SIAM J. ALG. DISC. METH.
Vol. 8, No. 4, October 1987

020

SOME COMPLETENESS RESULTS ON DECISION TREES AND GROUP TESTING*

DING-ZHU DU† AND KER-I KO‡

Abstract. The computational complexity of the group testing problem is investigated under the minimax measure and the decision tree model. We consider the generalizations of the group testing problem in which partial information about the decision tree of the problem is given. Using this approach, we demonstrate the NP-hardness of several decision problems related to various models of the group testing problem. For example, we show that, for several models of group testing, the problem of recognizing a set of queries that uniquely determines each object is co-NP-complete.

Key words. group testing, decision trees, NP-completeness, #P-completeness

AMS(MOS) subject classifications. 68Q25, 68P10

1. Introduction. Many combinatorial search problems involve the minimization of the heights of decision trees. Such problems can often be described as two-person query games, where one player A selects an object x from a finite domain D and assumes the role of an oracle while the other player B tries to identify the object x by making queries to A about the object. Consider, as an example, the problem of group testing [3], [7], [14], [23]–[26], [29]. The domain of the problem is the set $\mathcal{S}_{n,d}$ of all subsets of $\{1, \cdots, n\}$ that have size d. The player B tries to identify a set $S \in \mathcal{S}_{n,d}$ by making queries about S. Each query is a subset $T \subseteq \{1, \cdots, n\}$ and its answer, provided by A, is either "YES" if the intersection $S \cap T$ is nonempty, or "NO" otherwise. As another example, we may consider the problem of sorting by decision tree as a two-person query game [16], in which a domain consists of all permutations over $\{1, \cdots, n\}$ and, to identify a permutation α, queries of the form "$\alpha(i) < \alpha(j)$?" may be asked. An algorithm for such a search problem is essentially a general procedure to produce, for each domain, a decision tree of which each path uniquely determines an object in the domain. An optimal algorithm is one which produces, for each domain, a decision tree of the minimum height. For example, for the problem of group testing, a decision tree may be described as follows: Each node of the tree is a subset $T \subseteq \{1, \cdots, n\}$, and has two children, identified by answers YES and NO to the query T. Each path of the tree consists of a sequence of queries $(T_1, \cdots, T_m)$ with their answers $(a_1, \cdots, a_m)$ such that there is exactly one $S \in \mathcal{S}_{n,d}$ having the property that $S \cap T_i$ is nonempty if and only if a_i = YES for $i = 1, \cdots, m$.

Except for a very few simple search problems, the problem of finding an optimal algorithm for a shortest decision tree problem appears to be intractable. For example, in spite of extensive studies, the optimal algorithms for sorting and group testing problems remain as open questions (cf. [20]). For the group testing problems, people have conjectured that they are indeed intractable [12]; however, no formal proofs for these conjectures have been found.

* Received by the editors May 27, 1986; accepted for publication July 23, 1987. A preliminary version of this paper was presented at the Third SIAM Conference on Discrete Mathematics, Clemson University, Clemson, South Carolina, May 1986.

† Mathematical Sciences Research Institute, Berkeley, California 94720. Current address, Institute of Applied Mathematics, Academia Sinica, Beijing, China.

‡ Department of Mathematics, University of California at Santa Barbara, Santa Barbara, California 93106. Current address, Department of Computer Science, SUNY at Stony Brook, Stony Brook, New York 11794.
The research of this author was supported in part by the National Science Foundation under grants DCR83-12472 and DCR-8501226.

In the study of computational complexity of combinatorial optimization problems, a search problem is usually formulated as a decision problem so that the lower bound results are easier to be developed (often through the reductions from known NP- or PSPACE-complete problems). For the shortest decision tree problem, the associated decision problem may be formulated as follows:

> Given a domain D and an integer k, determine whether there is a decision tree of height $\leqq k$ of which each path uniquely determines an object in D.

It is not hard to see that the above problem is often solvable in polynomial space. (For given D and k, we may guess nondeterministically a decision tree of height k and verify that for each of its path, there is only one object consistent with the queries and answers of this path. Note that at any step of the computation, this algorithm needs only $O(k)$ space to store one path of the decision tree, although the complete tree contains about 2^k many nodes.) On the other hand, the domain of the problem often has a very simple form so that it is difficult to obtain a reduction from other (PSPACE-)complete problems to it since such a reduction would usually require rich structures in the problem in question (cf. [5], [8]). Indeed, it follows from the research in abstract complexity theory that if the input to a problem may be defined by two integers (here, n and d), then the problem cannot be PSPACE-complete unless P = PSPACE [6]. So, in order to obtain any completeness results on the shortest decision tree problems, we must reformulate the problems to add more complex structures to the problem instances. A general approach to this is to treat the problem as a special case of a more general problem whose problem instances take more general forms. For instance, Even and Tarjan [5] have extended the game Hex to general graphs and showed that the generalized Hex game, or the Shannon switching game on vertices, is PSPACE-complete, while the complexity of the more common version of Hex remained open. In this paper, we follow this approach to the group testing problem and demonstrate several completeness results on the generalized group testing problem.

We first introduce some terminologies about two-person query games. A *query history* is a set of queries together with their answers. The *solution space* associated with a query history H is the set of all objects in the domain which are *consistent* with the query history H. In other words, let $\mathrm{ANS}_x(y)$ denote the answer given by player A to the query y when x is the object to be identified. Then, the solution space associated with a query history $H = \{(y_1, a_1), \cdots, (y_m, a_m)\}$, where y_i's are queries and a_i's are corresponding answers, is the set $\{x \in \text{domain} \mid \mathrm{ANS}_x(y_i) = a_i \text{ for } i = 1, \cdots, m\}$. The *initial solution space* is simply the given domain. A shortest decision tree problem may thus be rephrased as the problem of using the minimum number of queries to reduce the solution space from the given domain to a singleton space.

We note that while the initial solution spaces often have simple structures, the solution spaces associated with arbitrary query histories may have complex structures. For example, it was pointed out in [18] that many researchers have conjectured that, for the sorting problem, the problem of determining the size of the solution space associated with a query history is #P-complete. The first two problems considered in this paper are concerned with the structure of the general solution spaces associated with given query histories. The first asks whether a given query history is consistent (or, whether the player A has been cheating), and the second asks what the size of the solution space associated with a given query history is.

CONSISTENCY PROBLEM. Given a domain D and a query history H, determine whether the query history H is consistent; i.e. whether the solution space associated with H is nonempty.

COUNTING PROBLEM. Given a domain D and a query history H, determine the size of the solution space associated with H.

Our third problem is concerned with the nonadaptive query games. In a nonadaptive query game, the player B must present a set of queries before he/she gets any answer from the player A [15]. Again, the goal here is to find a smallest set of queries which uniquely determines each object in the domain. The following problem asks a simpler recognition question of such a *determinant* set of queries.

DETERMINACY PROBLEM. Given a domain D and a set Q of queries, determine whether each set of answers to the queries in Q uniquely determines an object in the domain.

We will study the above questions in the context of the group testing problem. We consider several variations of the original group testing problem, derived from different domains and different oracles. In the following, for each set S, let $|S|$ denote the size of S; for each n and d, let $\mathcal{S}_n$ denote the set of all subsets of $\{1, \cdots, n\}$ and $\mathcal{S}_{n,d}$ the set of all sets S in $\mathcal{S}_n$ with $|S| = d$. For each pair of objects x and y, $\mathrm{ANS}_x(y)$ is the answer given by player A to query y when x is the object to be identified.

MODEL A_k ($k \geqq 1$). Given a domain $\mathcal{S}_n$ and an answering function ANS_S (as the oracle) of the type

$$\mathrm{ANS}_S(T) = \begin{cases} i & \text{if } |S \cap T| = i < k, \\ k & \text{if } |S \cap T| \geqq k, \end{cases}$$

determine the set S.

MODEL A'_k ($k \geqq 1$). Given a domain $\mathcal{S}_{n,d}$ and an answering function ANS_S of the same type as in Model A_k, determine the set S.

MODEL B. Given a domain $\mathcal{S}_n$ and an answering function ANS_S (as the oracle) of the type

$$\mathrm{ANS}_S(T) = \begin{cases} 0 & \text{if } S \cap T = \varnothing, \\ 1 & \text{if } S \cap T \neq \varnothing \text{ and } \bar{S} \cap T \neq \varnothing, \\ 2 & \text{if } \bar{S} \cap T = \varnothing, \end{cases}$$

where $\bar{S} = \{1, \cdots, n\} - S$, determine the set S.

MODEL B'. Given a domain $\mathcal{S}_{n,d}$ and an answering function ANS_S of the same type as in Model B, determine the set S.

MODEL C. Given a domain $\mathcal{S}_n$ and an answering function ANS_S of the type

$$\mathrm{ANS}_S(T) = |S \cap T|,$$

determine the set S.

MODEL C'. Given a domain $\mathcal{S}_{n,d}$ and an answering function ANS_S of the same type as in Model C, determine the set S.

We remark that Models A_1 and A'_1 are the original group testing problems [3], [7], [14], [23]–[26], [29]; Models A_k and A'_k, with $k > 1$, have been considered in [2], [9], [11], [13], [21], [27]; Models B and B' have been considered in [10]; and Models C and C' are a classical combinatorial search problem [1], [4], [19].

The main results of this paper may be summarized as follows.

THEOREM 1. (a) *The consistency problem for Model A_1 is polynomial time solvable.*

(b) *The consistency problems for all other models (i.e., for Models A_k, $k > 1$, for Models A'_k, $k \geqq 1$, and for Models B, B', C and C') are* NP-*complete.*

THEOREM 2. *The counting problems for all models (i.e., for Models A_k and A'_k, $k \geqq 1$, and for Models B, B', C and C') are* #P-*complete.*

THEOREM 3. (a) *The determinacy problems for all models are in co*-NP.

(b) *The determinacy problem for Model* A_1 *is polynomial time solvable.*

(c) *The determinacy problem for Models* A_k, A'_k, $k \geqq 4$, *and for Models* B, B', C *and* C' *are co*-NP-*complete.*

The question of whether the determinacy problems for Models A_k, $k = 2, 3$, and for Models A'_k, $k \leqq 3$, are co-NP-complete remains open.

2. Consistency problems. We first restate the consistency problems for the models of group testing problem defined in § 1. In the following, Consistency-X denotes the consistency problem for Model X, where $X \in \{A_k, A'_k, B, B', C, C' | k \geqq 1\}$.

CONSISTENCY-X. Given an integer n (or, two integers n and d) and a set $H = \{(T_j, a_j) | j = 1, \cdots, m\}$, with $T_j \in \mathcal{S}_n$, $a_j \in \{0, 1, \cdots, n\}$ for $j = 1, \cdots, m$, determine whether the set $C = \{S \in \mathcal{S}_n \text{ (or, } \mathcal{S}_{n,d}) | \mathrm{ANS}_S(T_j) = a_j, j = 1, \cdots, m\}$ is nonempty.

It is interesting to observe the similarity between the group testing problem and the satisfiability problem (SAT) [8], where each query of the group testing problem may be regarded as a clause of variables for SAT. Therefore, our main tools for proving Theorems 1, 2 and 3 are variations of the satisfiability problem. For the proof of Theorem 1, we will use the following NP-complete problems.

VERTEX-COVER. Given a graph $G = (V, E)$ and an integer $k \leqq |V|$, determine whether there is a set $V' \subseteq V$ of size k such that each edge $e \in E$ is incident on some $v \in V'$.

ONE-IN-THREE-SAT. Given a set U of variables and a set $\mathscr{C}$ of clauses, with each $C \in \mathscr{C}$ containing exactly three variables from U, determine whether there is a truth assignment t on U such that each clause C in $\mathscr{C}$ contains exactly one TRUE variable.

NOT-ALL-EQUAL-SAT. Given U and $\mathscr{C}$ as in One-in-three-SAT, determine whether there is a truth assignment t on U such that each clause C in $\mathscr{C}$ contains at least one TRUE variable and at least one FALSE variable.

Remark. The original versions of One-in-three-SAT and Not-all-equal-SAT, as stated in [8], allow a clause C in $\mathscr{C}$ to contain both negated and nonnegated literals. The NP-completeness of our versions stated above can easily be proved from Schaefer's proof of the NP-completeness of the Generalized-SAT problem [22].

Now we apply these NP-complete problems to prove Theorem 1.

MODEL A_1. Let an instance $(n, H = \{(T_j, a_j) | j = 1, \cdots, m\})$ of Consistency-A_1 be given, where for each $j = 1, \cdots, m$, $T_j \in \mathcal{S}_n$ and $a_j \in \{0, 1\}$. Define

$$I = \{j | 1 \leqq j \leqq m, a_j = 0\},$$

and $J = \{j | 1 \leqq j \leqq m, a_j = 1\}$. Also let $X = \bigcup_{j \in I} T_j$ and $Y = \{1, \cdots, n\} - X$. Then, it is easy to check that

$$H \text{ is consistent iff for each } j \in J, T_j \cap Y \neq \varnothing.$$

This characterization of consistent query histories provides a simple polynomial-time algorithm for Consistency-A_1.

MODEL A_k, $k > 1$. We show that if $k > 1$, then One-in-three-SAT is polynomial-time reducible to Consistency-A_k.

Let an instance $(U, \mathscr{C})$ of One-in-three-SAT be given, where $U = \{x_1, \cdots, x_p\}$, $\mathscr{C} = \{C_1, \cdots, C_q\}$, $C_j \subseteq U$ and $|C_j| = 3$, for $j = 1, \cdots, q$. Define an instance

$$(n, H = \{(T_j, a_j) | j = 1, \cdots, m\})$$

of Consistency-A_k as follows:

$$n := p;\ m := q;$$

$$\text{for each } j = 1, \cdots, m, \text{ let } T_j := \{i | x_i \in C_j\} \text{ and } a_j := 1.$$

For each assignment t on U, let $S_t := \{i | t(x_i) = \text{TRUE}\}$. Then, the mapping from t to S_t is a natural one-to-one correspondence between the set of truth assignments on U and the set $\mathscr{S}_n$. Furthermore, a truth assignment t on U assigns exactly one TRUE variable to each clause in $\mathscr{C}$ if and only if $|S_t \cap T_j| = 1$ for all $j = 1, \cdots, m$. In other words, the instance $(U, \mathscr{C})$ has a solution (for the problem One-in-three-SAT) if and only if the instance (n, H) has a solution (for the problem Consistency-A_k). This completes the proof.

MODEL A'_1. We show that the problem Vertex-Cover is polynomial-time reducible to Consistency-A'_1.

Let (G, k) be a given instance of Vertex-Cover, where $G = (V, E)$ is a graph with vertex set $V = (v_1, \cdots, v_p\}$ and the edge set $E = \{e_1, \cdots, e_q\}$, and k is an integer less than or equal to p. Define an instance $(n, d, H = \{(T_j, a_j) | j = 1, \cdots, m\})$ of Consistency-A'_1 as follows.

$$n := p;\ m := q;\ d := k;$$

$$\text{for each } j = 1, \cdots, m, \text{ let } T_j := \{i | v_i \in e_j\} \text{ and } a_j := 1.$$

For each $V' \subseteq V$, define a set $S_{V'} \in \mathscr{S}_n$ by $S_{V'} = \{i | v_i \in V'\}$. Then, this is a one-to-one correspondence between subsets of V of size k and sets in $\mathscr{S}_{n,d}$. Furthermore, V' is a vertex cover of E if and only if $S_{V'} \cap T_j \neq \varnothing$ for all $j = 1, \cdots, m$. This shows that the mapping from (G, k) to (n, d, H) is a reduction from Vertex-Cover to Consistency-A'_1.

MODEL A'_k, $k > 1$. We show that if $k > 1$, then Consistency-A'_1 is polynomial-time reducible to Consistency-A'_k.

For a given instance $(n, d, H = \{(T_j, a_j) | j = 1, \cdots, m\})$ of Consistency-A'_1, define an instance $(n', d', H' = \{(T'_j, a'_j) | j = 1, \cdots, m\})$ of Consistency-A'_k as follows:

$$n' := n + k - 1;\ m' := m + k - 1;\ d' := d + k - 1;$$

$$\text{for each } j = 1, \cdots, m,$$

$$\text{if } a_j = 0 \text{ then let } T'_j := T_j \text{ and } a'_j := 0,$$

$$\text{if } a_j = 1 \text{ then let } T'_j := T_j \cup \{n+1, \cdots, n+k-1\} \text{ and } a'_j := k;$$

$$\text{for each } j = m+1, \cdots, m+k-1, \text{ let } T'_j := \{n + j - m\} \text{ and } a'_j := 1.$$

Assume that (n, d, H) is consistent for Model A'_1 and $S \in \mathscr{S}_{n,d}$ satisfies the condition that for all $j = 1, \cdots, m$, $S \cap T_j \neq \varnothing$ if and only if $a_j = 1$. Define

$$S' = S \cup \{n+1, \cdots, n+k-1\}.$$

Then, $S \in \mathscr{S}_{n',d'}$. Also, for all $j = 1, \cdots, m$,

$$\text{if } a_j = 0, \text{ then } |S' \cap T'_j| = |S \cap T_j| = 0 = a'_j, \quad \text{and}$$

$$\text{if } a_j = 1, \text{ then } |S' \cap T'_j| = |S \cap T_j| + (k-1) \geqq k = a'_j;$$

and, for all $j = m + 1, \cdots, m + k - 1$,

$$|S' \cap T'_j| = 1 = a'_j.$$

So, (n', d', H') is consistent for Model A'_k.

Conversely, if (n', d', H') is consistent for Model A'_k, then there is a set

$$S' \subseteq \{1, \cdots, n+k-1\}$$

such that $\text{ANS}_{S'}(T'_j) = a'_j$ for $j = 1, \cdots, m + k - 1$, where the answering function

$\mathrm{ANS}_{S'}$ is of the type of Model A'_k. Let $S = S' \cap \{1, \cdots, m\}$. We claim that $S \cap T_j = \emptyset$ if and only if $a_j = 0$ for all $j = 1, \cdots, m$.

First, if $a_j = 0$, then $T'_j = T_j$ and $a'_j = 0$. So, $S' \cap T'_j = \emptyset$ and hence $S \cap T_j = \emptyset$. Next, if $a_j = 1$, then $\mathrm{ANS}_{S'}(T'_j) = a'_j = k$ implies $|S' \cap T'_j| \geqq k$. Since

$$|S' \cap \{n+1, \cdots, n+k-1\}| \leqq k-1, \qquad |S \cap T_j| = |S' \cap T'_j \cap \{1, \cdots, n\}| \geqq 1.$$

This completes the proof for Model A'_k, $k > 1$.

MODEL B. We show that Not-all-equal-SAT is polynomial-time reducible to Consistency-B. The reduction is similar to the reduction from One-in-three-SAT to Consistency-A_2.

Let an instance $(U, \mathscr{C})$ of Not-all-equal-SAT be given, where $U = \{x_1, \cdots, x_p\}$, $\mathscr{C} = \{C_1, \cdots, C_q\}$, $C_j \subseteq U$ and $|C_j| = 3$, for $j = 1, \cdots, q$. Define an instance

$$(n, H = \{(T_j, a_j) \mid j = 1, \cdots, m\})$$

of Consistency-B as follows:

$$n := p;\ m := q;$$

$$\text{for each } j = 1, \cdots, m, \text{ let } T_j := \{i \mid x_i \in C_j\} \text{ and } a_j := 1.$$

Similarly to the reduction from One-in-three-SAT to Consistency-A_2, there is a natural one-to-one correspondence between the set of truth assignments on U and the set $\mathscr{S}_n$. Furthermore, a truth assignment t on U assigns at least one TRUE variable and at least one FALSE variable to each clause in $\mathscr{C}$ if and only if $1 \leqq |S_t \cap T_j| \leqq 2$ for all $j = 1, \cdots, m$, where S_t is the set in $\mathscr{S}_n$ corresponding to t. This shows that the mapping defined above is a reduction from Not-all-equal-SAT to Consistency-B.

MODEL B'. We show that Vertex-Cover is polynomial-time reducible to Consistency-B'.

Let (G, k) be a given instance of Vertex-Cover, where $G = (V, E)$ is a graph with the vertex set $V = \{v_1, \cdots, v_p\}$ and the edge set $E = \{e_1, \cdots, e_q\}$, and k is an integer less than or equal to p. Define an instance $(n, d, H = \{(T_j, a_j) \mid j = 1, \cdots, m\})$ of Consistency-B' as follows:

$$n := p + q;\ m := q;\ d := k;$$

$$\text{for each } j = 1, \cdots, m, \text{ assume that } e_j = \{v_{j_1}, v_{j_2}\}, \text{ and}$$

$$\text{let } T_j := \{j_1, j_2, p + j\} \text{ and } a_j := 1.$$

Let $V' \subseteq V$ be a vertex cover for G of size k. Then the set

$$S_{V'} = \{i \mid 1 \leqq i \leqq p, v_i \in V'\}$$

has the property $1 \leqq |S_{V'} \cap T_j| \leqq 2$ for all $j = 1, \cdots, m$. Also, $|S_{V'}| = d$. So, (n, d, H) is consistent.

Conversely, let $S \subseteq \{1, \cdots, n\}$, $|S| = d$, be a solution to the instance (n, d, H). Define $V' := \{v_i \mid i \in S, 1 \leqq i \leqq p\} \cup \{v_{j_1} \mid p + j \in S\}$. Then, $|V'| \leqq d = k$ because $|S| = d$. Also, V' is a vertex cover for G: for each $j = 1, \cdots, q$, if $p + j \notin S$ then j_1 or j_2 is in S and hence v_{j_1} or v_{j_2} is in V'; if $p + j \in S$, then $v_{j_1} \in V'$. This completes the proof.

MODEL C. The reduction from One-in-three-SAT to Consistency-A_2 is actually also a reduction from One-in-three-SAT to Consistency-C, because the output instances from the reduction always have $a_j = 1 < 2$.

MODEL C'. We show that Consistency-C is polynomial-time reducible to Consistency-C'.

Let an instance $(n, H = \{(T_j, a_j) | j = 1, \cdots, m\})$ of Consistency-C be given. Define an instance $(n', d', H' = \{(T'_j, a'_j) | j = 1, \cdots, m\})$ of Consistency-C' as follows:

$n' := 2n$; $d' := n$; $m' := n + m$;

for $j = 1, \cdots, m$, let $T'_j := T_j$ and $a'_j := a_j$, and

for $j = m+1, \cdots, m+n$, let $T'_j := \{j-m, n+j-m\}$ and $a_j := 1$.

If $S \in \mathscr{S}_n$ is consistent with H, define $S' = S \cup \{k + n | 1 \leqq k \leqq n, k \notin S\}$. Then, $S' \in \mathscr{S}_{n',d'}$, and

$$|S \cap T_j| = |S' \cap T'_j| \quad \text{for } j = 1, \cdots, m,$$

$$|S' \cap T'_j| = 1 \quad \text{for } j = m+1, \cdots, m+n.$$

This shows that S' is consistent with H'.

Conversely, if $S' \in \mathscr{S}_{n',d'}$ is consistent with H', then $S = S' \cap \{1, \cdots, n\}$ satisfies the condition that for all $j = 1, \cdots, m$, $|S \cap T_j| = |S' \cap T'_j|$, because for all $j = 1, \cdots, m$, $T'_j \subseteq \{1, \cdots, n\}$ and so $|S' \cap T'_j| = |S' \cap T'_j \cap \{1, \cdots, n\}| = |S \cap T_j|$. This completes the proof.

3. Counting problems. We restate the counting problems for Model X, where $X \in \{A_k, A'_k, B, B', C, C' | k \geqq 1\}$.

COUNTING-X. Given an integer n (or, two integers n and d) and a set $H = \{(T_j, a_j) | j = 1, \cdots, m\}$, with $T_j \in \mathscr{S}_n$, $a_j \in \{0, 1, \cdots, n\}$ for $j = 1, \cdots, m$, determine the size of the set $C = \{S \in \mathscr{S}_n \text{ (or, } \mathscr{S}_{n,d}) | \text{ANS}_S(T_j) = a_j, j = 1, \cdots, m\}$.

It is easy to see that for any model X, the problem Counting-X is in #P because the problem Consistency-X is in NP. (For the definitions of the class #P and #P-completeness, see [8] and [28].) In this section, we show that the counting problems for all models are #P-complete. We remark that this type of #P-completeness results has been conjectured for the sorting problem [18] and has been proved for a simplified Mastermind game [17]. The following #P-complete problems will be used in the proof of Theorem 2.

MONOTONE-#2SAT. Given a set U of variables and a set $\mathscr{C}$ of clauses, with each $C \in \mathscr{C}$ containing exactly two variables from U, determine the number of truth assignments t on U such that each clause C in $\mathscr{C}$ contains at least one TRUE variable.

ONE-IN-THREE-#SAT. Given $(U, \mathscr{C})$ as in One-in-three-SAT, determine the number of solutions to $(U, \mathscr{C})$.

NOT-ALL-EQUAL-#SAT. Given $(U, \mathscr{C})$ as in Not-all-equal-SAT, determine the number of solutions to $(U, \mathscr{C})$.

Monotone-#2SAT has been shown in [28] to be #P-complete. We first establish the #P-completeness of One-in-three-#SAT and Not-all-equal-#SAT. We note that a counting problem is not a decision problem and hence the polynomial-time many-one reductions are not necessarily applicable to them. Instead, the polynomial-time Turing reductions are usually used to prove the #P-completeness results, although the notion of many-one reductions preserving the number of solutions (or, *parsimonious reductions*) does provide a stronger definition of #P-completeness (cf. [8]). In this section, we refer to #P-completeness as the one with respect to the polynomial-time Turing reductions.

LEMMA 1. *One-in-three-*#SAT *is* #P-*complete.*

Proof. The fact that One-in-three-#SAT is in #P is clear. We show that Monotone-#2SAT is polynomial-time Turing reducible to One-in-three-#SAT.

Let an instance $(U, \mathscr{C})$ of Monotone-#2SAT be given, where $U = \{x_1, \cdots, x_p\}$, $\mathscr{C} = \{C_1, \cdots, C_q\}$ and for each $j = 1, \cdots, q$, $C_j \subseteq U$ and $|C_j| = 2$. Define an instance

$(V, \mathscr{D})$ of One-in-three-#SAT as follows:

$$V := U \cup \{u_j, v_j, w_j \mid j = 1, \cdots, q\} \cup \{y_1, y_2, y_3, z\};$$

for each $j = 1, \cdots, q$, assume that $C_j = \{x_{j_1}, x_{j_2}\}$ and let

$$C_{j,1} := \{x_{j_1}, u_j, y_1\}, C_{j,2} := \{x_{j_2}, v_j, y_1\}, C_{j,3} := \{u_j, v_j, w_j\};$$

let $D_1 := \{y_1, y_2, z\}$, $D_2 := \{y_2, y_3, z\}$, $D_3 := \{y_1, y_3, z\}$;

$$\mathscr{D} = \{C_{j,k} \mid j = 1, \cdots, q; k = 1, 2, 3\} \cup \{D_1, D_2, D_3\}.$$

Assume that t is a truth assignment on U such that for each $j = 1, \cdots, q$, there is a variable x_k in C_j with $t(x_k) = \text{TRUE}$. Define a truth assignment t' on V as follows:

for each $i = 1, \cdots, p$, $t'(x_i) = t(x_i)$;
$t'(y_1) = t'(y_2) = t'(y_3) :=$ FALSE; $t'(z) :=$ TRUE;
for each $j = 1, \cdots, q$, assuming that $C_j = \{x_{j_1}, x_{j_2}\}$,
Case 1. if $t(x_{j_1}) =$ TRUE, $t(x_{j_2}) =$ FALSE
then $t'(u_j) = t'(w_j) :=$ FALSE and $t'(v_j) :=$ TRUE;
Case 2. if $t(x_{j_1}) =$ FALSE, $t(x_{j_2}) =$ TRUE
then $t'(v_j) = t'(w_j) :=$ FALSE and $t'(u_j) :=$ TRUE;
Case 3. if $t(x_{j_1}) = t(x_{j_2}) =$ TRUE
then $t'(u_j) = t'(v_j) :=$ FALSE and $t'(w_j) :=$ TRUE.

It is easy to check that t' assigns the value TRUE to exactly one variable in each clause in $\mathscr{D}$. Therefore, each solution t of $(U, \mathscr{C})$ is mapped to a solution t' of $(V, \mathscr{D})$, and the mapping is one-to-one.

Furthermore, we note that if t'' is a solution of $(V, \mathscr{D})$ then, to assign exactly one TRUE value to each of D_1, D_2 and D_3, t'' must assign TRUE to z and FALSE to y_1, y_2, y_3. Furthermore, for each $j = 1, \cdots, q$, t'' cannot assign the value TRUE to both u_j and v_j; this implies that one of x_{j_1} and x_{j_2} must be TRUE. Finally, for each $j = 1, \cdots, q$, if two solutions t_1 and t_2 of $(U, \mathscr{C})$ agree at x_{j_1} and x_{j_2} and $t_1(y_1) = t_2(y_1) =$ FALSE, then they must agree at u_j, v_j and w_j. The above observations show that the mapping defined above (from t to t') is a bijection between the solutions of $(U, \mathscr{C})$ and the solutions of $(V, \mathscr{D})$. This completes the proof. □

LEMMA 2. *Not-all-equal-*#SAT *is* #P-*complete.*

Proof. Again, it is clear that Not-all-equal-#SAT is in #P, and we show that Monotone-#2SAT is polynomial-time Turing reducible to Not-all-equal-#SAT.

Let an instance $(U, \mathscr{C})$ of Monotone-#2SAT be given, where $U = \{x_1, \cdots, x_p\}$, $\mathscr{C} = \{C_1, \cdots, C_q\}$ and for each $j = 1, \cdots, q$, $C_j \subseteq U$ and $|C_j| = 2$. Define an instance $(V, \mathscr{D})$ of Not-all-equal-#SAT as follows:

$$V := U \cup \{u_j, v_j \mid j = 1, \cdots, q\} \cup \{y_1, y_2, y_3, z\};$$

for each $j = 1, \cdots, q$, assume that $C_j = \{x_{j_1}, x_{j_2}\}$ and let

$$C_{j,1} := \{x_{j_1}, u_j, z\}, C_{j,2} := \{x_{j_2}, v_j, z\},$$

$$C_{j,3} := \{x_{j_1}, x_{j_2}, u_j\}, C_{j,4} := \{u_j, v_j, y_1\},$$

$$C_{j,5} := \{u_j, v_j, y_2\}, C_{j,6} := \{u_j, v_j, y_3\};$$

let $D := \{y_1, y_2, y_3\}$;

$$\mathscr{D} = \{C_{j,k} \mid j = 1, \cdots, q; k = 1, \cdots, 6\} \cup \{D\}.$$

We first note that if t' is a solution of $(V, \mathscr{D})$ (i.e., t' is a truth assignment on V such that t' assigns at least one TRUE value and at least one FALSE value to each clause in $\mathscr{D}$), then $t'(u_j) \neq t'(v_j)$ for $j = 1, \cdots, q$, because $t'(y_1)$, $t'(y_2)$ and $t'(y_3)$ cannot be all equal.

Now, assume that t is a solution of $(U, \mathscr{C})$. Define a truth assignment t' on $V - \{y_1, y_2, y_3\}$ as follows:

$t'(z) :=$ FALSE;
for each $i := 1, \cdots, p$, $t'(x_i) := t(x_i)$;
for each $j = 1, \cdots, q$, assuming that $C_j = \{x_{j_1}, x_{j_2}\}$,
Case 1. if $t(x_{j_1}) =$ TRUE, $t(x_{j_2}) =$ FALSE
then $t'(u_j) :=$ FALSE and $t'(v_j) :=$ TRUE;
Case 2. if $t(x_{j_1}) =$ FALSE, $t(x_{j_2}) =$ TRUE
then $t'(u_j) :=$ TRUE and $t'(v_j) :=$ FALSE;
Case 3. if $t(x_{j_1}) = t(x_{j_2}) =$ TRUE
then $t'(u_j) :=$ FALSE and $t'(v_j) :=$ TRUE.

We then extend t' into truth assignments on V such that $t'(y_1)$, $t'(y_2)$, $t'(y_3)$ are not all equal. There are six such extensions. It is obvious that each of these extensions is a solution of $(V, \mathscr{D})$. Next, for each of such extensions t'', define $\bar{t}''(w)$ to be the negation of $t''(w)$ for all $w \in V$. We get six more truth assignments which are solutions of $(V, \mathscr{D})$. (For the problem Not-all-equal-SAT, the negation of any solution is itself a solution.) We note that all these assignments are distinct. Furthermore, two distinct solutions t_1 and t_2 of $(U, \mathscr{C})$ define two disjoint sets of solutions of $(V, \mathscr{D})$. To see this, if a solution t_1'' of $(V, \mathscr{D})$ derived from t_1 is equal to a solution t_2'' of $(V, \mathscr{D})$ derived from t_2, then $t_1''(z) = t_2''(z)$. Hence, either $t_1 = t_1''|_U = t_2''|_U = t_2$ or $\bar{t}_1 = t_1''|_U = t_2''|_U = \bar{t}_2$, where $\bar{t}_1$ and $\bar{t}_2$ are the negations of t_1 and t_2, respectively. So, we get

$$12 \cdot (\#\text{ of solutions of } (U, \mathscr{C})) \leqq \#\text{ of solutions of } (V, \mathscr{D}).$$

Now, if t'' is a solution of $(U, \mathscr{C})$ then, as shown above, $t''(u_j) \neq t''(v_j)$ for all $j = 1, \cdots, q$. Assume that $t''(z) =$ FALSE. Then, to assign at least one TRUE value to both $C_{j,1}$ and $C_{j,2}$, at least one of $t''(x_{j_1})$ and $t''(x_{j_2})$ must be TRUE. Thus, $t''|_U$ is a solution of $(U, \mathscr{C})$, and t'' must be one of those 12 assignments defined by $t = t''|_U$. Similarly, if $t''(z) =$ TRUE, then $t = \bar{t}''|_U$ is a solution of $(U, \mathscr{C})$ and t'' is one of the 12 assignments defined by t. So, this shows that the number of solutions of $(V, \mathscr{D})$ is exactly 12 times the number of solutions of $(U, \mathscr{C})$. This completes the proof. □

With Lemmas 1 and 2, Theorem 2 is easy to prove. First, we show that for each model X, with $X \in \{A_k, B, C | k \geqq 1\}$, the problem Counting-$X$ is polynomial-time Turing reducible to Counting-X'.

LEMMA 3. *Let $X \in \{A_k, B, C | k \geqq 1\}$. Then, Counting-$X$ is polynomial-time Turing reducible to Counting-X'.*

Proof. Let (n, H) be an instance of Counting-X. Then, the number of sets S in $\mathscr{S}_n$ which are consistent with H is the sum of the number of sets S' in $\mathscr{S}_{n,d}$ which are consistent with H (with respect to the same type of answering functions) as d ranges over $\{0, \cdots, n\}$. □

MODEL A_1. We show that Monotone-#2SAT is a polynomial-time Turing reducible to Counting-A_1.

Let an instance $(U, \mathscr{C})$ of Monotone-#2SAT be given, where $U = \{x_1, \cdots, x_p\}$, $\mathscr{C} = \{C_1, \cdots, C_q\}$ and for each $j = 1, \cdots, q$, $C_j \subseteq U$ and $|C_j| = 2$. Define an instance $(n, H = \{(T_j, a_j) | j = 1, \cdots, m\})$ of Counting-A_1 as follows:

$$n := p;\ m := q;$$

$$\text{for each } j = 1, \cdots, m, \text{ let } T_j := \{i | x_i \in C_j\} \text{ and } a_j := 1.$$

Then, there is a natural one-to-one correspondence between truth assignments t on U and subsets S_t in $\mathcal{S}_n$, defined by $S_t = \{i | t(x_i) = \text{TRUE}\}$. This mapping also preserves the solutions of the two instances $(U, \mathscr{C})$ and (n, H). Thus, the number of solutions of these two instances are equal. This completes the proof.

MODEL A_k, $k > 1$ AND MODEL C. In § 2, it is proved that if $k > 1$ then One-in-three-SAT is polynomial-time (many-one) reducible to Consistency-A_k (and to Consistency-C). A close inspection of the reduction shows that the reduction actually preserves the number of solutions of the two problems. Thus, it also serves as a reduction from One-in-three-#SAT to Counting-A_k (and to Counting-C).

MODEL B. The polynomial-time (many-one) reduction from Not-all-equal-SAT to Consistency-B, as proved in § 2, also preserves the number of solutions. Thus, it also serves as a reduction from Not-all-equal-#SAT to Counting-B.

MODELS A'_k, $k \geqq 1$, MODEL B' AND MODEL C'. The #P-completeness of Counting-X', for $X \in \{A_k, B, C | k \geqq 1\}$, is established through Lemma 3 and the #P-completeness of Counting-X.

4. Determinacy problems. We restate the determinacy problems for Model X, where $X \in \{A_k, A'_k, B, B', C, C' | k \geqq 1\}$.

DETERMINACY-X. Given an integer n (or, two integers n and d) and a set $Q = \{T_j | j = 1, \cdots, m\}$, with $T_j \in \mathcal{S}_n$ for $j = 1, \cdots, m$, determine whether, for any two sets S_1, S_2 in $\mathcal{S}_n$ (or, in $\mathcal{S}_{n,d}$), $S_1 \neq S_2$ implies $\text{ANS}_{S_1}(T_j) \neq \text{ANS}_{S_2}(T_j)$ for some $j = 1, \cdots, m$.

We will call a set Q of queries *determinant* for Model X (with respect to size n) if the above problem Determinacy-X has an affirmative answer for input (n, Q). It is easy to see that for any model X, the problem Determinacy-X is in co-NP. We show, in this section, that most of them are actually co-NP-complete. Our main tools are the NP-complete problems One-in-three-SAT and Not-all-equal-SAT. Their precise definitions were given in § 2.

MODEL A_1. We give, in the following, a simple characterization of determinant sets Q of queries for Model A_1. This characterization provides a polynomial-time algorithm for Determinacy-A_1.

LEMMA 4. *A set Q is determinant for Model A_1 with respect to size n if and only if for every $i = 1, \cdots, n$, the singleton set $\{i\}$ is in Q.*

Proof. The backward direction is obvious, because the set $\{i\}$ distinguishes between two sets S_1 and S_2 whenever $i \in S_1 - S_2$.

For the forward direction, we consider two sets $S_1 = \{1, \cdots, n\}$ and $S_2 = S_1 - \{i\}$. Then, the only set T that can distinguish between S_1 and S_2 is $T = \{i\}$ so that $\text{ANS}_{S_1}(T) = 1$ and $\text{ANS}_{S_2}(T) = 0$. □

MODEL B. We show that Not-all-equal-SAT is polynomial-time reducible to the complement of Determinacy-B, and hence Determinacy-B is co-NP-complete.

Let an instance $(U, \mathscr{C})$ of Not-all-equal-SAT be given, where $U = \{x_1, \cdots, x_p\}$, $\mathscr{C} = \{C_1, \cdots, C_q\}$ and for each $j = 1, \cdots, q$, $C_j \subseteq U$ and $|C_j| = 3$. Define an instance

(n, Q) of Determinacy-B as follows:

$$n := p;$$
$$\text{for each } j = 1, \cdots, q, \text{ let } T_{j,0} := \{i \mid x_i \in C_j\}, \text{ and}$$
$$\quad \text{for each } k = 1, \cdots, p, \text{ let } T_{j,k} := T_{j,0} \cup \{k\};$$
$$\text{let } Q = \{T_{j,k} \mid j = 1, \cdots, q; k = 0, \cdots, p\}.$$

(Note that for each j, there are exactly $(p-2)$ $T_{j,k}$'s; however, the total number of $T_{j,k}$'s in Q varies, depending on the set $\mathscr{C}$.)

Assume that t is a truth assignment on U such that for every $j = 1, \cdots, q$, t does not assign equal values to all three variables in C_j. Define $S_1 = \{i \mid t(x_i) = \text{TRUE}\}$ and $S_2 = \{1, \cdots, n\} - S_1$. Then, for each $j = 1, \cdots, q$, $S_1 \cap T_{j,0} \neq \varnothing$, and $S_2 \cap T_{j,0} \neq \varnothing$. This implies that for all $j = 1, \cdots, q$ and for all $k = 0, \cdots, n$, $\text{ANS}_{S_1}(T_{j,k}) = \text{ANS}_{S_2}(T_{j,k}) = 1$. So, Q is not determinant for Model B.

Conversely, assume that Q is not determinant and there are two sets $S_1, S_2 \subseteq \{1, \cdots, n\}$ such that $S_1 \neq S_2$ and $\text{ANS}_{S_1}(T_{j,k}) = \text{ANS}_{S_2}(T_{j,k})$ for all $j = 1, \cdots, q$, and $k = 0, \cdots, p$. Then, we claim that $\text{ANS}_{S_1}(T_{j,0})$ must be equal to 1 for all $j = 1, \cdots, q$.

Suppose, otherwise, that for some j, $\text{ANS}_{S_1}(T_{j,0}) = 0$ or 2. If $\text{ANS}_{S_1}(T_{j,0}) = 0$, then $T_{j,0} \cap S_1 = T_{j,0} \cap S_2 = \varnothing$. This implies that for any $k = 1, \cdots, p$,

$$x_k \in S_1 \Leftrightarrow T_{j,k} \cap S_1 \neq \varnothing \Leftrightarrow \text{ANS}_{S_1}(T_{j,k}) = 1$$
$$\Leftrightarrow \text{ANS}_{S_2}(T_{j,k}) = 1 \Leftrightarrow T_{j,k} \cap S_2 \neq \varnothing \Leftrightarrow x_k \in S_2;$$

or, $S_1 = S_2$. Similarly, if $\text{ANS}_{S_1}(T_{j,0}) = 2$, then $T_{j,0} \subseteq S_1$ and $T_{j,0} \subseteq S_2$. So, for any $k = 1, \cdots, p$,

$$x_k \in S_1 \Leftrightarrow T_{j,k} \subseteq S_1 \Leftrightarrow \text{ANS}_{S_1}(T_{j,k}) = 2$$
$$\Leftrightarrow \text{ANS}_{S_2}(T_{j,k}) = 2 \Leftrightarrow T_{j,k} \subseteq S_2 \Leftrightarrow x_k \in S_2;$$

or, $S_1 = S_2$. Both cases lead to contradictions. So the claim is proven.

Now, define a truth assignment t on U by $t(x_i) = \text{TRUE}$ if and only if $i \in S_1$. The claim that $\text{ANS}_{S_1}(T_{j,0}) = 1$, for all $j = 1, \cdots, q$, implies that t assigns at least one TRUE value and at least one FALSE value to each C_j in $\mathscr{C}$. This completes the proof.

MODEL B'. We show that Determinacy-B is polynomial-time reducible to Determinacy-B'.

Let an instance (p, Q) of Determinacy-B be given such that $Q = \{T_j \mid j = 1, \cdots, q\}$ and each T_j is in $\mathscr{S}_p$. Define an instance of Determinacy-B' as follows:

$$n := 2p; \; m := 2q; \; d := p;$$
$$\text{for each } j = 1, \cdots, q, \text{ let } W_j := \{i + p \mid i \in T_j\};$$
$$\text{let } Q' := \{T_j, W_j \mid j = 1, \cdots, q\}.$$

If Q is not determinant for Model B, then there are $S_1, S_2 \in \mathscr{S}_p$ such that $S_1 \neq S_2$ and for each $j = 1, \cdots, q$, $\text{ANS}_{S_1}(T_j) = \text{ANS}_{S_2}(T_j)$. Define $S_3 := S_1 \cup \{i + p \mid i \notin S_1\}$ and

$S_4 := S_2 \cup \{i + p | i \notin S_2\}$. Then, $S_3 \neq S_4$ and $|S_3| = |S_4| = p$. Furthermore, for each $j = 1, \cdots, q$.

$$\mathrm{ANS}_{S_3}(T_j) = \mathrm{ANS}_{S_1}(T_j) = \mathrm{ANS}_{S_2}(T_j) = \mathrm{ANS}_{S_4}(T_j), \quad \text{and}$$

$$\mathrm{ANS}_{S_3}(W_j) = 2 - \mathrm{ANS}_{S_1}(T_j) = 2 - \mathrm{ANS}_{S_2}(T_j) = \mathrm{ANS}_{S_4}(W_j).$$

So, Q' is not determinant for Model B'.

Conversely, if Q' is not determinant for Model B', then there exist $S_3, S_4 \in \mathcal{S}_n$ such that $S_3 \neq S_4$ and for each $j = 1, \cdots, q$, $\mathrm{ANS}_{S_3}(T_j) = \mathrm{ANS}_{S_4}(T_j)$ and $\mathrm{ANS}_{S_3}(W_j) = \mathrm{ANS}_{S_4}(W_j)$. Since $S_3 \neq S_4$, either $S_3 \cap \{1, \cdots, p\} \neq S_4 \cap \{1, \cdots, p\}$ or $S_3 \cap \{p+1, \cdots, 2p\} \neq S_4 \cap \{p+1, \cdots, 2p\}$. In the former case, we define $S_1 := S_3 \cap \{1, \cdots, p\}$ and $S_2 := S_4 \cap \{1, \cdots, p\}$; and, in the latter case, $S_1 := \{i | i + p \in S_3\}$ and $S_2 := \{i | i + p \in S_4\}$. Then, $S_1 \neq S_2$ but for each $j = 1, \cdots, q$, $\mathrm{ANS}_{S_1}(T_j) = \mathrm{ANS}_{S_2}(T_j)$. So, Q is not determinant for Model B.

MODEL C. We first simplify the problem.

LEMMA 5. *Let $Q = \{T_j | j = 1, \cdots, m\}$ be given such that $T_j \in \mathcal{S}_n$ for all $j = 1, \cdots, m$. Then, Q is not determinant for Model C, with respect to size n, if and only if there exist $S_1, S_2 \in \mathcal{S}_n$ such that $S_1 \cup S_2 \neq \varnothing$, $S_1 \cap S_2 = \varnothing$ and for each $j = 1, \cdots, m$, $|S_1 \cap T_j| = |S_2 \cap T_j|$.*

Proof. The backward direction is obvious. For the forward direction, we note that if S_1' and S_2' are two sets in $\mathcal{S}_n$ such that $S_1' \neq S_2'$ and for each $j = 1, \cdots, m$, $|S_1' \cap T_j| = |S_2' \cap T_j|$. Then, the sets $S_1 = S_1' - S_2'$ and $S_2 = S_2' - S_1'$ satisfy the required condition. □

We now show that One-in-three-SAT is polynomial-time reducible to the complement of Determinacy-C.

Let $(U, \mathcal{C})$ be a given instance of One-in-three-SAT such that $U = \{x_1, \cdots, x_p\}$, $\mathcal{C} = \{C_1, \cdots, C_q\}$ and for every $j = 1, \cdots, q$, $C_j \subseteq U$ and $|C_j| = 3$. Without loss of generality, we assume that every x_i in U occurs in some C_j in $\mathcal{C}$. Define an instance (n, Q) of Determinacy-C as follows:

$n := p + 9q + 1$; $m := 10q$;

for convenience, for each $j = 1, \cdots, q$, and $k = 1, 2, 3$, let

$$u(j,k) := p + 9(j-1) + k,$$

$$v(j,k) := p + 9(j-1) + k + 3,$$

$$w(j,k) := p + 9(j-1) + k + 6;$$

also let $y := p + 9q + 1$;

for each $j = 1, \cdots, q$, assume that $C_j = \{x_{j_1}, x_{j_2}, x_{j_3}\}$ (with $j_1 < j_2 < j_3$), and define

$$T_{j,0} := \{j_1, j_2, j_3, y\},$$

$$T_{j,1} := \{j_2, j_3, u(j,1), v(j,1)\},$$

$$T_{j,2} := \{j_1, j_3, u(j,2), v(j,2)\},$$

$$T_{j,3} := \{j_1, j_2, u(j,3), v(j,3)\},$$

for each $j = 1, \cdots, q$, and each $k = 1, 2, 3$, define

$$U_{j,k} := \{u(j,k), w(j,k)\} \quad \text{and} \quad V_{j,k} := \{v(j,k), w(j,k)\};$$

let $Q := \{T_{j,h}, U_{j,k}, V_{j,k} | j = 1, \cdots, q; h = 0, \cdots, 3; k = 1, 2, 3\}$.

Assume that t is a truth assignment on U such that for each $C_j \in \mathscr{C}$, t assigns exactly one TRUE value to the variables in C_j. Define two sets $S_1, S_2 \in \mathscr{S}_n$ as follows:

$$S_1 := \{i \mid 1 \leqq i \leqq p, t(x_i) = \text{TRUE}\} \cup \{u(j,k), v(j,k) \mid \text{the } k\text{th variable } x_{j_k} \text{ in } C_j \text{ has } t(x_{j_k}) = \text{TRUE}\} \cup \{y\},$$

$$S_2 := \{i \mid 1 \leqq i \leqq p, t(x_i) = \text{FALSE}\} \cup \{w(j,k) \mid \text{the } k\text{th variable } x_{j_k} \text{ in } C_j \text{ has } t(x_{j_k}) = \text{TRUE}\}.$$

Obviously, $S_1 \cup S_2 \neq \varnothing$, $S_1 \cap S_2 = \varnothing$. We claim that for all $R \in Q$, $|S_1 \cap R| = |S_2 \cap R|$. For each $j = 1, \cdots, q$, we check the following:

(i) $|S_1 \cap T_{j,0}| = |S_2 \cap T_{j,0}|$: Among $\{j_1, j_2, j_3\}$, one is in S_1 and two are in S_2; and y is in S_1.

(ii) For $k = 1, 2, 3$, $|S_1 \cap T_{j,k}| = |S_2 \cap T_{j,k}|$: If $t(x_{j_1}) = \text{TRUE}$ and $t(x_{j_2}) = t(x_{j_3}) = \text{FALSE}$, then $u(j, 1)$, $v(j, 1)$ are in S_1. So, $S_1 \cap T_{j,1} = \{u(j, 1), v(j, 1)\}$ and $S_2 \cap T_{j,1} = \{j_1, j_2\}$; and $S_1 \cap T_{j,2} = S_1 \cap T_{j,3} = \{j_1\}$, $S_2 \cap T_{j,2} = \{j_3\}$ and $S_2 \cap T_{j,3} = \{j_2\}$. The other two cases are similar.

(iii) For $k = 1, 2, 3$, $|S_1 \cap U_{j,k}| = |S_2 \cap U_{j,k}|$ and $|S_1 \cap V_{j,k}| = |S_2 \cap V_{j,k}|$: From the definitions of S_1 and S_2, for any $j = 1, \cdots, q$ and $k = 1, 2, 3$, $u(j, k) \in S_1 \Leftrightarrow w(j, k) \in S_2 \Leftrightarrow v(j, k) \in S_1$.

Conversely, assume that Q is not determinant for Model C. Then, by Lemma 5, there exist $S_1, S_2 \in \mathscr{S}_n$ such that $S_1 \cup S_2 \neq \varnothing$, $S_1 \cap S_2 = \varnothing$ and for all $R \in Q$, $|S_1 \cap R| = |S_2 \cap R|$. First note the following fact:

(iv) For all $j = 1, \cdots, q$ and $k = 1, 2, 3$,

$$u(j,k) \in S_1 \Leftrightarrow w(j,k) \in S_2 \Leftrightarrow v(j,k) \in S_1, \quad \text{and}$$

$$u(j,k) \in S_2 \Leftrightarrow w(j,k) \in S_1 \Leftrightarrow v(j,k) \in S_2.$$

Next, we claim the following properties (v) and (vi).

(v) For any $j = 1, \cdots, q$, $|S_1 \cap T_{j,0}| \neq 1$.

Proof of (v). Assume otherwise that $|S_1 \cap T_{j,0}| = 1$. Then $|S_2 \cap T_{j,0}| = 1$. The following case analysis shows that this leads to a contradiction.

Case 1. $S_1 \cap T_{j,0} = \{j_1\}$, $S_2 \cap T_{j,0} = \{j_2\}$. Then, $j_2, j_3 \notin S_1$ and $j_1, j_3 \notin S_2$. So, $S_1 \cap T_{j,1} = S_1 \cap \{u(j, 1), v(j, 1)\}$ and $S_2 \cap T_{j,1} = \{j_2\} \cup (S_2 \cap \{u(j, 1), v(j, 1)\})$. By fact (iv) and the fact that $S_1 \cap S_2 = \varnothing$, we can see that $|S_1 \cap T_{j,1}| \neq |S_2 \cap T_{j,1}|$. This is a contradiction.

Case 2. $S_1 \cap T_{j,0} = \{j_1\}$, $S_2 \cap T_{j,0} = \{y\}$. Then, $j_2, j_3 \notin S_1$ and $j_1, j_2, j_3 \notin S_2$. So, $S_1 \cap T_{j,2} = \{j_1\} \cup (S_1 \cap \{u(j, 2), v(j, 2)\})$ and $S_2 \cap T_{j,2} = S_2 \cap \{u(j, 2), v(j, 2)\}$. Again, a contradiction.

Other cases. All other cases are symmetric to either Case 1 or Case 2.

(vi) $\{1, \cdots, p\} \subseteq S_1 \cup S_2$.

Proof of (vi). Assume otherwise that there is an i, $1 \leqq i \leqq p$, such that $i \notin S_1 \cup S_2$. Assume, without loss of generality, that x_i occurs as the first variable in C_j for some $j = 1, \cdots, q$; i.e., $x_i = x_{j_1}$.

Since $j_1 \notin S_1 \cup S_2$ and $S_1 \cap S_2 = \varnothing$, $|S_1 \cap T_{j,0}| = |S_2 \cap T_{j,0}| \leqq 1$. By claim (v), $S_1 \cap T_{j,0} = S_2 \cap T_{j,0} = \varnothing$. So, $y \notin S_1 \cup S_2$. However, this implies that for all $h = 1, \cdots, q$, $|S_1 \cap T_{h,0}| = |S_2 \cap T_{h,0}| \leqq 1$, and hence, by claim (v), $S_1 \cap T_{h,0} = S_2 \cap T_{h,0} = \varnothing$. This implies that $\{1, 2, \cdots, p\} \cap (S_1 \cup S_2) = \varnothing$.

In addition, fact (iv) shows that for any $h = 1, \cdots, q$ and $k = 1, 2, 3$, $|S_1 \cap T_{h,k}|$ is either 0 or 2. Since $|S_1 \cap T_{h,k}| = 2$ would imply that $|S_2 \cap T_{h,k}| = 0$ and make

$$|S_1 \cap T_{h,k}| \neq |S_2 \cap T_{h,k}|,$$

we must have $S_1 \cap T_{h,k} = \emptyset$. As a consequence, $S_1 = S_2 = \emptyset$. This is a contradiction, and so (vi) is proven.

Now we complete the proof of the reduction. Since $\{1, \cdots, p\} \subseteq S_1 \cup S_2$, y must be in $S_1 \cup S_2$. Assume, without loss of generality, that $y \in S_1$. Define a truth assignment t on U by $t(x_i) = \text{TRUE}$ if and only if $i \in S_1$. Then, for each $j = 1, \cdots, q$, $|S_1 \cap T_{j,0}| = |S_2 \cap T_{j,0}|$ implies that $|S_1 \cap T_{j,0}| = 2$. Since $y \in S_1$, $|S_1 \cap \{j_1, j_2, j_3\}| = 1$. That is, there is exactly one $k \in \{1, 2, 3\}$ such that $t(x_{j_k}) = \text{TRUE}$. This completes the proof for Model C.

MODEL A_k, $k \geqq 4$. Assume that $k \geqq 4$ and that Q is a set of queries each of size $\leqq 4$. Then, Q is determinant for Model C if and only if Q is determinant for Model A_k, because the answering functions for both models behave exactly the same on queries of size $\leqq 4$. In the above, for the problem Determinacy-C, we have actually shown a reduction from One-in-three-SAT to the complement of the following special case of Determinacy-C.

DETERMINACY-$C4$. Given an integer n and a set Q of queries each of size $\leqq 4$, determine whether Q is determinant for Model C with respect to size n.

From the above discussion, this problem is also a special case for Model A_k. So, it also proves that Determinacy-A_k is co-NP-complete.

MODEL C' AND MODEL A'_k, $k \geqq 4$. We can show that Determinacy-$C4$ is polynomial-time reducible to Determinacy-C' and Determinacy-A'_k, for $k \geqq 4$. The reductions are similar to the reduction from Determinacy-B to Determinacy-B'. The key point is that for the answering functions for Model C and Model A_k, $k \geqq 4$, the following property holds for all T of size $\leqq 4$:

$$\text{ANS}_S(T) = |T| - \text{ANS}_{\bar{S}}(T),$$

where $\bar{S} = \{1, \cdots, n\} - S$. This property allows us to carry out the reductions as in the case for Determinacy-B'. We omit the details. (Note that the above property holds for queries T of any size if we only consider Model C. However, for Model A_k, $k \geqq 1$, it only holds for queries T of size $\leqq k$.)

5. Discussion. In the last three sections, we have demonstrated several NP-hardness results on problems related to group testing. The NP-completeness of the consistency problems and the #P-completeness of the counting problems show that the solution spaces associated with arbitrary query histories have complex structures. The co-NP-completeness of the determinacy problems shows that the recognition version of the nonadaptive group testing problems is intractable. It is interesting to compare this problem with the problem of finding a minimal determinant set for Model C, for which a polynomial-time almost-optimal algorithm has been found by Cantor and Mills [1] and Linström [19].

While the complexity for the above three problems has been characterized precisely for most models considered, we have left many more questions open. To name the most important ones, we consider the following two problems concerned with the minimization of the heights of decision trees in the generalized form.

MINIMUM TEST PROBLEM. Given a domain D, a query history H and an integer k, determine whether there is a decision tree of height $\leqq k$ such that each path of the decision tree uniquely determines an object in the solution space associated with the query history H.

MINIMUM NONADAPTIVE TEST PROBLEM. Given a domain D, a query history H and an integer k, determine whether there is a set Q of k queries such that each set of answers to the queries in Q uniquely determines an object in the solution space associated with the query history H.

In the above, the minimum test problem is the generalization of the basic shortest decision tree problem we discussed in § 1, and the minimum nonadaptive test problem is the corresponding problem for the nonadaptive case. It is not hard to see that for models considered in this paper, the minimum nonadaptive test problems are in Σ_2^P, and the minimum test problems are in PSPACE, where Σ_2^P is the class of languages recognized by nondeterministic oracle Turing machines in polynomial time relative to oracle sets in NP [8], and PSPACE is the class of languages recognized by deterministic Turing machines in polynomial space [8]. Furthermore, the proofs of the NP-completeness of the consistency problems can easily be modified to show the NP-hardness of the minimum nonadaptive test problems and the minimum test problems for the same models. In view of the difficulty of getting optimal algorithms for these problems even for simple initial solution spaces and the complex structure of general solution spaces, we conjecture that the minimum nonadaptive test problems for most models are Σ_2^P-complete and the minimum test problem for most models are PSPACE-complete.

Other interesting questions include the following:

(1) Instead of the query history, we may use different representations for a solution space, for example, by listing its elements explicitly. What are the effects of these different representations of solution spaces on the computational complexity of the questions considered here?

(2) Do these NP-hardness results hold for the group testing problems with respect to the average-case complexity?

(3) Can we prove completeness results for other searching problems which involve the minimization of the heights of decision trees?

Acknowledgment. The authors thank Professor Ronald Book for his suggestions and encouragement.

REFERENCES

[1] D. G. CANTOR AND W. H. MILLS, *Determination of a subset from certain combinatorial properties*, Canad. J. Math., 18 (1966), pp. 42–48.

[2] X. M. CHANG AND F. K. HWANG, *The minimax number of calls for finite population multiple-access channel*, manuscript.

[3] R. DORFMAN, *The detection of defective members in large population*, Ann. Math. Statist., 14 (1943), pp. 4436–4440.

[4] P. ERDÖS AND A. RENYI, *On two problems of information theory*, Publ. Hung. Acad. Sci., 8 (1963), pp. 241–254.

[5] S. EVEN AND R. E. TARJAN, *A combinatorial problem which is complete in polynomial space*, J. Assoc. Comput. Mach., 23 (1976), pp. 710–719.

[6] S. FORTUNE, *A note on sparse complete sets*, SIAM J. Comput., 8 (1979), pp. 431–433.

[7] M. R. GAREY, *Optimal binary identification procedures*, SIAM J. Appl. Math., 23 (1972), pp. 173–186.

[8] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability*, W. H. Freeman, San Francisco, 1979.

[9] E. P. GUDJOHNSEN, D. TOWSLEY AND J. K. WOLF, *On adaptive polling techniques for computer communication networks*, ICC 80 Conference Record 1, Seattle, WA, 1980, pp. 13.3.1–13.3.5.

[10] F. K. HWANG, *Three versions of a group testing game*, this Journal, 5 (1984), pp. 145–153.

[11] ———, *When is group testing querying profitable in communication networks?*, manuscript.

[12] ———, Private communications.

[13] F. K. HWANG AND X. M. CHANG, *Cutoff points for roll call protocals in multiple access system*, IEEE Trans. Inform. Theory, 33 (1987), pp. 577–581.

[14] F. K. HWANG, T. T. SONG AND D. Z. DU, *Hypergeometric group testing and generalized hypergeometric group testing problems*, this Journal, 2 (1981), pp. 426–428.

[15] F. K. HWANG AND V. T. SOS, *Non-adaptive hypergeometric group testing*, Studia Sci. Math. Hungar., to appear.

[16] D. E. KNUTH, *The Art of Computer Programming*, Vol. 3, *Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.

[17] K. KO AND S. TENG, *On the number of queries necessary to identify a permutation*, J. Algorithms, 7 (1986), pp. 449–462.

[18] N. LINIAL, *The information-theoretic bound is good for merging*, SIAM J. Comput., 13 (1984), pp. 795–801.

[19] B. LINSTRÖM, *On a combinatorial problem in number theory*, Canad. Math. Bull., 8 (1965), pp. 477–490.

[20] G. K. MANACHER, *The Ford–Johnson sorting algorithm is not optimal*, J. Assoc. Comput. Mach., 26 (1979), pp. 441–456.

[21] N. MEHRAVARI, *Generalized binary binomial group testing*, this Journal, 7 (1986), pp. 159–166.

[22] T. J. SCHAEFER, *The complexity of satisfiability problems*, Proc. 10th Annual ACM Symposium on Theory of Computing, 1978, pp. 216–226.

[23] M. SOBEL, *Binomial and hypergeometric group-testing*, Studia Sci. Math. Hungar., 2 (1968), pp. 19–42.

[24] M. SOBEL AND P. A. GROLL, *Group-testing to eliminate efficiently all defectives in a binomial sample*, Bell System Tech. J., 38 (1959), pp. 1179–1252.

[25] A. STERRETT, *On the decision of defective members of large population*, Ann. Math. Statist., 28 (1957), pp. 1033–1036.

[26] J. THOMAS, B. S. PASTERNACK, S. J. VACIRCA AND D. L. THOMPSON, *Application of group testing procedures in radiological health*, Health Physics, 25 (1973), pp. 259–266.

[27] B. S. TSYBAKOV, V. A. MIKHAILOV AND N. B. LIKHANOV, *Bounds for packet transmission rate in a random-multiple-access system*, Problemy Peredachi Informatsii, 19 (1983), pp. 61–81. (In Russian.)

[28] L. G. VALIANT, *The complexity of computing the permanent*, Theoret. Comput. Sci., 9 (1979), pp. 189–201.

[29] G. S. WATSON, *A study of the group screening method*, Technometrics, 3 (1961), pp. 371–388.