AMSI Summer School 2022

Computational Complexity

Pre-course Quiz

September 29, 2021

1 Quizzes

Notation. For $n \in \mathbb{N}$, $[n] := \{1, 2, \ldots, n\}$.

1. Let $G = (V, E)$ be a simple undirected (finite) graph: the vertex set V is a finite set and the edge set E consists of some size-2 subsets of V, i.e. $E \subseteq \{\{v, v'\} : v, v' \in V, v \neq v'\}$.

   A subset of vertices, $W \subseteq V$, is a *vertex cover* in G if every edge in E is incident to at least one vertex in W.

   A subset of vertices, $U \subseteq V$, is an *independent set* in G if no two vertices in U share an edge.

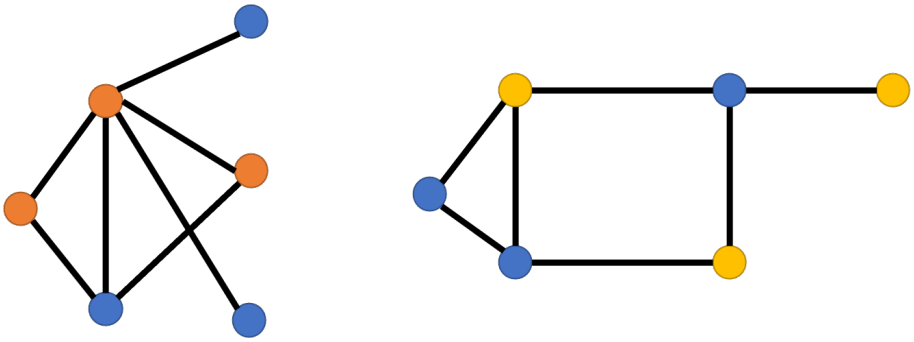

Figure 1: Left: Orange vertices form a vertex cover.
Right: Yellow vertices form an independent set.

   Let $n = |V|$, the number of vertices of G. For $k \in [n]$, show that G has a size-k independent set if and only if it has a size-$(n - k)$ vertex cover.

2. Recall that $\mathbb{Q}[x, y]$ is the ring of rational polynomials in two variables x and y. Let $f(x, y) \in \mathbb{Q}[x, y]$ be a *non-zero* polynomial of degree d. Let $S = [2d]$. Prove that

$$\Pr_{(a,b) \in S \times S}[f(a, b) = 0] \leq 1/2.$$

   That is, when a and b are sampled from S independently and uniformly at random, the probability of $f(a, b) = 0$ is at most $1/2$.

3. A *boolean function* is a function of the form $f : \{0,1\}^n \to \{0,1\}$.

   A *boolean circuit* in n variables $x_1, \ldots, x_n$ of size ℓ is a sequence $(s_1, \ldots, s_\ell)$, where each s_i is in one of the following forms:

   - $\mathrm{op}_1 \vee \mathrm{op}_2$, where op_1 (resp. op_2) is either x_j for some $j \in \{1, \ldots, n\}$, or s_k for some $k \in \{1, \ldots, i-1\}$, or $b \in \{0,1\}$.

     This $\vee$ represents the boolean *or* operation, so $\vee : \{0,1\} \times \{0,1\} \to \{0,1\}$ is defined as $\vee(x, y) = 0$ if and only if $x = y = 0$.
   - $\mathrm{op}_1 \wedge \mathrm{op}_2$, where op_1 (resp. op_2) is either x_j for some $j \in \{1, \ldots, n\}$, or s_k for some $k \in \{1, \ldots, i-1\}$, or $b \in \{0,1\}$.

     This $\wedge$ represents the boolean *and* operation, so $\wedge : \{0,1\} \times \{0,1\} \to \{0,1\}$ is defined as $\wedge(x, y) = 1$ if and only if $x = y = 1$.
   - $\neg\mathrm{op}$, where op is either x_j for some $j \in \{1, \ldots, n\}$, or s_k for some $k \in \{1, \ldots, i-1\}$, or $b \in \{0,1\}$.

     This $\neg$ represents the boolean *not* operation, so $\neg : \{0,1\} \to \{0,1\}$ is defined as $\neg(x) = 1$ if and only if $x = 0$.

   (a) What is the number of boolean functions of the form $f : \{0,1\}^n \to \{0,1\}$?

   (b) What is the number of boolean circuits in n variables of size ℓ? If you wish, you can just give an estimation of the upper bound.

   (c) Conclude that when $n \geq 10$, there exists a boolean function $f : \{0,1\}^n \to \{0,1\}$ that cannot be computed by any boolean circuit in n variables of size n^2.

2 Background of the quizzes

1. Given a graph G, we can formulate two algorithmic problems. The first one is to compute an independent set of size s. The second one is to compute a vertex cover of size t. This quiz is used to establish that these two algorithmic problems are "equivalent", i.e. if one can be solved efficiently, then so can the other.

2. You may already realise that this holds for polynomials in any number of variables. That is, let $f(x_1, \ldots, x_n) \in \mathbb{Q}[x_1, \ldots, x_n]$ be a *non-zero* polynomial in n variables of degree d. Then we have that $\Pr_{(a_1,\ldots,a_n)\in S\times\cdots\times S}[f(a_1, \ldots, a_n) = 0] \leq 1/2$, where a_i's are selected at random independently and uniformly from $[2d]$.

   The above is known as Schwartz-Zippel lemma or the polynomial identity testing lemma. It gives an efficient randomised (or probabilistic) algorithm to decide whether a polynomial is the zero polynomial, or whether two polynomials are the same.

3. This quiz actually shows that polynomial-size boolean circuits cannot compute all boolean functions. This observation goes back to Claude Shannon, the founder of information theory. It may be viewed as a finite version of the counting argument to show that there exist problems that cannot be solved (i.e. not recognisable) by Turing machines.

3 Answers to the quizzes

1. For the only-if direction, suppose $U \subseteq V$ is an independent set of size k in G. Let $W = V \setminus U$. We claim that W is a vertex cover of size $(n-k)$. To see this, consider any edge $\{v, v'\} \in E$. If neither v nor v' is in W, then $\{v, v'\}$ is an edge connecting two vertices in U, contradicting that U is an independent set.

   The if direction follows the same proof strategy. That is, suppose W is a vertex cover of size $(n-k)$ in G. Let $U = V \setminus W$, so $|U| = k$. We can then verify that U is an independent set in G.

2. Recall that a non-zero univariate polynomial $f(x)$ of degree d has at most d roots. This will be used repeatedly in the following.

   To prove the claim, we only need to show that the number of zeros of $f(x, y)$ in $S \times S$ is upper bounded by $2d^2$.

   Write $f(x, y) \in \mathbb{Q}[x, y]$ as

   $$f(x, y) = g_0(y) \cdot x^d + g_1(y) \cdot x^{d-1} + \cdots + g_{d-1}(y) \cdot x + g_d(y),$$

   where $g_i(y) \in \mathbb{Q}[y]$ is of degree i.

   Let $k \in [d]$ be the *minimum* integer such that $g_k(y)$ is not the zero polynomial. That is, $g_0(y), \ldots, g_{k-1}(y)$ are all zero. This k exists because $f(x, y)$ is not the zero polynomial. So $f(x, y) = g_k(y) \cdot x^k + \cdots + g_{d-1}(y) \cdot x + g_d(y)$.

   Let $B_1 = \{b \in [2d] : g_k(b) = 0\}$, and $B_2 = [2d] \setminus B_1$. We note the following:

   - As g_k is of degree k, $|B_1| \leq k$.
   - If $b \in B_2$, then $g_k(b) \neq 0$, so $f(x, b)$, as a polynomial in $\mathbb{Q}[x]$ of degree $d-k$, has at most $d-k$ zeros.

   So the number of zeros of f in $[2d] \times [2d]$ is at most $|B_1| \cdot 2d + (2d - |B_1|) \cdot (d-k) \leq 2kd + 2d(d-k) \leq 2d^2$. The result then follows.

3. (a) For each boolean string $\vec{b}$ of length n (that is, $\vec{b}$ is an element in $\{0,1\}^n$), there are two possible values for f on $\vec{b}$. There are 2^n boolean strings of length n. So the number is

   $$2^{2^n}.$$

   (b) For each s_i, there are three types for the operation (one of $\{\vee, \wedge, \neg\}$), and $n + i + 1$ possibilities for each operand. So there are at most $2(n+i+1)^2 + (n+i+1) = (n+i+1)(2n+2i+3)$ possibilities for s_i. The number of size-ℓ boolean circuits is therefore

   $$\prod_{i=1}^{\ell} \big((n+i+1)(2n+2i+3)\big).$$

   A convenient upper bound is then

   $$(n+\ell+1)^{\ell}(2n+2\ell+3)^{\ell} \leq (2n+2\ell+3)^{2\ell}.$$

   (c) When $\ell = n^2$, then the above upper bound becomes $2^{2n^2 \cdot \log_2(2n+2n^2+3)}$. Recall that the number of boolean functions is 2^{2^n}. So when n is large enough, $2n^2 \cdot \log_2(2n+2n^2+3)$ is less than 2^n, so the desired conclusion follows. (You may work out how large n needs to be for $2n^2 \cdot \log_2(2n+2n^2+3) < 2^n$, but that's probably not the most interesting part.)