PH = PSPACE

Valerii Sopin[0000−0001−9953−6072]

VvS@myself.com, https://www.researchgate.net/profile/Valerii-Sopin

Using $\mathbb{PSPACE}$-completeness of quantified boolean logic it is claimed that $\mathbb{PH} = \mathbb{PSPACE}$.

Next theorem shows that the quantified Boolean formula problem is generalisation of the Boolean Satisfiability Problem, where determining of interpretation that satisfies a given Boolean formula is replaced by existence of Boolean functions that makes a given QBF to be tautology.

Theorem 1. *The quantified Boolean formula*

$$\Omega_1 x_1 \in \{0,1\}\ \Omega_2 x_2 \in \{0,1\}\ \ldots\ \Omega_n x_n \in \{0,1\}\ \ \phi(x_1, \ldots, x_n),$$

where $\phi(x_1, \ldots, x_n)$ is a Boolean formula, Ω_s, $s = i_1, \ldots, i_j$, is the quantifier $\exists$ and Ω_t, $t \neq i_1, \ldots, i_j$, is the quantifier $\forall$, j is the number of variables with the quantifier $\exists$, is a true quantified Boolean formula if and only if there are Boolean functions y_q, where y_q depends only on variables with the quantifier $\forall$ and indexes less i_q, $q = 1, \ldots, j$, that after substituting $x_{i_q} := y_q$ the given quantified Boolean formula becomes tautology.

Proof. It follows from simple recursive algorithm for determining whether a QBF is true. We take off the first quantifier and check both possible values for the first variable:

$$A = \Omega_2 x_2 \in \{0,1\}\ \ldots\ \Omega_n x_n \in \{0,1\}\ \ \phi(0, \ldots, x_n),$$

$$B = \Omega_2 x_2 \in \{0,1\}\ \ldots\ \Omega_n x_n \in \{0,1\}\ \ \phi(1, \ldots, x_n).$$

If $\Omega_1 = \exists$, then return A disjunction B (that's it, A or B is true; to avoid unambiguous, if A and B is true, take A for determining the function, so the value depends only on values of previous variables).

If $\Omega_1 = \forall$, then return A conjunction B (A and B is true).

Notice that a Boolean function determines the truth table (one-to-one correspondence).

`Example` $\forall x_1 \exists z_1 \forall x_2 \exists z_2 \forall x_3 \exists z_3\ \phi(x_1, z_1, x_2, z_2, x_3, z_3)$ is a true QBF if and only if there exist such Boolean functions $y_1 : \{0,1\} \to \{0,1\}$, $y_2 : \{0,1\}^2 \to \{0,1\}$, $y_3 : \{0,1\}^3 \to \{0,1\}$ that given Boolean formula $\phi(x_1, y_1(x_1), x_2, y_2(x_1, x_2), x_3, y_3(x_1, x_2, x_3))$ is tautology.

Theorem 2.

$$\prod\nolimits_4 = (co\text{-}\mathbb{NP})^{\mathbb{NP}^{(co\text{-}\mathbb{NP})^{\mathbb{NP}}}} = \mathbb{PSPACE}$$

Proof. From [1][6] we know that without loss of generality we can assume a quantified Boolean formula to be in form (prenex normal form), where existential and universal quantifiers alternate. We assume it, for simplicity.

We wish that an quantified Boolean formula

$$\forall x_1 \in \{0,1\}\ \exists y_1 \in \{0,1\}\ \forall x_2 \in \{0,1\}\ \exists y_2 \in \{0,1\}\ \ldots\ \forall x_n \in \{0,1\}\ \exists y_n \in \{0,1\}$$

$$\phi(x_1, y_1, \ldots, x_n, y_n)$$

would be equivalent to

$\forall(x_1, x_2, \ldots, x_n)\ \exists(y_1, \ldots, y_n)\{$

$$\phi(x_1, y_1, x_2, y_2, \ldots, x_n, y_n)\ \wedge$$

$$\wedge\ \forall(\hat{x}_n)\ \exists(z_n)\ \ \phi(x_1, y_1, x_2, y_2, \ldots, x_{n-1}, y_{n-1}, \hat{x}_n, z_n)\ \wedge$$

$$\wedge\ \forall(\hat{x}_{n-1}, \hat{x}_n)\ \exists(z_{n-1}, z_n)\ \ \phi(x_1, y_1, x_2, y_2, \ldots, x_{n-2}, y_{n-2}, \hat{x}_{n-1}, z_{n-1}, \hat{x}_n, z_n)\ \wedge\ \ldots$$

$$\ldots\ \wedge\ \forall(\hat{x}_2, \ldots, \hat{x}_n)\ \exists(z_2, \ldots, z_n)\ \ \phi(x_1, y_1, \hat{x}_2, z_2, \ldots, \hat{x}_{n-2}, z_{n-2}, \hat{x}_{n-1}, z_{n-1}, \hat{x}_n, z_n)$$

$$\}$$

Namely, iterations of $\forall x \exists y$ reduce to conjunctions of separated $\forall \hat{x} \exists z$, as in the beginning we fix values of $\{y_q,\ q = 1, \ldots, n\}$ and conjunctions jointly check that for predetermined $\{y_l,\ l < q\}$ suitable continuation $\{y_l,\ l \geq q\}$ can be found. In each conjunction we consider $\{y_l, l < q\}$ as functions dependent on all $\{x_i,\ i < q\}$ and $\{y_l, l \geq q\}$ as functions dependent on every $\{x_i,\ i = 1, \ldots, n\}$ (if $\forall x_1\ F(x_1, 0) = F(x_1, 1)$, then variable x_2 is dummy variable for Boolean formula $F(x_1, x_2)$). From here, if it is a true quantified Boolean formula, the above confirms it. However, another implication is not always true. Let's exam when two parts are different, allowing ϕ to have also odd number of variables with preserving alternations for quantifiers for foregoing induction.

$\mathbf{m = 1}$**:** for a Boolean formula of one variable the equivalence obviously holds.

$\mathbf{m = 2}$**:** inconsistency can possibly happen only with $\exists y\ \forall x\ \phi(y, x)$; we have 16 different Boolean formulas of two variables and the equivalence is violated only for $XOR : (y \oplus x), \neg(y \oplus x)$.

$\mathbf{m \geq 3}$**:** taking off the first quantifier and checking both possible values for the first variable in way we did in Proposition, we come to the **m - 1** case. Indeed, for example, considering $\mathbf{m = 3}$, we have

$$\forall z\ \exists y\ \forall x\ \ \phi(z, y, x) \equiv \exists y\ \forall x\ \ \phi(0, y, x)\ AND\ \exists y\ \forall x\ \ \phi(1, y, x),$$
$$\exists t\ \forall x\ \exists y\ \ \phi(t, x, y) \equiv \forall x\ \exists y\ \ \phi(0, x, y)\ OR\ \forall x\ \exists y\ \ \phi(1, x, y),$$

where the second expression can be viewed as negation of the first expression. Consequently, it is enough to inspect only first expression due to double negation.

If $\exists y\ \forall x\ \ \phi(0, y, x) \equiv \forall x\ \exists y\ \ \phi(0, y, x)$ and $\exists y\ \forall x\ \ \phi(1, y, x) \equiv \forall x\ \exists y\ \ \phi(1, y, x)$, then $\forall z\ \exists y\ \forall x\ \ \phi(z, y, x) \equiv \forall z\ \forall x\ \exists y\ \ \phi(z, y, x) \equiv \forall z\ \exists \xi\ \forall x\ \exists y\ \ \phi(z, y, x)$. Otherwise, the equivalence is false due to XOR issue from $\mathbf{m = 2}$. Then $\exists y\ \forall x\ \ \phi(0, y, x)$ or $\exists y\ \forall x\ \ \phi(1, y, x)$ is false. Therefore, $\forall z\ \exists y\ \forall x\ \ \phi(z, y, x)$ is false.

Thus, using mathematical induction we have shown that XOR issue from $\mathbf{m = 2}$ appears whenever the equivalence we want doesn't work and the emergence means that the real value is false, but the displayed formula says that it is true. So, for each

$$\forall(x_1, x_2, \ldots, x_n)\ \exists(y_1, \ldots, y_n)\ \forall(\hat{x}_i, \ldots, \hat{x}_n)\ \exists(z_i, \ldots, z_n)\ \ \phi(x_1, y_1, x_2, y_2, \ldots, \hat{x}_i, z_i, \ldots, \hat{x}_{n-1}, z_{n-1}, \hat{x}_n, z_n)$$

we additionally need to verify that $\phi(x_1, y_1, x_2, y_2, \ldots, \hat{x}_i, z_i, \ldots, \hat{x}_{n-1}, z_{n-1}, \hat{x}_n, z_n)$ as formula of two variables ($x \in \{x_1, \ldots, x_{i-1}, \hat{x}_i, \ldots, \hat{x}_n\}, y \in \{y_1, \ldots, y_{i-1}, z_i, \ldots, z_n\}$) (all other arguments are specified; there are n^2 such formulas) is not equivalent to $\exists y\ \forall x\ (x \oplus y)$ or $\exists y\ \forall x\ \neg(x \oplus y)$. Otherwise, we consider the value ϕ with precise $x_1, y_1, x_2, y_2, \ldots, \hat{x}_i, z_i,\ \ldots, \hat{x}_{n-1}, z_{n-1}, \hat{x}_n, z_n$ as *false*, even if it is *true*. This extra condition can be checked in polynomial time.

To conclude, definition of alternating Turing machine shows that $(\text{co-}\mathbb{NP})^{\mathbb{NP}^{(\text{co-}\mathbb{NP})^{\mathbb{NP}}}}$ is enough and this way we solve complete problem for $\mathbb{PSPACE}$.

`Corollaries`:

(1) The polynomial hierarchy collapses and $\mathbb{BQP} \subseteq \mathbb{PH}$;
(2) $\mathbb{P}^{\mathbb{PP}} = \mathbb{PSPACE}$;
(3) If $\mathbb{P} = \mathbb{NP}$, then $\mathbb{P} = \mathbb{PSPACE}$. If $\mathbb{NP} = \text{co-}\mathbb{NP}$, then $\mathbb{NP} = \mathbb{PSPACE}$.
(4) If $\mathbb{NP} \subseteq \mathbb{BQP}$, then $\mathbb{BQP} = \mathbb{PSPACE}$.

Remark 1. Dependency quantified Boolean formulas (DQBFs) are a generalization of ordinary quantified Boolean formulas. While the latter is restricted to linear dependencies of existential variables in the quantifier prefix, DQBFs allow arbitrary dependencies, which are explicitly specified in the formula. This makes decision problem with a DQBF to be $\mathbb{NEXP}$-complete (the complexity class $\mathbb{NEXP}$ is the set of decision problems that can be solved by a non-deterministic Turing machine using exponential time, i.e., in $O(2^{p(n)})$ time, $p(n)$ is a polynomial function of n). Theorem 2 is not applicable to the case of DQBFs directly as the looping is possible. Is it within reach to generalise Theorem 2 for it?

Remark 2. Multiset $\{\mathbb{P}, \mathbb{NP}, \mathbb{NP}^{\mathbb{NP}}, \mathbb{NP}^{\mathbb{NP}^{\mathbb{NP}}}, \mathbb{NP}^{\mathbb{NP}^{\mathbb{NP}^{\mathbb{NP}}}}\}$ is a miracle, which shows that there is always a key.

Remark 3. A related point to consider is L. Gordeew, E. H. Haeusler, Proof Compression and NP Versus PSPACE, Studia Logica, 107:1, 2019, 55–83; L. Gordeew, E. H. Haeusler, Proof Compression and NP Versus PSPACE II, Bulletin of the Section of Logic , 49:3, 2020, 213–230.

Sopin Valerii, "PH = PSPACE", arXiv:1411.0628, 2014