NP #P = ∃PP and other remarks about maximized counting

David Monniaux

$\mathsf{NP}^{\#\mathsf{P}} = \exists\mathsf{PP}$ and other remarks about maximized counting

David Monniaux

CNRS / VERIMAG

February 23, 2022

We consider the following decision problem DMAX#SAT, and generalizations thereof: given a quantifier-free propositional formula $F(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}, \mathbf{y}$ are tuples of variables, and a bound B, determine if there is $\mathbf{x}$ such that $\#\{\mathbf{y} \mid F(\mathbf{x}, \mathbf{y})\} \geq B$. This is the decision version of the problem of MAX#SAT: finding $\mathbf{x}$ and B for maximal B.

Theorem 1. DMAX#SAT *is* $\exists\mathsf{PP}$*-complete.*

Proof. It is in $\exists\mathsf{PP}$: it is well-known that taking (F, B) as input and checking if $\#\mathrm{SAT}(F) \geq B$ is in PP.

Take a problem in $\exists\mathsf{PP}$. It can be reformulated as: take input x, choose nondeterministic bits y, construct a formula $F(x, y)$ with $N(x, y)$ variables, and check that $\#\{z \in \{0,1\}^{N(x,y)} \mid F(x, y)(z)\} \geq 2^{N(x,y)-1}$. The condition $F(x, y)(z)$ can also be reformulated as $\exists z'\ G(x, y, z, z')$ where G simulates the action of the Turing machine that produces F (if necessary by using temporary values in z') then the semantics of the formula over z. The result follows. □

Torán [1, theorem 4.1 (ii)] showed that $\exists\mathsf{PP} = \mathsf{NP}^{\#\mathsf{P}}$; actually, a generalization of this. However, prior to becoming aware of that result, we had worked out another proof, which we present here.

The following gadgets enables us to transform multiple equality tests over model counts $\#\mathrm{SAT}(F_1) = C_1 \wedge \ldots \wedge \#\mathrm{SAT}(F_m) = C_m$ into a single equality test over model counts.

Definition 1. Let F and G be two quantifier-free propositional formulas with m and n variables respectively. Without loss of generality, we assume these variables to be $x_1, \ldots, x_m$ and $x_1, \ldots, x_n$. Let $\phi_2^{m,n}(F, G)$ be the following formula over $m + n + 2$ variables:

$$(F(x_1, \ldots, x_m) \wedge \neg x_{m+1} \wedge \ldots \wedge \neg x_{m+n+2}) \vee (G(x_1, \ldots, x_m) \wedge x_{m+1}) \qquad (1)$$

By $|F|$ we denote the size of a formula as the number of its Boolean operators, and by $\#\mathrm{SAT}(F)$ we denote the number of its models.

Lemma 1. $|\phi_2^{m,n}(F, G)| = |F| + |G| + n + 3$. *Furthermore,* $\#\mathrm{SAT}(F)$ *and* $\#\mathrm{SAT}(G)$ *are respectively the remainder and quotient of* $\#\mathrm{SAT}(\phi_2^{m,n}(F, G))$ *by* 2^{n+1}.

Definition 2. Let $F_0, \ldots$ be propositional formulas with n variables. Let $\phi_1^n(F_0) = F_0$, $\phi_{k+1}^n(F_0, \ldots, F_k) = \phi_2^{kn+2(k-1),n}(\phi_k(F_0, \ldots, F_{k-1}), F_k)$.

Lemma 2. $|\phi_k^n(F_0, \ldots, F_{k-1})| = \sum_i |F_i| + (k-1)(n+3)$. *Furthermore,* $\#\mathrm{SAT}(F_i)$ *is the digit of order* i *(starting with* $i = 0$*) of the decomposition of* $\#\mathrm{SAT}(\phi_2^{m,n}(F, G))$ *in base* 2^{n+1}.

The following gadget will be used to add a number of models to an existing formula:

Definition 3. Let $M_c^n(x_0, \ldots, x_{n-1})$, where $0 \leq c \leq 2^n$, be the formula that specifies that $\sum_i 2^i x_i \leq c$.

Lemma 3. $|M_c^n|$ *is linear in* n*, and* $\#\mathrm{SAT}(M_c^n) = c$.

The following gadget will be used to turn an equality test on the number of models of a formula into a "greater than or equality" inequality test on the number of models of another formula:

Definition 4. Let F be a propositional formula over n variables, and $0 \leq \Delta \leq 2n - 1$. Let $\psi^n(F)$ be the formula over $2n + 1$ variables

$$F(x_1, \ldots, x_n) \wedge ((\neg F(x_{n+1}, \ldots, x_{2n}) \wedge \neg x_{2n+1}) \vee (M_{2\Delta}^n(x_{n+1}, \ldots, x_{2n}) \wedge x_{2n+1})) \tag{2}$$

Let K_Δ^n be the polynomial $K_\Delta^n(X) = X(2^n - X + 2\Delta)$.

Lemma 4. $|\psi^n(F)|$ *has size linear in* $|F|$*, and* $\#\mathrm{SAT}(\psi^n(F)) = K_\Delta^n(\#\mathrm{SAT}(F))$. *Furthermore,* $K_\Delta^n(\#\mathrm{SAT}(F)) \geq K_\Delta^n(2^{n-1} + \Delta)$ *if and only if* $\#\mathrm{SAT}(F) = 2^{n-1} + \Delta$.

Theorem 2. $\exists\mathsf{PP} = \mathsf{NP}^{\mathsf{PP}[1]} = \mathsf{NP}^{\mathsf{PP}} = \mathsf{NP}^{\#\mathsf{P}}$.

Proof. Inclusions from left to right are trivial. We shall now prove that $\mathsf{NP}^{\#\mathsf{P}}$ is included in $\exists\mathsf{PP}$ by transforming a nondeterministic Turing machine M deciding a problem $D(x)$ in time $P(|x|)$ with a #SAT oracle into an equivalent decision procedure in $\exists\mathsf{PP}$.

We proceed in steps:

1. M calls the oracle at most $P(|x|)$ times, over formulas of $P(|x|)$ variables, and the outputs of each oracle call may be used for computing the inputs to further oracle calls. Instead, we transform the machine to nondeterministically choose all inputs $F_1, \ldots, F_{P(|x|)}$ and candidate outputs to the oracle calls (#SAT), and then only at the end we verify that the candidate outputs $C_1, \ldots, C_{P(|x|)}$ match the real outputs $\#\mathrm{SAT}(F_1), \ldots, \#\mathrm{SAT}(F_{P(|x|)})$ (we reject otherwise).

2. We replace these calls by a single call to $\#\mathrm{SAT}(\phi_{P(|x|)}^{P(|x|)}(F_1, \ldots, F_{P(|x|)}))$, and a single verification that the output V of this call matches the candidate outputs $C_1, \ldots, C_{P(|x|)}$ according to the decomposition in Lemma 2.

3. We replace this call and equality test $\#\mathrm{SAT}(\phi_{P(|x|)}^{P(|x|)}(F_1, \ldots, F_{P(|x|)})) = Y$ by a single call to #SAT and an inequality test as follows. Let n be the number of variables of $\phi_{P(|x|)}^{P(|x|)}(F_1, \ldots, F_{P(|x|)})$. Two cases:

- $Y \geq 2^{n-1}$, then write $Y = 2^{n-1} + \Delta$. Then we replace the equality test by an inequality test $\#\mathrm{SAT}(\psi^n(\phi^{P(|x|)}_{P(|x|)}(F_1, \ldots, F_{P(|x|)}))) \geq K^n_\Delta$, according to Lemma 4.
- $Y < 2^{n-1}$, then write $Y = 2^{n-1} - \Delta$. Then we replace the equality test by an inequality test $\#\mathrm{SAT}(\psi^n(\neg\phi^{P(|x|)}_{P(|x|)}(F_1, \ldots, F_{P(|x|)}))) \geq K^n_\Delta$, according to Lemma 4.

4. We have thus reduced the procedure to the nondeterministic (possiblyfailing) construction of a pair (G, B) followed by a test $\#\mathrm{SAT}(G) \geq B$. This test is well-known to be complete for PP.

□

Corollary 1. DMAX#SAT *is hard for the polynomial hierarchy.*

Proof. Obviously, $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{NP}^{\#\mathsf{P}}$, and the former class is hard for the polynomial hierarchy by Toda's theorem. □

Remark 1. The above lemmas and theorems consider quantifier-free propositional formulas over a number of free variables. In fact, we can use arbitrary predicates with a certain number of free variables: for instance, take inputs $x_1, \ldots, x_n$, and return true or false depending on whether a certain polynomial-time nondeterministic Turing machine parameterized by $x_1, \ldots, x_n$ has an accepting run or not. Equivalently, we could consider predicates of the form $\exists z_1, \ldots, z_p\, F(x_1, \ldots, x_n, z_1, \ldots, z_p)$. We then obtain the result $\exists\mathsf{PNP} = \mathsf{NP}^{\#\mathsf{NP}}$.

By going with classes of predicates arbitrarily high in the counting hierarchy, we obtain the general theorem [1, theorem 4.1 (ii)]: for any class K is the counting hierarchy, $\exists\mathsf{P}K = \mathsf{NP}^{\mathsf{P}K[1]} = \mathsf{NP}^{\mathsf{P}K} = \mathsf{NP}^{\#K}$.

This result is not limited to the counting hierarchy, it is appropriate for classes of predicates stable by certain operations (conjunction, disjunction, conjunction with extra propositional inputs...).

References

[1] Jacobo Torán. "Complexity Classes Defined by Counting Quantifiers". In: *J. ACM* 38.3 (1991), pp. 753–774. DOI: 10.1145/116825.116858.