap \mathrm{PA}$ and thus prove (a). The other claims follow by similar considerations.

Let $F(\mathbf{X}^1,\ldots,\mathbf{X}^m)$ be a Boolean formula in 3-CNF, $\mathbf{X}^i = X_1^i,\ldots,X_{n_i}^i$, and $n = \max\{n_i \mid 1 \leq i \leq m\}$. As in Theorem 4.1, $\varepsilon^i = \varepsilon_1^i,\ldots,\varepsilon_n^i \in \{0,1\}^n$ is encoded by $a_{\varepsilon^i} = p_1^{\varepsilon_1^i} p_2^{\varepsilon_2^i}\ldots p_n^{\varepsilon_n^i} q_1^{\delta_1^i} q_2^{\delta_2^i}\ldots q_n^{\delta_n^i}$ (where $\delta_k^i = 1-\varepsilon_k^i$, $p_1,\ldots,p_n,q_1,\ldots,q_n$ are the first $2n$ primes).

Again we translate $F(\mathbf{X}^1,\ldots,\mathbf{X}^m) \equiv \bigwedge_j Y_{j_1} \vee Y_{j_2} \vee Y_{j_3}$ into a Boolean combination of nondivisibility relations. The substitutions

$$X_k^i \mapsto (q_k \nmid x_i); \qquad \neg X_k^i \mapsto (p_k \nmid x_i)$$

transform $F(\mathbf{X}^1,\ldots,\mathbf{X}^m)$ into a Presburger-formula

$$G(x_1,\ldots,x_m) \equiv \forall u\forall v\forall w\left\{\bigwedge_j (r_{j_1}u \neq z_{j_1}) \vee (r_{j_2}v \neq z_{j_2}) \vee (r_{j_3}w \neq z_{j_3})\right\},$$

where the r_{j_k} are primes from $p_1,\ldots,p_n,q_1,\ldots,q_n$ and z_{j_k} variables from $x_1,\ldots,x_m$.

$F(\varepsilon^1,\ldots,\varepsilon^m)$ is true if and only if $G(a_{\varepsilon^1},\ldots,a_{\varepsilon^m})$ is true in Presburger arithmetic. What remains is to relativise the variables x_i in $G(x_1,\ldots,x_m)$ to the numbers a_ε that encode an $\varepsilon \in \{0,1\}^n$. For this, we use the $\forall^2$-formula $A_{n_i}(x_i)$ as constructed in the proof of Theorem 4.1 and their negations $\neg A_{n_i}(x_i)$ which are of course $\exists^2$-formulas.

The reduction then goes like this:

$$\exists\mathbf{X}^1\forall\mathbf{X}^2\ldots\exists\mathbf{X}^m F(\mathbf{X}^1,\ldots,\mathbf{X}^m) \mapsto H,$$

where

$$H \equiv \exists x_1\forall x_2\ldots\exists x_m\left(\bigvee_{\substack{i=2\\ i\text{ even}}}^{m-1} \neg A_{n_i}(x_i)\right) \vee \left(\bigwedge_{\substack{i=1\\ i\text{ odd}}}^{m} A_{n_i}(x_i) \wedge G(x_1,\ldots,x_m)\right).$$

Now, $\neg A_{n_i}$ are $\exists^2$-formulas, hence their disjunction is also equivalent to an $\exists^2$-formula; the formulas G and A_{n_i} are $\forall^3$-respectively $\forall^2$-formulas, hence their conjunction can also be written as an $\forall^3$-formula. The quantifier prefix of the whole formula H is thus $\exists_1\forall_2\ldots\exists_m\exists^2\forall^3$ and $F(\mathbf{X}^1,\ldots,\mathbf{X}^m)$ is in B_m if and only if $\mathrm{PA} \models H$. □

Thus we have a characterisation of all levels of the polynomial-time hierarchy by complete formula classes from Presburger arithmetic.

Of course, Theorem 4.3 extends to all quantifier prefixes with m alternations which contain those explicitly mentioned in the theorem; e.g., for all $t_1,\ldots,t_m \in \mathbb{N}$ we have Σ_m^P-completeness of the formula class $[\exists^{t_1}\forall^{t_2}\ldots\exists^{t_m}\exists^2\forall^3] \cap \mathrm{PA}$. That is, we have shown that *almost all* finite-prefix classes of PA are complete in a level of the

polynomial-time hierarchy. Some classes, however, remain open; the most interesting of them is the class with prefix $\exists\forall$. Is it in P? Is it NP-complete. We have to leave this as an open problem.

5. Lower bounds for length of formulas

From the fact that $[\exists\forall\forall]\cap \mathrm{PA}$ is NP-complete it follows that—unless P = NP—there is no polynomial procedure which transforms formulas $\exists y\, G(x_1, x_2, x_3, y)$ into equivalent quantifier-free formulas of PA. In this section we shall show that if a stronger, nonuniform, version of the hypotheses $\mathrm{P} \neq \mathrm{NP}$ holds, then there are formulas $\exists y\, G(x_1, x_2, x_3, y)$ for which equivalent quantifier-free formulas of polynomial length *do not even exist.* This is a considerably stronger claim than the nonexistence of a "fast" elimination procedure, which does not imply that there are no "short" quantifier-free formulas equivalent to G, but only that such "short" formulas would be difficult to produce.

We denote by GCH (*generalised Cook hypothesis*) the conjecture that there is a set $L \subseteq \{0,1\}^*$ in NP with nonpolynomial circuit complexity. Nonpolynomial circuit complexity means that there is no family of Boolean circuits $(C_n)_{n\in\mathbf{N}}$ of size $|C_n| \leq p(n)$ (p a polynomial) such that C_n decides $L \cap \{0,1\}^n$.

The GCH implies $\mathrm{P} \neq \mathrm{NP}$ because every $T(n)$-time-bounded Turing machine can be simulated by a family $(C_n)_{n\in\mathbf{N}}$ of circuits of size $|C_n| = \mathrm{O}(T(n) \log T(n))$ (see, e.g., [14]). On the other hand, the GCH is generally believed to be true because its converse would imply that the polynomial-time hierarchy collapses to its second level, i.e., $\mathrm{PH} = \Sigma_2^{\mathrm{P}}$ (see [10]). For a general discussion of the relationship between circuit complexity and other complexity measures see [14].

Let now L be any problem in NP, with $L \subseteq \{0,1\}^*$.

Lemma 5.1. *There is a family of Boolean formulas*

$$(F_n^L(X_1, \ldots, X_n, Y_1, \ldots, Y_{q(n)}))_{n\in\mathbf{N}}$$

which are constructible in time $p(n)$ (p, q polynomials) such that

$$F_n^L(w_1, \ldots, w_n, Y_1, \ldots, Y_{q(n)}) \in 3\mathrm{SAT} \text{ iff } w \in L$$

for all $w = w_1 w_2, \ldots, w_n \in \{0,1\}^n$.

This lemma is of course just a tiny generalisation of the NP-completeness of 3SAT. The only difference is that we have not just a reduction $w \mapsto F_w^L$ for all $w \in \{0,1\}^*$; in addition we state that the F_w^L can be obtained from $F_{|w|}^L(X, Y)$ by substitution of the variables X by w.

The proofs for the NP-completeness of SAT and 3SAT (see, e.g., [8]) translate without any problems to a proof of Lemma 5.1.

We now use essentially the same reduction from Boolean formulas in 3-CNF to Presburger formulas as in the proof of Theorem 4.1, but consider the variables $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_{q(n)}$ separately. This means that we translate literals as follows

$$X_i \mapsto q_i \nmid x, \qquad Y_j \mapsto q_j \nmid y,$$
$$\neg X_i \mapsto p_i \nmid x, \qquad \neg Y_j \mapsto p_j \nmid y$$

(where $i \leq n, j \leq q(n)$). Applying this reduction to the formulas $F_n^L(X, Y)$ in exactly the same way as in the proof of Theorem 4.1, we obtain a family of Presburger-formulas $G_n^L(x) \equiv \exists y \{A_{q(n)}(y) \wedge G_n'(x, y)\}$. $A_{q(n)}(y)$ expresses that, for all $i \leq q(n)$, either p_i or q_i divides y, but not both simultaneously (p_i and q_i are the first $2q(n)$ primes). If $a, b \in \mathbf{N}$ encode $\mathrm{w} \in \{0,1\}^n$ and $\varepsilon \in \{0,1\}^{q(n)}$ respectively, then $G_n'(a, b)$ is true in PA if and only if $F_n(\mathrm{w}, \varepsilon)$ is a true Boolean formula. $A_{q(n)}$ and G_n' are both $\forall^2$-formulas, so the $G_n^L(x)$ are again $\exists\forall\forall$-formulas.

All this is summarised by the following lemma.

Lemma 5.2. *Let $L \in \{0,1\}^*$ be any set in* NP. *There is a family of Presburger formulas $G_n^L(x)$ with prefix $\exists\forall\forall$, constructible in polynomial time such that, for all $\mathrm{w} = w_1, \ldots, w_n \in \{0,1\}^n$*

$$\mathrm{PA} \models G_n^L(a_\mathrm{w}) \text{ iff } \mathrm{w} \in L$$

where $a_\mathrm{w} = p_1^{w_1} \ldots p_n^{w_n} q_1^{1-w_1} \ldots q_n^{1-w_n}$.

Theorem 5.3. *The GCH implies: for all polynomials p there exist formulas $\exists y\, H(x_1, x_2, x_3, y)$, with no equivalent formulas $\tilde{H}(x_1, x_2, x_3)$ of length $|\tilde{H}| \leq p(|H|)$ (H, $\tilde{H}$ quantifier-free).*

Proof. Assume the contrary and look at the formulas $G_n^L(x) \equiv \exists y \forall u \forall v\, H_n^L(x, y, u, v)$ constructed above. As the H_n^L are quantifier-free and of length polynomial in n, we can derive—by applying the assumption three times—that there is a family $(\tilde{H}_n^L(x))_{n \in \mathbf{N}}$ of quantifier-free formulas such that

(a) $|\tilde{H}_n^L(x)| \leq p(n)$ for a polynomial p;

(b) $\mathrm{PA} \models \tilde{H}_n^L(a_\mathrm{w})$ iff $\mathrm{w} \in L$ (w and a_w as above).

Now, for every quantifier-free formula $F(x)$, there certainly is a polynomial algorithm which, given w, constructs a_w and decides whether $F(a_\mathrm{w})$ is true. Hence there are Boolean circuits of polynomial size deciding for given w whether $H_n^L(a_\mathrm{w})$ is true, that is, whether $\mathrm{w} \in L$. So L has polynomial circuit complexity and as L was an arbitrary set in NP, this contradicts the GCH. □

This method can be applied not only to Presburger arithmetic but to any logical theory where lower bounds for the complexity of a finite-prefix class are known. Among other results, such applications will be presented in a forthcoming paper on the complexity of subclasses of logical theories.

Acknowledgment

The results presented in this paper are part of the author's Ph.D. Thesis, directed by Prof. B. Scarpellini. I would like to thank him for many helpful discussions.

References

[1] L. Berman, The complexity of logical theories, *Theoret. Comput. Sci.* **11** (1980) 71-77.
[2] E. Börger, Decision problems in predicate logic, in: *Proc. Logic Colloquium 82* (North Holland, Amsterdam, 1984) 263-301.
[3] A.K. Chandra, D.C. Kozen and L. Stockmeyer, Alternation, *J. ACM* **28** (1981) 114-133.
[4] D.C. Cooper, Theorem proving in arithmetic without multiplication, *Machine Intelligence* **7** (1972) 91-100.
[5] J. Ferrante and Ch. Rackoff, *The Computational Complexity of Logical Theories*, Lecture Notes in Mathematics **718** (Springer, Berlin, 1979).
[6] M.J. Fischer and M.O. Rabin, Super-exponential complexity of Presburger arithmetic, *SIAM-AMS Proc.* **7** (1974) 27-41.
[7] M. Fürer, The complexity of Presburger arithmetic with bounded quantifier alternation depth, *Theoret. Comput. Sci.* **18** (1982) 105-111.
[8] M.R. Garey and D.S. Johnson, *Computers and Intractability. A Guide to the Theory of* NP-*Completeness*, (Freeman, San Francisco, 1979).
[9] J. Von Zur Gathen and M. Sieveking, A bound on solutions of linear integer equalities and inequalities, *Proc. AMS* **72** (1978) 155-158.
[10] R.M. Karp and R.J. Lipton, Some connections between nonuniform and uniform complexity classes, in: *Proc. 12th ACM Symp. on Theory of Computing* (1980) 302-309.
[11] H. Lenstra, Integer programming with a fixed number of variables, *Math. Oper. Res.* **8** (1983) 538-548.
[12] C.R. Reddy and D.W. Loveland, Presburger arithmetic with bounded quantifier alternation, in: *Proc. 10th ACM Symp. on Theory of Computing* (1978) 320-325.
[13] B. Scarpellini, Complexity of subcases of Presburger arithmetic, *Trans. AMS* **284** (1984) 203-218.
[14] U. Schöning, *Complexity and Structure*, Lecture Notes in Computer Science **211** (Springer, Berlin, 1986).
[15] L. Stockmeyer, The polynomial-time hierarchy, *Theoret. Comput. Sci.* **3** (1977) 1-22.
[16] C. Wrathall, Complete sets and the polynomial-time hierarchy, *Theoret. Comput. Sci.* **3** (1977) 23-33.