The complexity of proving that a graph is Ramsey

Massimo Lauria, Pavel Pudlák, Vojtěch Rödl, and Neil Thapen

Presented by Pavel Veselý

A bit of proof complexity: Resolution

Resolution refutation of a propositional formula ϕ in CNF is a sequence of clauses which are either clauses of ϕ or clauses derived by the *resolution rule* from previous clauses: From clauses $A \vee x$ and $B \vee \neg x$ infer the clause $A \vee B$.

There is a resolution refutation of ϕ iff ϕ is unsatisfiable.

Definition. $L(\phi)$ = length (number of clauses) of a shortest resolution refutation of ϕ. ($L(\phi) = \infty$ for a satisfiable ϕ.)

c-Ramsey graphs

Definition. For a constant $c > 0$, an n-vertex graph is c-Ramsey if it does not contain a clique or an independent set of size $c \cdot \log n$.

We define a formula Ψ_G expressing a property that G is *not* c-Ramsey and show the following:

Theorem 1. *For every graph G it holds that $L(\Psi_G) \geq n^{\Omega(\log n)}$.*

W.l.o.g., there are $n = 2^k$ vertices in G.

From $L(\phi)$ to a game

Definition. The *width* a clause is the number of literals in it and the width of ϕ is the maximum width of its clause. Similarly, the width of resolution refutation Π is the maximum width of a clause in Π.

We define $W(\phi)$ to be the minimum width of a refutation of ϕ.

We use the following theorem by Ben-Sasson and Wigderson:

Theorem 2. *For every CNF formula ϕ with m variables and width w:*

$$L(\phi) \geq 2^{\Omega((W(\phi)-w)^2/m)}.$$

The lower bound on $W(\phi)$ follows from analysis of a game between Prover and Adversary:

- Prover claims that ϕ is unsatisfiable,
- Adversary claims to know a satisfying assignment.
- Prover asks Adversary for values of variables, but has a limited memory.
- Prover wins if the partial assignment in his memory falsifies a clause of ϕ.
- Adversary wins if it has a strategy to play forever.

Lemma 3. *Given an unsatisfiable ϕ, Prover needs only $W(\phi) + 1$ memory locations to win the game against any Adversary.*

We thus show a winning strategy for Adversary if Prover has small memory.

Definition. A pattern is a partial assignment to k variables. Formally, it is a string $p = p_1 \dots p_k \in \{*, 0, 1\}^k$. We say that p is consistent with binary string v (a vertex) if for all $i \in [k]$ either $p_i = v_i$ or $p_i = *$. The size $|p|$ of p is the number of bits set to 0 or 1. The empty pattern is a string of k stars.

Lower bounds for random graphs

Theorem 4. *If $G \sim \mathcal{G}(n, \frac{1}{2})$ is a random graph, then with high probability $L(\Psi_G) = n^{\Omega(\log n)}$.*

We use the following property P of random graphs: For any $U \subseteq V(G)$ with $|U| \leq \frac{1}{3}k$ and for any pattern p with $|p| \leq \frac{1}{3}k$, p is consistent with at least one vertex in $N(U) = \bigcap_{v \in U} N(v)$.

Lemma 5. *For $G \sim \mathcal{G}(n, \frac{1}{2})$ the property P holds with high probability.*

Lemma 6. *For any G with property P, there is an Adversary strategy which wins against any Prover who uses at most $\frac{1}{9}k^2$ memory locations.*

Lower bounds for c-Ramsey graphs

Definition. Given sets $A, B \subseteq V(G)$ we define their mutual density by

$$d(A, B) = \frac{|E(A, B)|}{|A| \cdot |B|}$$

where $E(A, B)$ is the set of edges with one endpoint in A and the other in B.

We use the following property of c-Ramsey graphs due to Prömel and Rödl:

Lemma 7. *There exists constants $\beta > 0, \delta > 0$ such that if G is a c-Ramsey graph, then there is a set $S \subseteq V(G)$ with $|S| \geq n^{3/4}$ such that for all $A, B \subseteq S$, if $|A|, |B| \geq |S|^{1-\beta}$ then $\delta \leq d(A, B)$.*

We derive the following property which we use instead of P:

Corollary 8. *Let $X, Y_1, Y_2, \dots, Y_r \subseteq S$ be such that $|X| \geq rm^{1-\beta}$ and $|Y_1|, \dots, |Y_r| \geq m^{1-\beta}$. Then there exist $v \in X$ such that $d(v, Y_i) \geq \delta$ for each $i = 1, \dots, r$.*

Lemma 9. *There is a constant $\varepsilon > 0$, independent of n and G, such that there exists a strategy for the Adversary in the game which wins against any Prover who is limited to $\varepsilon^2 k^2$ memory locations.*