2-TQBF is in P

Chris Calabro

March 24, 2008

Abstract

It is well known that 2-SAT $\in$ P, but the analogous result for 2-TQBF is a bit less well known. I couldn't it find anywhere, so I wrote this. Later I was told by Sam Buss that it was shown to be in linear time in [PAT79]. My proof here at least is easier to understand, but less insightful.

Theorem 1. 2-TQBF $\in$ P.

Proof. Let E, A represent players Exists, Forall. Given the QBF $F = \exists x_1 \, \forall x_2 \, \cdots \, Q_n x_n \, \phi$, where $\phi \in$ 2-CNF and $Q_n \in \{\exists, \forall\}$ is $\exists$ iff n is odd, E, A play a game where in round i, E sets x_i to a bit if i odd, and A sets x_i if i is even, and after round n, E wins iff ϕ evaluates to true, else A wins.

Each clause is 1 of the following types:

$$(), (A), (A\ A), (E), (E\ A), (E\ E).$$

where E, A means the literal is under the control of player E, A. If a, b are literals, we write $a < b$ to mean that the variable of a must be played earlier in the game than the variable of b. E.g. in the formula $F = \exists x_1 \, \forall x_2 \phi$, $x_1 < x_2$. Also, we write $(Ex\ Ay)$ to mean the clause $(x\ y)$ where x is under control of E and y is under control of A. Here is the algorithm.

$M(F \in$ 2-QBF)
  while some rule below applies, apply it (in any order):
    1. pure literal rule.
      i.e. if a E, A literal is pure, replace it with $1, 0$
      and remove false literals and true clauses.
    2. resolve the formula if it is not fully resolved.
    3. remove tautological clauses,
      i.e. those of the form $(a\ \bar{a})$.
    4. if there is a clause of the form $(), (A)$, or $(A\ A)$, then reject.
    5. if there is a clause of the form (E), replace it with 1
      and remove false literals and true clauses.
    6. remove a redundant variable,
      i.e. where we have $(a\ \bar{b})$ and $(\bar{a}\ b)$.

case I: $(a\ \bar{b})$ is $(E\ E)$. replace b by a.
case II: $(a\ \bar{b})$ is $(E\ A)$.
if $a < b$, then reject.
if $a > b$, then replace a by b.
case III: $(a\ \bar{b})$ is $(A\ A)$. reject.
accept

First, we show that application of each rule does not affect the truth value of F. This is only slightly not obvious for rule 6, case II: if $a < b$, then player A will set b to the opposite of however player E sets a, and so E loses. If $a > b$, then player E must set a to b.

Next, we show that this algorithm terminates in poly time. Each rule but 2,3 eliminates a variable, so there can be at most n such applications. Rule 2 adds clauses and 3 eliminates them, but rule 2 cannot generate new tautological clauses, and rule 3 cannot make a fully resolved formula not fully resolved. Also both rules are idempotent. So neither can be applied more than once without applying another rule in between. So the number of times the rules can be applied is $\leq 3n$. Resolution takes poly time because there are $O(n^2)$ 2-clauses. The other rules take poly time as well.

Next we show correctness. If M rejects, then since application of the rules did not affect the truth value of F, it is obvious that F is false. If M accepts, then no rule applies. It is sufficient to show that in this case there are no A variables, since then the truth value of F, which is fully resolved, is equivalent to the absence of an empty clause.

So suppose indirectly that F contains an A variable and no rule applies. To derive a contradiction, we will develop an infinite sequence $a_1, a_2, a_3, \ldots$ of literals with distinct variables s.t. for some literal b and all i, F contains $(Ea_i\ Ab)$, and for all $j < i$, F contains $(E\bar{a}_j\ Ea_i)$.

Because of rules 3,4,5, there can only be 2 kinds of clauses in F: $(E\ E)$ and $(E\ A)$, with distinct variables in each case. Since F contains an A variable, there must be a clause $(Ea_1\ Ab)$. This establishes the base case. So suppose $a_1, \ldots, a_i$ have been constructed for some $i \geq 1$. Because of the pure literal rule, $\bar{a}_i$ appears in a clause C.

case 1 : $C = (E\bar{a}_i\ A\bar{b})$. A contradiction since rule 6 would apply.

case 2 : $C = (E\bar{a}_i\ Ac)$ where $c \neq \bar{b}$ (but possibly $c = b$). A contradiction since resolution would yield $(Ab\ Ac)$ and rule 4 would apply.

case 3 : $C = (E\bar{a}_i\ Ea_{i+1})$ where a_i, a_{i+1} involve distinct variables. Resolution shows that for each $j < i$, F contains $(E\bar{a}_j\ Ea_{i+1})$. For each $j < i$, a_j, a_{i+1} have distinct variables, since otherwise rule 6 would apply. Finally, resolution shows that F contains $(Ea_{i+1}\ Ab)$.

This establishes the inductive case.

Since F has only finitely many variables, this is a contradiction. □

Before you get your hopes up, this does not prove P = PSPACE, since we cannot represent a 3-CNF as a 2-QBF, as we show next.

Theorem 2. $\neg\exists\phi \in$ 2-CNF $\forall x, y, z \in 2$

$$(\exists w_1 \forall w_2 \cdots Q_n w_n \ \phi(x, y, z, w_1, \ldots, w_n) \leftrightarrow x \vee y \vee z).$$

I.e. you can't express a 3-clause as a 2-QBF.

Proof. Suppose not. Choose a minimal such ϕ; i.e. s.t. you can't remove any quantified variable nor any clause and still have the desired property. ϕ can only have $(E\ E)$ and $(E\ A)$ clauses. Also neither x nor $\bar{x}$ can appear with an A variable, since otherwise, some setting of x would make the formula false.

We claim that $\bar{x}$ does not occur. If it did, it would appear as $C = (\bar{x}\ w)$ for some w. w cannot be any of $y, \bar{y}, z, \bar{z}$, otherwise some setting of those 2 vars would make the formula false. So w is an E literal: $C = (\bar{x}\ Ew)$. We claim that we can get rid of clause C. If $x \vee y \vee z$, then the original formula is true, so it is true without C. If $\bar{x} \wedge \bar{y} \wedge \bar{z}$, then the original formula is false, but C is true. So the original formula must be false without C. So we can get rid of C, contradicting minimality of the original formula. So $\bar{x}$ does not occur, and similarly for $\bar{y}, \bar{z}$.

x occurs in some clause, else the formula does not depend on x. So there is a clause $(x\ Ew_1)$. Every w_i occurs both positively and negatively, else the formula is not minimal. So $\bar{w}_1$ occurs somewhere. If it occurs in a $(E\bar{w}_1\ Ew_2)$ clause, we continue. In such a way, we develop a sequence $w_1, w_2, \ldots$. There are only finitely many variables, so we must stop at either x, y, z, an A variable, or some w_i that occurred already, with $i > 1$ (if $i = 1$, then we have the clause $(w_1\ \bar{w}_1)$, in which case the formula is not minimal). If we stop at x, then resolution would derive the clause (x), which is impossible. If we stop at y (or z), then resolution would derive the clause $(x\ y)$ (or $(x\ z)$), which is impossible. If we stop at an A variable, then resolution would derive the clause $(x\ A)$, which is impossible. If we stop at w_i, then $w_1, \ldots, w_i$ are equivalent and we can reduce the number of variables (since $i > 1$), contradicting minimality of the formula. We have a contradiction in all cases. □

References

[PAT79] M. Plass, B. Aspval, R. Tarjan. A linear time algorithm for testing the truth of certain quantified boolean formulas. Information Processing Letter, 8:121–123, 1979.