Quantified Boolean Formula

December 17, 2019

1 quantified boolean formula

main proof: Satisfiability of 2-QB-3CNF with n variables and poly(n) clauses can be solved in time$2^{n-\Omega(n^{1/2})}$

To proof this, we need to use a lemma proofed in the other paper.

lemma:there is a deterministic algorithm A solving satisfiability of CNFs with m clauses and n variables in time O($poly(m)2^{n-n/(1+log(m))}$)

We will run a recursive algorithm R on the given 3-CNFs formula an n variables and two quantifier blocks.

n=the number of variables in the original formula.

u= the number of universal variables,e= the number of existential variables such that $n = u + e$

Here we divide the algorithm R into 4 cases, and fix a parameter ϵ later.

a) the trivial cases.

If there is a clauses containing only universal literals or an empty clauses, return 0, if there are no clauses left ti satisfy,return 1.

if there are no universal literals in the formula, the instance is a 3-SAT. If the 3-SAT is satisfiable, return 1, else return 0.

b) if $e > \sqrt{n}$

Try all $2n - e$ assignments to the universals and using lemma solve the remaining 3-CNF in 1.4^e time. this case takes $O^*(2^u 1.4^e)$

c) if $e \leq \sqrt{n}$

then suppose there exists a clause with two universal and 1 existe.ntial variables. Let it be $(u_i \vee u_j \vee e_k)$. Perform the next three recalls:

- set $(u_i = 1)$ and call the algorithm R on the formula, named **type A**
- set $(u_i = 0)$ $(u_j = 1)$.call R **type B**
- set $(u_i = 0)$,$(u_j = 0)$,$(e_k = 1)$ call R. **type C**

In this case, we first need to bound the number of leaves N of the recursion tree. we classify the leaves according how many times type C occur. denote i the number of type C occur, and $f(i)$ denote the number of leaves for which i calls of type C occur on the path from root to leaf, and $0 \leq i \leq \sqrt{n}$.

$f(0) = O(\phi^u) = O(\phi^n)$, $f(i) \leq C_n^i O(\phi^u)$, as there are C_n^i ways of choosing the i levels of the recursion tree at which calls of type C are made.

thus the number of leaves can be bounded by$N \leq \sum_{i=1}^{i=\sqrt{n}} f(i) \leq 2^{n-\Omega(n)}$

then we partition the leaves into deep leaves and shallow leaves, and divide the shallow leaves into light leaves and heavy leaves, sum the cost of each type of leaves we can obtained the cumulative cost which is $O(2^{n-\Omega(n)})$.

d) if all clauses contain at most one universal literal

- if $u < \epsilon n$, then solve the QBF using brute force search
- else, make a DNF with 2^e conjuncts, which has $O(2^e u)$ clauses, using lemma can solved in $O^*(2^{u-\Omega(u/e)})$ time