# NP-COMPLETE DECISION PROBLEMS FOR QUADRATIC POLYNOMIALS[†]

Kenneth Manders and Leonard Adleman

Group in Logic and Methodology of Science
and
Computer Science Division
Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California at Berkeley
Berkeley, California 94720

## 1. ABSTRACT

In this article we show the NP-completeness of some simple number-theoretic problems. Natural simplifications of these problems invariably are known to be in P. Our research was motivated by the question whether one could study non-deterministic computation without loss of generality on a restricted, number theoretically significant class of nondeterministic Turing machines, the nondeterministic diophantine machines defined below [1,2]. The results suggest this is true.

Because of the relative difficulty of the reduction of the satisfiability problem used in the proof, and the distinctly number-theoretic character of the problems shown to be NP-complete, we hope that the NP-completeness of these problems will play a role in showing the NP-completeness of further problems of a numerical nature, much as the satisfiability problem has in showing the NP-completeness of combinatorial problems ([3],[5]).

The results illustrate an intimate connection between problems in computational theory, such as "P = NP?", and problems in number theory, e.g. about quadratic congruences in one unknown, which are just beyond the range in which efficient algorithms exsit. Thus our work exposes an interface between the state-of-the-art in number theory and in the theory of computation. Also, our results can be seen as the solution to a natural and important revision of Hilbert's 10th problem: Give a feasible algorithmic procedure to decide whether an arbitrary diophantine equation has solutions. Unless P = NP this is impossible, even for a class of quadratic diophantine equations in two unknowns for which a decision procedure in the original sense is in fact available. Certainly, these results present a striking rigorous demonstration that number theorists' intuitions about where problems about diophantine equations and quadratic congruences start to be truly difficult are justified.

The theorems are stated and discussed in Sections 2 and 3 and proven in Section 4. A list of open problems suggested by the results is included at the end of the paper.

## 2. DEFINITIONS

(a) NP is the collection of relations on the natural numbers accepted by some nondeterministic Turing machine (NDTM) in polynomial time.*

(b) P is the collection of relations on the natural numbers accepted by some deterministic Turing machine (DTM) in polynomial time.

(c) A set $S \in NP$ is said to be NP-complete if and only if for any set $A \in NP$ there is a deterministic polynomial-time computable recursive function $f$ such that

$$(\forall x \in \omega)[x \in A \Leftrightarrow f(x) \in S] \ .$$

Theorem 1. The problem of accepting the set of diophantine equations (in a standard binary encoding)

$$\alpha x_1^2 + \beta x_2 - \gamma = 0 \qquad \alpha, \beta, \gamma \in \omega$$

which have (natural number) solutions is NP-complete.

Theorem 2. The problem of accepting the set of quadratic congruences (in a standard encoding)

$$x^2 \equiv \alpha \bmod \beta$$

with solutions satisfying $0 \leq x \leq \gamma$, $\alpha, \beta, \gamma \in \omega$ is NP-complete, even if the prime factorization of

*A set A is accepted in polynomial time on a NDTM (NDDM, DTM) if and only if there is a NDTM (NDDM, DTM) M and a polynomial q such that

$\alpha \in A \Leftrightarrow$ there is some computation of M on input $\alpha$ which halts in $q(|\alpha|)$ steps (where $|\alpha|$ is the length of $\alpha$ in binary) .

[†] Research supported by National Science Foundation Grant DCR72-03725-A02.

$\beta$ and all solutions to the congruence modulo all prime powers occurring in this factorization are given gratis.

The strength of the theorems can be made more clear by considering closely related problems known to be in P:

(1) The problem of accepting those diophantine equations in one variable which have solutions.

(2) The problem of accepting those linear diophantine equations in many variables which have solutions.

(3) The problem of accepting the set of quadratic congruences

$$x^2 \equiv \alpha \bmod \beta \qquad \alpha, \beta \in \omega$$

with natural number solutions, where the factorization of $\beta$ is given.

Thus little simplification of the problems shown NP-complete by our arguments is needed to obtain a problem in P.

It is interesting to compare the number of unknowns in the equations about which our results give information with the number of unknowns for which unsolvability results are available. It followed from the solution to Hilbert's 10th problem in 1970 [7] that for some fixed $n$, the problem of deciding the set of solvable $n$ variable diophantine equations was unsolvable. Since then much effort has been devoted to determining the minimum $n$ for which this is true. The best published result is $N = 13$ [8]; this has since been improved to 9 [9]. In [8], it is conjectured that $N = 3$ may be possible, and this seems to agree with experience in studying diophantine equations. However, it is improbable that any result like $N = 3$ could be proved by present methods. Our results show that this subcase is in fact intractable.

## 3. NONDETERMINISTIC DIOPHANTINE MACHINES[†]

We introduce a new type of computational device called a Nondeterministic Diophantine Machine (NDDM). The interest of this new formalism for computability stems from the fact that it is both number-theoretically and machine-theoretically convenient and hence provides a direct interface between number theory and machine theory ([1],[2]). Use of such a convenient formalism has been made possible by the development of advanced techniques in number theory for the solution of Hilbert's 10th problem.

Given a multivariable polynomial $p(x_1,\ldots,x_n)$ with integer coefficients, the corresponding $\mathrm{NDDM}^n$ is the nondeterministic Turing machine (NDTM) with the following algorithm:

"On input $a \in \omega$, guess $a_2,\ldots,a_n \in \omega$.
If $p(a,a_2,\ldots,a_n) = 0$, accept $a$."[††]

For example, if $p(x_1,x_2) = x_1 - x_2^2$, then the corresponding NDDM has the algorithm

"On input $a \in \omega$, guess $a_2 \in \omega$.
If $a_1 - a_2^2 = 0$, accept $a$."

It is easy to see that this NDDM accepts exactly the set of perfect squares in polynomial time.

In discussing the relationships among NP, P and D, the class of sets accepted in polynomial time on NDDM's, the following definitions are useful.

(i) (alternative characterization of D) For all $n \in \omega$, $D^n$ is the set of all numerical relations $R$ definable by a formula of the form:

$$\begin{aligned} &<x_1,x_2,\ldots,x_m> \in R \\ &\Leftrightarrow \exists y_1,\ldots,y_n \\ &\quad \le 2^{q(|(x_1+x_2+\cdots+x_m)|)} [P(x_1,x_2,\ldots,x_m, \\ &\qquad\qquad y_1,\ldots,y_n) = 0] \end{aligned}$$

where $q$ and $p$ are polynomials. $D = \bigcup_{i \in \omega} D^i$.

(ii) For any m-ary numerical relation $R$ and any $\ell$-ary numerical relation $S$: R is D-reducible to S (notation: $R \le_D S$) if and only if $R$ is definable by a formula of the form:

$$\begin{aligned} &<x_1,\ldots,x_m> \in R \\ &\Leftrightarrow \exists y_1,\ldots,y_\ell,y_{\ell+1},\ldots,y_n \\ &\quad \le 2^{q(|x_1+\cdots+x_m|)} \\ &\qquad [P(x_1,\ldots,x_m,y_1,\ldots,y_n) = 0 \\ &\qquad\ \& <y_1,\ldots,y_\ell> \in S] \end{aligned}$$

where $q$ and $p$ are polynomials.

In more intuitive terms, $R \subseteq W$ is D-reducible to $S \subset W$ if and only if there is a polynomial $P(x_1,\ldots,x_n)$ such that the NDTM $M$ with the following algorithm:

"On input $x$, guess $x_2,\ldots,x_n$.
If $x_2 \in S$ and $P(x,x_2,\ldots,x_n) = 0$,
accept $x$."

accepts $R$ in polynomial time. Clearly $R \le_D S$ and $S \in D$ implies $R \in D$.

(iii) For any numerical relation $R$ in NP: R is NP(D)-complete if and only if every numerical relation in NP is D-reducible to $R$. Clearly if $R$ is NP(D)-complete then $R \in D \Leftrightarrow NP = D$ (see open problems).

(iv) $D^{2(K)}$ will denote the subclass of $D^2$ where the relevant polynomial $P(x_1,\ldots,x_m,y_1,y_2)$ is of degree $\le K$ in the variables $y_1$, $y_2$ (e.g.

[†] Primary motivation for this section is provided in [1] and [2].

[††] For definiteness we will assume that addition is done in linear time and multiplication in time $n^2$.

$P(x_1,y_1,y_2) = x_1y_1^2y_2$ is of degree 3 in $y_1$, $y_2$).

Theorem 3. Let $S$ be the set

$$\{<\alpha,\beta,\gamma>|\ \alpha,\beta,\gamma \in \omega\ \&\ (\exists x_1x_2 \in \omega)\ [\alpha x_1^2 + \beta x_2 - \gamma = 0]\}$$

a) The problem of accepting $S'$ is NP-complete.

b) $S \in D^{2(2)}$

Theorem 4. a) There is an NP-complete problem in $D$.

b) There is a NP(D)-complete problem in NP (in fact in P).

Theorem 5. The following are equivalent.

(a) $\bar{S} \in NP$

(b) $(D^{2(2)})^c \subseteq NP$

(c) $(NP)^c = NP$

where $A^c$ denotes the set of complements of sets in $A$ and $\bar{A}$ denotes the complement of $A$.

Theorem 3 answers an open problem in [1]: "Find a set $A \in D$ such that for all $B \subseteq \omega$, if $B \in D$, then $B$ is polynomial reducible to $A$." That any such set would be NP-complete (as Theorem 3 implies) was rather unexpected. One would have expected the $D^i$, $i \in \omega$ to be a hierarchy of progressively harder problems (in the sense of P-reducibility); moreover, it was suspected that number-theoretic problems obviously in NP would be less than NP-complete: The deep structure of number theory should allow development of nontrivial and efficient algorithms for such problems. Theorem 3 indicates that all this is wrong.

Theorem 4 illustrates a symmetry of $P$ and $D$ as subclasses of NP, also seen in other contexts ([2]). Theorem 5 is, in many respects, as strong as possible: We can show ([2]) that

(i) $(D^{2(1)})^c \subseteq P \subseteq NP$

(ii) $(D')^c \subseteq D \cap P \subseteq NP$

## 4. PROOFS OF THE THEOREMS

Theorems 3 and 5 are direct consequences of Theorem 1; Theorem 4 follows from the proof of Theorem 1: the D-complete set in P is obtained by trivial modification of the deterministic algorithm constructed in that proof; the NDDM needed for a proof that this set is D-complete is obtained by trivial modification of the P-complete NDDM whose existence is implied by Theorem 1. We will not elaborate on this argument in the present abstract.

Theorems 1 and 2 are obtained by a common argument. Let $S$ be the set of satisfiable propositional formulas in conjunctive normal form with at most 3 littorals per clause. By Cook [3] it suffices to show that there is a deterministic polynomial-time algorithm which reduces the problem of membership to a problem of the form(s) mentioned in the theorem, and that the problems themselves are in NP. Both problems considered are solvable by a nondeterministic "guess a solution and check whether it is correct" algorithm in polynomial time, and hence in NP. This is because, as is easily verified, there is a bound on the size of possible solutions to either problem given by a polynomial in the coefficients $\alpha$, $\beta$, $\gamma$.

For a complete proof of Theorem 2 it is further necessary to show that the reduction algorithm also yields the factorization of the modulus and the solutions of the congruence module the prime powers dividing the modulus; it will be evident from the discussion below that this is indeed the case.

We now give the reduction algorithm, followed by proof of correctness and analysis of computation time. The reader may wish to merely skim the algorithm initially, referring back to it when this is suggested in the proof and analysis.

### 4.1. The Algorithm

"On input $\phi$, read $\phi$ and eliminate all duplicate conjuncts and those in which, for some variable $x_i$, both $x_i$ and $\bar{x}_i$ occur. Count the $\ell$ variables occurring in the remaining formula $\phi_R$. Let

$$\Sigma = \{\sigma_1,\ldots,\sigma_m\}$$

be a standard enumeration of all possible disjunctive clauses, formed from $x_1,\ldots,x_\ell$ and their complements, with at most 3 littorals per clause. Setting

$$\varepsilon_j = \begin{cases} 1 & \text{if } \sigma_j \text{ occurs in } \phi_R \\ 0 & \text{otherwise} \end{cases}, \quad j=1,2,\ldots,m$$

compute $\tau_\phi = \sum_{j=1}^{m} \varepsilon_j \cdot 8^j$

[Comment: $\tau_\phi$ is the only quantity computed which depends specifically on $\phi_R$, rather than just on the number $\ell$ of variables occurring in $\phi_R$.]

Compute:

$$f_i^+ = \sum_{\substack{x_i \text{ occurs} \\ \text{in } \sigma_j}} 8^j, \quad i = 1,2,\ldots,\ell$$

$$f_i^- = \sum_{\substack{\bar{x}_i \text{ occurs} \\ \text{in } \sigma_j}} 8^j, \quad i = 1,2,\ldots,\ell$$

Set $n = 2m + \ell$, and let $P_0,P_1,\ldots,P_n$ be the first $n+1$ primes exceeding

$$\sqrt[n+1]{4\cdot(n+1)\cdot 8^{m+1}}\ ^\dagger$$

Compute

$$m_j = P_j^{n+1}, \quad j = 0,1,\ldots,n$$

$$K = \prod_{j=0}^{n} P_i^{n+1}$$

$$\bar{m}_j = K/m_j, \quad j = 0,1,\ldots,n .$$

$^\dagger$i.e., exceeding 12.

Determine parameters $\lambda_j$, $j = 0,1,\dots,n$, as follows:

(a) $j = 0$: Let $\lambda_0$ be the least $\lambda_0 \in \omega$ such that

$$\begin{cases} \lambda_0\bar{m}_0 \equiv 1 \text{ modulo } 8^{m+1} \\ \lambda_0\bar{m}_0 \not\equiv 0 \text{ modulo } P_0 \end{cases}$$

(b) $j = 1,2,\dots,2m$: Let $\lambda_j$ be the least $\lambda_j \in \omega$ such that

$$\begin{cases} \lambda_j\bar{m}_j \equiv -\frac{1}{2}\, 8^k \text{ modulo } 8^{m+1} \;, & j = 2k-1 \\ \lambda_j\bar{m}_j \equiv -8^k \quad \text{modulo } 8^{m+1} \;, & j = 2k \\ \lambda_j\bar{m}_j \not\equiv 0 \quad \text{modulo } P_j \end{cases}$$

(c) $j = 2m+1,\dots,2m+\ell$: Let $\lambda_j$ be the least $\lambda_j \in \omega$ such that

$$\begin{cases} \lambda_j\bar{m}_j \equiv \frac{1}{2}(f^+_{j-2m} - f^-_{j-2m}) \text{ modulo } 8^{m+1} \\ \lambda_j\bar{m}_j \not\equiv 0 \qquad\qquad \text{modulo } P_j \end{cases}$$

Compute:

$$H = \sum_{j=0}^{n} \lambda_j\bar{m}_j$$

$$\begin{cases} \tau \equiv \tau_\phi + H + \sum_{i=1}^{\ell} f^-_i \text{ modulo } 8^{m+1} \\ 0 \le \tau \le 8^{m+1} \end{cases}$$

Compute and output:

(a) For Theorem 1:

$$K(\tau^2 - x_1^2) + (K+1)^3 \cdot 2 \cdot 8^{m+1}(H^2 - x_1^2) - (x_2 - H^2) \cdot 2 \cdot 8^{m+1}\ K$$
$$= 0$$

(b) For Theorem 2:

$$\begin{cases} x^2 \equiv (2 \cdot 8^{m+1} + K)^{-1}(K\tau^2 + 2 \cdot 8^{m+1}H^2) \text{ module } 2 \cdot 8^{m+1} \cdot K \\ 0 \le x \le H \end{cases}$$

where $(2 \cdot 8^{m+1} + K)^{-1}$ is the inverse of $(2 \cdot 8^{m+1} + K)$ modulo $2 \cdot 8^{m+1} \cdot K$."

### 4.2 Analysis of Computation Time

$\ell$, and hence $m$ and $n$, are bounded by a polynomial in the length of the input $\phi$. Hence, by the Prime Number Theorem, the primes $p_0,\dots,p_n$ are also bounded by such a polynomial. It follows that the size (number of digits in binary representation) of $p_j^n$, K, and H is bounded by a polynomial in the length of $\phi$; hence the same is true of the output of the algorithm.

Moreover, we can obtain all quantities needed deterministically within polynomial time in the length of the input: The primes can be found as we have exponential time in their length to do so. The $\lambda_j$'s are easily obtained from an inverse modulo $8^{m+1}$ of $\bar{m}_j$, which in turn can be obtained quickly using the Euclidean algorithm to show that $m_j$ is relatively prime to $8^{m+1}$. (See, for instance, [6].) All other computations are trivial, given the bounds on the size of the numbers involved.

### 4.3 Proof of Correctness[†]

We show the equivalence of the satisfiability of the conditions output by the algorithm to the satisfiability of the original formula $\phi$, in a number of stages.

Either of the conditions output is equivalent to the system

$$\left.\begin{array}{rl} \text{(i)} & 0 \le |x| \le H \\ \text{(ii)} & (\tau+x)(\tau-x) \equiv 0 \bmod 2 \cdot 8^{m+1} \\ \text{(iii)} & (H+x)(H-x) \equiv 0 \bmod K \end{array}\right\} \qquad \text{(I)}$$

This can be seen directly in the case of condition (b); condition (a) is obtained from (b) after addition of two extra terms (the factor $(K+1)^3$, and the term $H^2$ in $(x_2-H^2)$) in order to ensure adequate reflection of condition (I)(i) in the resulting diophantine equation.

Lemma 1. Let $\tau$ be odd, $x \in \mathbb{Z}$, $k \ge 3$.

$$(\tau+x)(\tau-x) \equiv 0 \bmod 2^{k+1}$$
$$\Leftrightarrow \tau + x \equiv 0 \text{ or } \tau - x \equiv 0 \bmod 2^k .$$

(The straightforward proof is left to the reader.)

We note that in our case the conditions of Lemma 1 are satisfied. Hence the system (I) is equivalent to the system

$$\left.\begin{array}{rl} \text{(i)} & 0 \le |x| \le H \\ \text{(ii)} & \tau + x \equiv 0 \bmod 8^{m+1} \\ \text{(iii)} & (H+x)(H-x) \equiv 0 \bmod K \end{array}\right\} \qquad \text{(II)}$$

Lemma 2. Let K and H be as in the algorithm. The general solution of the system

$$\left.\begin{array}{rl} (1) & 0 \le |x| \le H, \quad x \in \mathbb{Z} \\ (2) & (H+x)(H-x) \equiv 0 \bmod K \end{array}\right\}$$

is given by

$$x = \sum_{j=0}^{n} \alpha_j\lambda_j\bar{m}_j \;, \quad \alpha_j \in \{-1,+1\} \;, \quad j = 0,1,\dots,n$$

Proof (of Lemma 2). It is easy to verify that all x of the given form satisfy the system. We now show that these are the only solutions.

Let x be a solution to the system (1)-(2). Then

$$(H+x)(H-x) \equiv 0 \bmod m_j \;, \quad j = 0,1,\dots,n .$$

Assume (for reductio) that for some $j_0$,

$$p_{j_0} | (H+x) \text{ and } p_{j_0} | (H-x)$$

[†] In this section, $|x|$ will denote the absolute value of x.

(Notation: $a|b$ means $a$ divides $b$; this is equivalent to $b \equiv 0 \bmod a$.) Then $p_{j_0}|(H+x)+(H-x) = 2H$. But $p_{j_0} > 2$, $p_{j_0}$ prime, so we must have $p_{j_0}|H$, i.e. $p_{j_0}|\sum_{j=0}^{n}\lambda_j\bar{m}_j$. But by definition of $\bar{m}_j$, $p_{j_0}|\bar{m}_j$ for all $j \neq j_0$. Hence it would have to be that $p_{j_0}|\lambda_{j_0}\bar{m}_{j_0}$, which conflicts with the choice of $\lambda_{j_0}$.

Thus rejecting our assumption, we conclude that for each $j$,

$$m_j|H+x \quad \underline{\text{or}} \quad m_j|H-x \ .$$

We define

$$\alpha_j = \begin{cases} 1 & \text{if } m_j|H-x \\ -1 & \text{if } m_j|H+x \end{cases} \qquad j = 0,1,\ldots,n \ .$$

$$x' = \sum_{j=0}^{n} \alpha_j\lambda_j\bar{m}_j \ .$$

Then we get:

$$x' \equiv x \bmod m_j \quad \text{for all } j \quad \text{so} \quad x' \equiv x \bmod K$$

$$\left.\begin{array}{l} -H \le x' \le H \\ -H \le x \le H \end{array}\right\} \Rightarrow |x-x'| \le 2H$$

But by our choice of $p_j \ge \sqrt[n+1]{4(n+1)8^{m+1}}$, and the fact that $\lambda_j < 2\cdot 8^{m+1}$ (for each $j$), each term of $H$ is bounded by $K/2(n+1)$. Hence $2H < K$; we now conclude that $x = x'$. Thus any solution of (1)-(2) is indeed of the form given. (End of proof.)

By Lemma 2, the system (II) is equivalent to:

$$\left.\begin{array}{ll} \text{(i)} & \tau + x \equiv 0 \bmod 8^{m+1} \\ \text{(ii)} & x = \sum_{j=0}^{n} \alpha_j\lambda_j\bar{m}_j \ , \quad \alpha_j \in \{-1,+1\}, \\ & \qquad\qquad j = 0,1,\ldots,n \ . \end{array}\right\} \text{(III)}$$

<u>Lemma 3</u>. The solutions to (III) are exactly those $x$ for which $\alpha_0 = -1$, and

$$\left.\begin{array}{l} R_k = 0 \quad \text{for} \quad k = 1,2,\ldots,m, \quad \text{where} \\ R_k = \sum\limits_{\substack{\beta_{i+2m}=1 \\ x_i \text{ occurs} \\ \text{in } \sigma_k}} 1 + \sum\limits_{\substack{\beta_{i+2m}=0 \\ \bar{x}_i \text{ occurs} \\ \text{in } \sigma_k}} 1 - \varepsilon_k - (\beta_{2k-1}+2\beta_{2k}) \\ \beta_j = \begin{cases} 0 & \text{if } \alpha_j = -1 \\ 1 & \text{if } \alpha_j = 1 \end{cases} , \quad j = 1,2,\ldots,n \ . \end{array}\right\} \text{(IV)}$$

<u>Proof</u> (of Lemma 3). Somewhat more explicitly written, (III)(i) becomes

$$\tau_\phi + \sum_{i=1}^{\ell} f_i^- + H + \sum_{j=0}^{n} \alpha_j\lambda_j\bar{m}_j \equiv 0 \bmod 8^{m+1}$$

Combining the last two terms, we get

$$\tau_\phi + \sum_{i=1}^{\ell} f_i^- + \sum_{j=0}^{n} 2\cdot\beta_j\cdot\lambda_j\bar{m}_j \equiv 0 \bmod 8^{m+1}$$

from which, with the definitions of $\tau_\phi$, $f_i^+$, $f_i^-$, $\lambda_j\bar{m}_j$, we get

$$2\beta_0 + \sum_{k=1}^{m} R_k\cdot 8^k \equiv 0 \bmod 8^{m+1} \ ,$$

where $R_k$ is as defined in the statement of the lemma. From that definition and the facts of the situation we note that

$$-4 \le R_k \le +3$$

and from this we find (by an induction argument on $m$) that the conditions (IV) are necessary. On the other hand, these conditions are now clearly sufficient. (End of proof)

From the conditions (IV) given in Lemma 3 it is easy to see that the solutions to (III) and valuations $v(\cdot)$ satisfying $\phi_R$ correspond one-to-one in the following manner:

(i) $\beta_{2m+i} = \begin{cases} 1 & \text{if } v(x_i) = T \\ 0 & \text{if } v(x_i) = F \end{cases} \qquad i = 1,2,\ldots,\ell$

(ii) $\beta_{2k-1} + 2\beta_{2k}$ is the binary representation of the number of literals in the disjunctive clause $\sigma_k$ which are assigned $T$ by $v(\cdot)$, minus one if $\sigma_k$ occurs in $\phi_R$.

(iii) $\beta = 0$

Satisfaction of all clauses in $\phi_R$ is guaranteed by the fact that $\beta_{2k-1} + 2\beta_{2k} \ge 0$.

To conclude the proof, we note that the originally given propositional formula $\phi$ is satisfiable if and only if $\phi_R$ is satisfiable. Thus the algorithm given is correct.

## 4.4 <u>Comments on the Reduction</u>

It is of broader interest to clarify the basic idea of the reduction algorithm from which a general method for reducing computational problems to diophantine equations by deterministic computation can be derived. The crucial elements are contained in the system of definitions in the algorithm, and Lemma 2 of the above proof. A version of this pared to bare essentials may be obtained by picking $p_1,\ldots,p_n$ any sufficiently large primes, $K$, $m_j$, $\bar{m}_j$ as above, and for all $j$, $\lambda_j$ minimal such that

$$\begin{cases} \lambda_j\bar{m}_j \equiv 2^{i-1} \bmod 2^n \\ \lambda_j\bar{m}_j \equiv 0 \quad \bmod P_j \end{cases}$$

$$H = \sum \lambda_j\bar{m}_j$$

We then obtain

<u>Lemma 4</u> (Conversion Lemma). For any $\alpha \in \omega$, i.e. $\alpha = \sum_{i=0}^{n-1} \alpha_i 2^i + \bar{\alpha}2^n$, $\alpha_i \in \{0,1\}$, $\bar{\alpha} \in \omega$ the unique $a \in \omega$ satisfying

(i) $a < K$

(ii) $a \equiv \alpha \bmod 2^n$

(iii) $a(a-H) \equiv 0 \bmod K$

is

$$a = \sum_{j=1}^{n} \alpha_{j-1}\lambda_j\bar{m}_j \ ,$$

$\alpha_j$ coefficients of binary representation of $\alpha$ above.

The proof of Lemma 4 is of course analogous to that of Lemmas 2 and 3 above.

The crucial idea is to provide a meansof going back and forth between the representation of a sequence as the digits of a number in some base $b$ (e.g. 2), and as the residues of a number with respect to a system of relatively prime moduli. The first allows a global description of the shifting of the sequence, the second allows global formulation (i.e., for the whole sequence in a single congruence) of a condition on the individual elements of the sequence. All known reductions of recognition of correct Turing machine computation to diophantine equations with a number of variables independent of input size provide some means of reconciling these very different kinds of operations on the sequence studied; doing so is the fundamental problem of such relations and the principal challenge in obtaining tight bounds on the complexity of diophantine decision problems.

## 5. OPEN PROBLEMS

1. (The extent of the class $D^{2(2)}$.)

Consider the following sets:

$S_1 = \{\alpha \mid \exists x,y \in \omega\colon\ y^2 - \alpha x^2 = 1\}$

$S_2 = \{\alpha \mid \alpha \text{ composite}\}$

$S_3 = \{\langle\alpha,\beta\rangle \mid \exists z \in \omega\colon\ 1 < z < \beta \text{ and } z \text{ divides } \alpha\}$

$S_4 = \{\langle\alpha,\beta,\gamma\rangle \mid \exists y,z \in \omega\colon\ \alpha y^2 + \beta z - \gamma = 0\}$

All of these are in $D^{2(2)}$. But also:

$S_1$ is in P (see [12]).

$S_2$ is in P, if the Extended Riemann Hypothesis is true (see [10], [11]).

$S_3$ is in $NP \cap NP^c$ (see [11]). The problem of accepting $S_3$ is computationally equivalent to factorization ([11]) and hence probably not in P.

$S_4$ is NP-complete.

These examples illustrate that $D^{2(2)}$ is a microcosm of the principle subclasses of NP. This suggests that a more detailed determination of the extent of $D^{2(2)}$ would be valuable. Outstanding open problems are:

(a) Show that $D^{2(2)} \neq NP$.

(b) Show that $P \not\subseteq D^{2(2)}$.

On the other hand, it has been shown [14] by methods of algebraic topology that $S_2$, and its complement Pr, the set of primes, are not in $D^1$. Hence $D^{2(2)} \neq D^1$.

2. (Complete sets in $NP \cap NP^c$.)

If $S \in NP \cap NP^c$, i.e. $S \in NP$ and $S^c \in NP$, then the reduction algorithm described in the proof of Theorems 1 and 2 reduces the questions

$$x \in S\ ?$$
$$x \in S^c\ ?$$

to the question of the solvability of two very closely related diophantine equations. This fact may be of help in studying the class $NP \cap NP^c$, e.g. in answering the questions:

(a) Is there a complete set in $NP \cap NP^c$ (i.e. a set in $NP \cap NP^c$ to which every set in $NP \cap NP^c$ can be reduced by a deterministic polynomial time algorithm)?

(b) Is the set $S_3$ of Problem 1 above a complete set in $NP \cap NP^c$?

3. (Further classification of the set Pr of prime numbers.)

By Pratt [13], Pr $\in$ NP; hence in $NP \cap NP^c$ We can now ask:

(a) Is Pr $\in$ D?

(b) Is Pr NP(D)-complete?

4. What is the relationship of $D \cap D^c$ to P?

It is known [2] that $(D^1)^c \subset D$, so that $D^1 \subseteq D \cap D^c \cap P$.

5. Under what assumptions short of $P = D$ would $\mathrm{Th}(\langle NP, \leq_D\rangle)$ be equivalent to $\mathrm{Th}(\langle NP, \leq_P\rangle)$?

If $D \neq NP$, are there D-(reducibility) degrees between 0 (the degree of sets in D) and the degree of the NP(D)-complete set of Theorem 4? See, for the corresponding assertion about P-reducibility, Ladner [6].

6. Show that $D = NP$.

7. Is the $NP \overset{?}{=} D$ question independent of reasonable axioms for first-order number theory, e.g. "Peano Arithmetic"?

This is an extremely difficult problem. We hope that Theorems 1 and 2 will be of help in resolving this; they present the $NP = P$ question in simple, strictly number-theoretic terms (see Gill [4]).

## REFERENCES

[1] Adleman, L. and Manders, K., "Computational Complexity of Decision Procedures for Polynomials," 16th Annual Symposium on Foundations of Computer Science (Oct. 1975), 169-177.

[2] Adleman, L. and Manders, K., "Number-theoretic Aspects of Computational Complexity." In preparation.

[3] Cook, S.A., "The Complexity of Theorem Proving Procedures," Conf. Rec. 3rd ACM Symposium on Theory of Computing (1971), 151-158.

[4] Gill, J., "Axiomatic Independence of the Question NP = P?." To appear.

[5] Karp, R.M., "Reducibility Among Combinatorial Problems," in *Complexity of Computer Computation*, eds. R.N. Miller and J.W. Thatcher, Plenum Press, 1972, pp. 85-104.

[6] Ladner, R., "On the Structure of Polynomial Time Reducibility," *Journal of the Association for Computing Machinery* **22**, 1 (Jan. 1975).

[7] Matijasevič, Y., "Enumerable Sets are Diophantine (Russian)," *Dokl. Acad. Nauk SSSR* **191** (1970), 279-282.

[8] Matijasevič, Y. and Robinson, J., "Reduction of an Arbitrary Diophantine Equation to One in 13 Unknowns," *Acta Arithmetica* **27** (1975), 521-553.

[9] Matijasevič, Y. Private communication.

[10] Miller, G.L., "Riemann's Hypothesis and Tests for Primality," *7th ACM Symposium on Theory of Computing* (1975), 234-239.

[11] Miller, G.L. Ph.D. Thesis, Berkeley, 1975.

[12] Niven, I. and Zuckerman, H. *An Introduction to the Theory of Numbers*, John Wiley and Sons, Inc. (1972).

[13] Pratt, V., "Succinct Certificates for the Primes." To appear.

[14] Sato, D. Private communication.