CS388C: LECTURE 1

KURTIS DAVID

1. Circuits

Definition 1.1. Circuit (**ckt**) - a function $f : \{0,1\}^n \to \{0,1\}$. i.e. any function of n bits is computable by a **ckt** of *size* $\leq n \cdot 2^n$

Definition 1.2. Size (of a ckt) - # of nodes in a ckt. e.g. and, or, not gates.

Question 1.3. *Do $\exists$ functions requiring exponential size ckts?*

Answer: Yes.

Solution.

- Many possible $f : \{0,1\}^n \to \{0,1\}$
  - 2^n inputs, 2 outputs $\Rightarrow 2^{2^n}$ possible circuits
- "Few" ckts of small size s (few in comparison to above)

$$\text{\# ckts of size } s \leq (\text{\# possible ways to add a new gate})^s$$

But there are 3 types of nodes (add/or/not) and $\binom{s}{2}$ nodes to choose from when adding these gates so # possible ways = $\frac{3s(s-1)}{2}$.

$$\therefore \text{\# ckts of size } s \leq (2s^2)^s \leq s^{3s}$$

If we say $s = \frac{2^n}{3n}$ then we end up with ... $\leq \frac{2^n}{3n}^{\frac{2^n}{n}} = \frac{1}{3n}^{\frac{2^n}{n}} 2^{n\frac{2^n}{n}}$
$\therefore$ most fns require ckts of size $\geq \frac{2^n}{3n}$

■

2. Languages

Definition 2.1. Language - a subset of $\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$ mapped to $\Sigma = \{0,1\}$. i.e. $f : \Sigma^* \to \Sigma$ (all lengths of binary strings!)

To show $NP \neq P$, it suffices to show a language in NP requires ckt families of size $> n^c \; \forall c$.
Open Q: Find a language in NP requiring ckts of size $\omega(n)$

3. Basic Counting

Question 3.1. *How many ways can r objects be placed in n bins?*

Solution. Assume bins are distinguishable.

1a. Let objects be distinguishable.
$\Rightarrow n^r$ ways (since each object can be placed in any of the n bins).

1b. Now assume 1a AND ≤ 1 object/bin.
$\Rightarrow n\mathrm{P}r = \frac{n!}{(n-r)!}$. This is because after placing an object into some distinguishable bin, you can no longer choose this bin again.
(so the product would look like $n \cdot (n-1) \cdot (n-2) \cdot ...$)

2a. Let objects be indistinguishable.

2b. If ≤ 1 object/bin, simple. $\binom{n}{r}$, since we have repeated permutations.

For 2a, we must be more careful. One "trick" is to use a sticks argument – suppose all r objects are layed across a floor – to create bins, simply add *boundaries* before or after each object.
To have n bins, we will have $n-1$ boundaries. There exist at least r spaces, but because we can put all objects into the first bin, there must be at least $n-1$ spaces at the end to put all the boundaries at.
$\therefore$ there are $\binom{r+n-1}{n-1}$. ■

Remark. We only covered 4 situations, there is something called the **Twelvefold way** that includes when bins are indistinguishable, and if each bin has ≥ 1 object each.

Question 3.2. *How many triples (a, b, c) of nonnegative integers are there s.t. $a + b + c = 100$?*

Solution. This question can be reformulated as the previous, where we have $r = 100$ indistinguishable objects and $n = 3$ bins. Thus we have $\binom{102}{2}$ ways. ■

Question 3.3. *Suppose you can place n points on a circle s.t. no 3 chords between them meet at an interior point. How many interior intersection points are there?*

Solution.

Notice that an intersection point can be determine by 2 lines i.e. 4 points.
$\therefore$ in terms of n, we have $\binom{n}{4}$ ■

Question 3.4. *Show that # people who shake hands an odd # of times is even.*

Solution.

Create a handshake graph – let nodes be people, and edges represent a handshake between them.
Thus # handshakes $= \Sigma \deg(v) = 2 * edges$
But also # handshakes $= \Sigma_{\deg(v) \text{ odd}} \deg(v) + \Sigma_{\deg(v) \text{ even}} \deg(v)$ ■

4. Hypergraphs

Definition 4.1. Hypergraph - graph that contains (hyper)edges can contain an arbitrary # of nodes (instead of just 2)

Definition 4.2. k-uniform hypergraph - each hyperedge contains k nodes.
$deg(v) =$ # hyperedges containing v
$\Rightarrow k|\Sigma \mathrm{deg}(v)$