Computer Science Theory, Test 2 Review Problems
Prof. Toniann Pitassi

1. Answer True or False for each statement. No justification is needed.
   - T (a) $n = O(n^2)$
   - F (b) $n \log n = O(n)$
   - F (c) $n^n = O(2^n)$
   - T (d) Let A be mapping reducible to B. If B is decidable then A must be decidable.
   - F (e) Let A be mapping reducible to B. If A is decidable then B must be decidable.
   - F (f) If the complement of a language L is not recognizable then both L and $\neg L$ are not recognizable.
   - F (g) If A is NP-complete, $A \subseteq B$, and B is in NP then B is NP-complete.
   - F (h) If B is NP-complete and $A \subseteq B$ and A is in NP then A is NP-complete.

2. Let Double-CLIQUE denote the language consisting of all pairs (G, k) such that G is an undirected graph containing two disjoint cliques each of size k. Prove that Double-CLIQUE is NP-complete.

3. Prove that the following set is countable.
$$S = \{(i, j) \mid i \geq 0 \text{ and } j > i\}$$

4. Prove that the following set is [redacted]countable.
$$S = \{L \subseteq \{0,1\}^* \mid \text{the number of strings in } L \text{ is finite}\}$$

5. Prove that NP is closed under union. That is, for every $L_1, L_2 \in$ NP, $L_1 \cup L_2$ is also in NP.

6. Prove that NP is closed under concatenation.

7. Let L be the language consisting of all pairs $< M >$ such that M encodes a Turing machine and M accepts at least two inputs.
   - (a) Prove that L is recognizable.
   - (b) Prove that L is not decidable.

8. Recall that 3SAT is the set of all 3-CNF formulas ϕ such that ϕ is satisfiable. Let Search-3SAT be the following *search* problem: Given a 3CNF formula ϕ, output a satisfying assignment for ϕ if one exists, and otherwise output "ϕ is unsatisfiable". Prove that if 3SAT $\in$ P, then Search-3SAT can be solved in polynomial-time by a deterministic TM.

2. Double-Clique

(a) It is in NP.

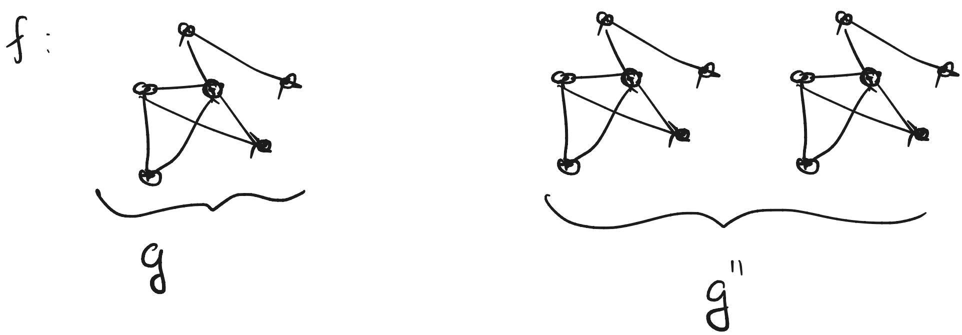
The verifier on input $((g,k),(V_1,V_2))$ checks that V_1, V_2 are disjoint subsets of the vertices of g and each have size k, and that g contains a clique on V_1 and on V_2.

(b) Double-clique is NP-hard

We give a polytime map reduction from Clique to Double-Clique

On input (g,k) to clique construct g' which contains 2 copies of g. Let $f(g,k) \to (g',k)$.

This mapping is polytime computable and g' contains ≥ 2 disjoint k-cliques iff g contains ≥ 1 k-clique



3. $S = \{(i,j) \mid i \geq 0, j > i\}$ is countable

For all $(i,j) \in S$, Let $f(i,j) \rightarrow 2^i \cdot 3^j$

This mapping is one-to one from S to $\mathbb{N}$ and therefore S is countable

4. $S = \{L \subseteq \{0,1\}^* \mid L$ has finite size$\}$ is countable.

To see this is countable we will give an enumeration of all elements of S (using dovetailing)

Let $x_1, x_2, \ldots$ be an enumeration of $\{0,1\}^*$.

For $i = 1, 2, \ldots$
  For $j = 1, \ldots, i$
    Enumerate all subsets of $\{x_1 \ldots x_i\}$ of size j

For any $L \in S$, Let x_{i^*} be the largest element in L (according to our ordering $x_1 < x_2 < \ldots$)

and say $|L| = k$. Then in the above loop, when $i = i^*$ and $j = k$, we will output L.

5. NP closed under union.

Let $L_1, L_2 \in NP$.

We want to show: $L_1 \cup L_2 \in NP$.

Let V_1 be a polytime verifier for L_1, V_2 for L_2.

Our verifier V for $L_1 \cup L_2$:

On input (w, c):

View $c = c_1, c_2$.

Run $V_1(w, c_1)$

Run $V_2(w, c_2)$

accept iff at least one of V_1, V_2 accepts.

6. NP closed under concatenation:

Let $L_1, L_2 \in NP$. We want to show

$L = \{w \mid w = w_1 w_2,\ w_1 \in L_1,\ w_2 \in L_2\} \in NP$

Let M_1 be a polytime NTM for L_1, M_2 for L_2.

Nondet polytime TM, M, for L:

On input w:

Use nondeterminism to guess a partition of $w = w_1 w_2$

Run M_1 on w_1

Run M_2 on w_2

Accept iff both accept

7. Let $L = \{\langle M\rangle \mid M \text{ accepts at least 2 inputs}\}$

(a) Prove L is r.e.

Let $x_1, x_2, \ldots$ be an enumeration of all inputs in Σ^*

Description of r.e. alg for L on input $\langle M\rangle$:

For $t = 1, 2, 3, \ldots$

- Simulate M on the first t inputs $(x_1, \ldots, x_t)$ for t steps each.
- If M accepts at least 2 of these inputs, halt and accept.
- Otherwise loop again.

(b) Show L is not recursive.

We will give a mapping reduction f from $HALT \to L$.

On input (M, x), Let $f(\langle M, x\rangle) \to \langle M'\rangle$ where M' is defined as follows.

M' on input w:

- Ignore input w and simulate M on x
- Halt and accept iff M halts on x

Correctness :

(1) $(M, x) \in \text{Halt} \Rightarrow f((M, x)) = \langle M' \rangle \in L$:

If M halts on x, then $\mathcal{L}(M')$ is all strings,
so $M' \in L$

(2) $(M, x) \notin \text{HALT} \Rightarrow f((M, x)) = M' \notin L$:

If M does not halt on x, then $\mathcal{L}(M') = \emptyset$
so $M' \notin L$

8. Search-3SAT : On input a 3CNF ϕ, output "UNSAT" if ϕ unsatisfiable; otherwise output a satisfying assignment.

We will show SEARCH-3SAT has a poly time alg if 3SAT $\in$ P.

<u>Algorithm for Search-3SAT</u> (assuming 3SAT $\in$ P):

On input ϕ:

1. Run 3SAT alg on ϕ
   If it rejects, halt + output "UNSAT"

2. Otherwise:

   Let α = empty partial assignment
   Let $\phi = \phi$
   For $i = 1, 2, 3, \ldots, n$ $\quad$ (n = # vars in ϕ)
   - Call 3SAT on $\phi_{x_i=1}$ $\quad$ ($\phi|_{x_i=1}$ is ϕ simplified by setting $x_i = 1$)
     - If it accepts:
       Let $\alpha = \alpha \cup \{x_i = 1\}$, $\phi = \phi|_\alpha$
     - Else (it rejects):
       Let $\alpha = \alpha \cup \{x_i = 0\}$, $\phi = \phi|_\alpha$

→ algorithm has n iterations, each loop takes polytime so overall runtime is poly in n

→ Correctness: follows since at every step, we are guaranteed at each iteration that some satisfying assignment exists that extends the current partial assignment α