DD2445 COMPLEXITY THEORY: LECTURE 22

RECAP OF LAST LECTURE

Proof complexity: How to prove that CNF formulas are unsatisfiable

Resolution refutation of F

Sequence $\pi = (C_1, C_2, \ldots, C_L)$ such that C_L is empty clause $\perp$ and for every $C_i \in \pi$ it holds that

a) $C_i \in F$ (axiom), or

b) C_i derived from C_j, C_k, $j,k < i$ by RESOLUTION RULE

$$\frac{B \vee x \qquad C \vee \bar{x}}{C \vee D}$$

Length/size of refutation = # clauses L

Our goal is to prove following theorem

THEOREM 1 (informal). Clique-colouring formulas expressing that there exist n-vertex graphs that are $(n-1)$-colourable but contain n-cliques are exponentially hard to refute for resolution.

Follows from:

1. Monotone circuit lower bound for clique
2. INTERPOLATION technique

MI II

THEOREM 2 [Pudlák '97]

Suppose $A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r})$ is an unsatisfiable CNF formula over disjoint sets of variables $\vec{p}, \vec{q}, \vec{r}$.

Let $\pi : A \wedge B \vdash \bot$ be a resolution refutation in length L. Then the following holds:

(1) There is an INTERPOLATING CIRCUIT $I(p)$ of size $O(L)$ such that
   (a) $I(p) = 0 \Rightarrow A(p, \vec{q})$ unsatisfiable
   (b) $I(p) = 1 \Rightarrow B(p, \vec{r})$ unsatisfiable

(2) From π one can construct resolution refutation
   (a) $\pi_A : A(p, \vec{q}) \vdash \bot$ if $I(p) = 0$
   (b) $\pi_B : B(p, \vec{r}) \vdash \bot$ if $I(p) = 1$
   in both cases of length $\leq L$

(3) If $\vec{p}$-variables occur only positively in $A(\vec{p}, \vec{q})$ or only negatively in $B(\vec{q}, \vec{r})$, then the circuit $I(p)$ can be made monotone

---

The clique-colouring formula lower bound in Thm 1 follows immediately from Thm 2, as argued last time, so today we focus on Thm 2

Build circuits with $\wedge, \vee$, and sel-gates for simplicity.

$$\text{sel}(x,y,z) = \begin{cases} y & \text{if } x = 0 \\ z & \text{if } x = 1 \end{cases}$$

Proof plan

First do part ②.

Then use ② to get ①

Maybe skip details on ③ (but they are in the LaTeX'ed lecture notes)

Key definitions (for proof):

g-clause: Clause C over variables $\vec{g}$ derivable from $A(g, \vec{g})$

r-clause: Clause C over variables $\vec{r}$ derivable from $B(g, \vec{r})$

Trivially true clause $\underline{1}$ considered to be both g-clause and r-clause.

Inductive proof of part ② From $\pi = (C_1, C_2, \ldots, C_L)$ construct sequence of clauses $\tilde{\pi} = (\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_L)$ which will contain candidate derivations π_A and π_B

Inductive hypothesis

P.1. $\tilde{C}_i$ is a g-clause or an r-clause

Write type $(\tilde{C}_i) = g / r$

P2. $\tilde{C}_i = 1$ only if $C_i \wedge \rho = 1$

If $\tilde{C}_i \neq 1$, then $\tilde{C}_i \subseteq C_i$

P3. If $\tilde{C}_i = 1$ then there is an associated axiom clause $\mathcal{E}_i$ such that

- $\mathcal{E}_i$ satisfied by $\rho(a) = 1$ for $a \in C_i \wedge \mathcal{E}_i$
- $\mathcal{E}_i \in A(\vec{p}, \vec{q})$ if type $(\tilde{C}_i) = q$
- $\mathcal{E}_i \in B(\vec{p}, \vec{r})$ if type $(\tilde{C}_i) = r$

Think of $\mathcal{E}_i$ as justification or excuse why we choose $\tilde{C}_i = 1$

This book-keeping is important for the monotonicity in part 3 (but we won't have time to discuss this in detail, so will not pay too much attention to $\mathcal{E}_i$)

Base case $C_i \in A(\vec{p}, \vec{q}) \wedge B(\vec{p}, \vec{r})$

Set $\tilde{C}_i = C_i \wedge \rho$

$\mathcal{E}_i = C_i$ if justification needed

type $(\tilde{C}_i) = q$ if $C_i \in A$, $= r$ if $C_i \in B$.

Inductive step

$C_i = C \vee D$ derived by resolution rule

$$\frac{C \vee x \qquad D \vee \bar{x}}{C \vee D}$$

from $C_j = C \vee x$

$C_k = D \vee \bar{x}$ $\quad j, k < i$

Have already constructed $\tilde{C}_j$ and $\tilde{C}_k$

$x \in \vec{p} \cup \vec{q} \cup \vec{r}$. Case analysis: | MIU

Case 1 ($x \in \vec{p}$)

If $\rho(x) = 0$, set $\tilde{C}_i = \tilde{C}_j$

$type(\tilde{C}_i) = type(\tilde{C}_j)$

$\mathcal{E}_i = \mathcal{E}_j$ if needed

P1 clearly OK

If $\tilde{C}_i \not\models \rho \neq 1$, then

$\tilde{C}_i \not\models \subseteq C_j \setminus \{a, \bar{a} \mid a \in \rho\}$

$\subseteq C \setminus \{a, \bar{a} \mid a \in \rho\}$

$\subseteq (C \vee D) \setminus \{a, \bar{a} \mid a \in \rho\}$

$= C_i \setminus \{a, \bar{a} \mid a \in \rho\}$

If $\tilde{C}_i = 1$, then $C \wedge \rho = 1$ since $\rho(x) = 0$, so $C_i \wedge \rho = (C \vee D) \wedge \rho = 1$ and $\mathcal{E}_i = \mathcal{E}_j$ works as justification axiom.

P2 & P3 OK

If $\rho(x) = 1$, set $\tilde{C}_i = \tilde{C}_k$

$type(\tilde{C}_i) = type(\tilde{C}_k)$

$\mathcal{E}_i = \mathcal{E}_k$ if needed

Argument same as for $\rho(x) = 1$.

Case 2 ($x \in \vec{q}$)

Divide into subcases depending on types of $\tilde{C}_j$ and $\tilde{C}_k$

(a) If one of $\tilde{C}_j$ and $\tilde{C}_k$ is r-clause, set $\tilde{C}_i$ to that clause (choose arbitrarily if both are r-clauses).
Set type($\tilde{C}_i$) = r — P1 clearly OK
Copy justification clause to $\tilde{C}_i$ if needed

Observe:
- $\tilde{C}_i$ does not contain any $\vec{q}$-variables (by IH)
- $x \in \vec{q}$ only variable that disappears in resolution step

therefore P2 & P3 OK.

(b) If $\tilde{C}_j$ or $\tilde{C}_k$ is q-clause not containing x (e.g., if $= 1$) Let $\tilde{C}_i$ = such q-clause without x (choose arbitrarily if both qualify)
Copy justification clause to $\tilde{C}_i$ if needed.

Note that since no variable in $\vec{p}$ disappears in resolution step, my (needed is) justification clause still OK.

(c) If $\tilde{C}_j$ or $\tilde{C}_k$ q-clause not containing $\bar{x}$ let $\tilde{C}_i$ be such q-clause. Argue as in (b).

(d) If none of previous cases apply, then we have g-clauses

$\tilde{C}_j = \tilde{C}_j' \vee x \qquad \tilde{C}_k = \tilde{C}_k' \vee \bar{x}$

both nontrivial (i.e., $\neq 1$)

Let $\tilde{C}_i$ resolvent of these clauses.

with type $(\tilde{C}_i) = g$

P1 Resolvent of two g-clauses $\Rightarrow$ g-clause

P2 $\tilde{C}_i = \tilde{C}_j \vee \tilde{C}_k \setminus \{x, \bar{x}\}$

$\subseteq (C_j \vee C_k \setminus \{a, \bar{a} \mid a \in g\}) \setminus \{x, \bar{x}\}$

$= C_i \setminus \{a, \bar{a} \mid a \in g\}$

P3 Vacuously OK since $\tilde{C}_i \neq 1$.

Case 3 ($x \in \vec{r}$)

Analogous.

If one of $\tilde{C}_j$ and $\tilde{C}_k$ is g-clause, copy that clause. Otherwise copy or construct r-clause. Details are left as an exercise.

Part 2 Now follows by induction principle. Final clause $C_L = \bot$ gets classified as g-clause or r-clause $\tilde{C}_L \subseteq C_L = \bot$ means $\tilde{C}_L = \bot$. Derived only from $A(g, \vec{g})$ if g-clause or $B(g, \vec{r})$ if r-clause

Proof of part (1)

Go over $\pi = (C_1, C_2, \ldots, C_L = 1)$

Look at construction of $\tilde{\pi} = (\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_L = 1)$

For each C_i construct subcircuit with output gate $\tilde{v}_i$

computing

$$type(\tilde{C}_i) = \begin{cases} 0 & \text{if } \tilde{C}_i \text{ } q\text{-clause} \\ 1 & \text{if } \tilde{C}_i \text{ } r\text{-clause} \end{cases}$$

Just inspect proof of part (2)

Base case

$C_i \in A(\vec{p}, \vec{q}) \Rightarrow \tilde{v}_i$ constant 0

$C_i \in B(\vec{p}, \vec{r}) \Rightarrow \tilde{v}_i$ constant 1

Induction step

Again case analysis over resolution variable $x \in \vec{p} \cup \vec{q} \cup \vec{r}$

Case 1 ($x \in \vec{p}$)

$type(\tilde{C}_i) = sel(x, type(\tilde{C}_j), type(\tilde{C}_k))$

$= sel(x, v_j, v_k)$

Case 2 ($x \in \vec{q}$)

$type(\tilde{C}_i) = 1$ if one of $\tilde{C}_j, \tilde{C}_k$ has type r

otherwise $= 0$

$type(\tilde{C}_i) = type(\tilde{C}_j) \vee type(\tilde{C}_k)$

$= v_j \vee v_k$

Case 3 ($x \in \vec{r}$)

type $(\tilde{C}_i) = v_j \wedge v_k$

Details left as exercise

By the induction principle, final gate v_L will compute type$(\tilde{C}_L)$ = type$(\perp)$

Yields correct interpolating circuit

Proof of part (3)

Selector gates are non-monotone. Need to get rid of them

If $\vec{p}$-variables appear only positively in $A(\vec{p}, \vec{q})$, solution is described in LaTeXed notes.

$\vec{p}$-variables only negative in $B(\vec{p}, \vec{r})$: Left as exercise.

This is where we use the "justification axioms" $\mathcal{E}_i$

///

CLIQUE FORMULA LOWER BOUND

kC I

Now know clique-colouring formula hard for resolution

Hard to rule out that unknown n-vertex graph has k-clique and is $(k-1)$-colourable

But what if graph = assignment $\rho : \{0,1\}^{\binom{n}{2}} \to \{0,1\}$ fixed?

Clique problem hard

Trivial algorithm $\binom{n}{k} n^{O(1)}$ time

Can do slightly better, but $n^{O(k)}$ upper bound best known

Believed to be hard

- even to approximate size of largest clique
- even on average for random graphs

$n^{\Omega(k)}$ believed to be correct lower bound

True if SAT requires exponential time (Exponential Time Hypothesis or ETH)

What about unconditional results?

For applied algorithms used in practice?

New result (Nov 2017) by

Atserias, Bonacina†, de Rezende,* Lauria†, Nordström, & Razborov

($\dagger$) former postdocs at KTH

($*$) current PhD student at KTH

THEOREM (Slightly informal)
For $k << \sqrt{n}$ it holds for a randomly sampled n-vertex graph G that
a) G does not have k-clique almost surely
b) Regular resolution requires length $n^{\Omega(k)}$ to prove this almost surely

---

A sequence of events E_n happen (asymptotically) almost surely (a.a.s.) if $\lim_{n\to\infty} Pr[E_n] = 1$.

COROLLARY
Essentially all state-of-the-art clique algorithms require time $n^{\Omega(k)}$ for randomly sampled graphs without k-cliques

(Because the reasoning that they use can be captured by regular resolution. Requires an argument.)

Need to make precise:
(i) "random" graphs
(ii) "regular" resolution
(iii) what formula is being refuted?

(i) Erdős-Rényi random graph $G(n,p)$

n-vertex (undirected) graph; think of $V = [n]$

For every potential edge (i,j), $i<j$, flip independent ~~edge~~ coin and include edge with probability p.

For $X \subseteq V$, $|X| = k$,

$$\Pr[X \text{ forms clique}] = p^{\binom{k}{2}}$$

Expected # cliques =

$$\binom{n}{k} \cdot p^{\binom{k}{2}} \qquad (*)$$

(by linearity of expectation)

For $p = n^{-2/(k-1)}$ get $(*) \approx 1$

For $p = n^{-2\eta/(k-1)}$ (η constant, $\eta < 1$)

$G \sim G(n,p)$ very unlikely to contain k-clique (but contains many cliques of size $\Omega(k)$)

(ii) Regular resolution

General resolution — general proof DAG without restrictions

Tree-like resolution — proof DAG is binary tree

Equivalent to decision tree for falsified clause search problem

Regular resolution:

General DAG-like proof
But on each path every variable resolved only once (once variable is eliminated, not introduced again)

Regular resolution exponentially stronger than tree-like resolution
(e.g., xorified pebbling formulas)

Regular resolution exponentially weaker than general resolution
(but separating formulas hard to find and quite contrived)

Regular resolution = Read-once branching program (ROBP)

ROBP for CNF formula F

One source = start node
Every sink labelled by clause $C \in F$
Every non-sink queries variable x
Outgoing edges labelled 0 and 1

Any assignment α defines path to sink

Branching program solves (falsified clause search problem for) F if for any α reach sink labelled by C s.t. $\alpha(C) = 0$

kC V

Read-once: along any path, query every variable at most once.

To get ROBP from regular resolution refutation, make 1 start node, flip direction of all edges, and let every node query variable resolved over

(Requires slightly more formal argument that we will skip)

Proof idea: "Bottleneck counting argument"

(1) Define random walk R_α in ROBP

(2) Show that any random walk passes through special type of "bottleneck node" w

(3) Prove that for any fixed bottleneck node w $\Pr[R_\alpha \text{ passes through } w] \lesssim n^{-k}$

(4) But

$$1 = \Pr[R_\alpha \text{ passes through } \underline{\text{some}} \text{ bottleneck}]$$
$$\leq \sum_{\text{bottlenecks } w} \Pr[R_\alpha \text{ passes through } w]$$
$$\lesssim (\#\text{ bottlenecks } w) \cdot n^{-k}$$

Implying that

$$\text{proof length} \geq \#\text{ bottlenecks} \gtrsim n^k$$

PREVIOUSLY KNOWN

Tree-like resolution

Complete $(k-1)$-partite graphs yield maximally hard formulas

General resolution

[Beame, Impagliazzo, Sabharwal '07]
lower bound $\exp(n^{\varepsilon})$ for $k = n^{\delta}$, $\delta \geq 5/6$

Nothing previously known even for regular resolution for smaller k.

We get regular resolution lower bound $n^{\Omega(k)}$ for $k \leq \sqrt{n}$

Should be true also for general, unrestricted resolution, but this remains open

Next time

Show $n^{\Omega(k)}$ lower bound for read-once branching program solving falsified clause search problem for clique formula for $G(n,p)$ random graph.