Homework 4

Name: 方嘉聪 ID: 2200017849

Problem 1.(16 points). A cut in an undirected graph is a seperation of the vertices V into two disjoint subsets S and T. The size of a cut is the number of edges that have one endpoint in S and the other in T. Let

$$\textbf{MAX-CUT} = \{\langle G, k\rangle \mid G \text{ has a cut of size } k \text{ or more }\}.$$

a.) (6 points) Give a randomized algorithm that, given an graph G such that G has a cut of size k or more, outputs a cut of expected size at least $k/2$.

b.) (10 points) Show that **MAX-CUT** is **NP**-complete.(Hint: Show that $\neq$SAT $\leq_p$ **MAX-CUT**.)

◀

Answer. a.) 考虑如下的随机算法:

Input: Graph $G = (V, E)$.

Output: A cut (S, T) of expected size at least $k/2$.

1: Assign each vertex $v \in V$ to S or T randomly with equal probability.

2: **return** (S, T).

记得到的割为 C, 那么对于 $\forall (u, v) \in E$, $\mathbb{P}[(u, v) \in C] = 1/2$. 记示性随机变量 $\mathbf{1}_e$ 表示 e 是否在 C 中, 那么:

$$E[|C|] = \sum_{e\in E} E[\mathbf{1}_e] = \sum_{e\in E} \mathbb{P}[e \in C] = \frac{1}{2}\sum_{e\in E} 1 = \frac{1}{2}|E| \geq \frac{k}{2}.$$

因此 C 的期望大小至少为 $k/2$, 证毕.

b.) 首先证明 **MAX-CUT** $\in$ **NP**. 对于 $\forall \langle G, k\rangle \in$ **MAX-CUT**, 存在一个证书 u 表示一个割 S, T. 用一个多项式时间的验证机 M 验证 (S, T) 是否合法的割且大小不小于 k, 即:

$$\langle G, k\rangle \in \textbf{MAX-CUT} \iff \exists u \ s.t. \ M(G, k, u) = 1.$$

u 显然是多项式长度的, 故 **MAX-CUT** $\in$ **NP**.

下面证明 **MAX-CUT** $\in$ **NP**-hard. 对 $\forall \psi \in \neq$SAT, 设 ψ 是一个 n 个变量 m 个子句的 3CNF, 那么我们构造一个图 $G = (V, E)$ 如下:

1. 对每个变量 x_i, 分别添加 $3m$ 个标号为 x_i 和 $\neg x_i$ 的节点.
2. 所有标号为 x_i 与 $\neg x_i$ 的节点之间两两连边.
3. 对于每个子句 C_j, 若 $C_j = (x_i \vee x_k \vee x_l)$, 那么在 x_i, x_k, x_l 对应标号的节点之间连边, 注意不要重复使用一个节点.

注意到取 $3m$ 个 $x_i/\neg x_i$ 足以保证第 3 步中有足够的冗余节点用来连接每个子句.
我们来证明如下结论:

$$\psi \in \neq \text{SAT} \iff \langle G, 9nm^2 + 2m\rangle \in \textbf{MAX-CUT}.$$

1.) $\Rightarrow$: 若 $\psi \in \neq$SAT, 考虑 ψ 的任意一个合法赋值 A, 令 $S = \{x \mid x = 1 \text{ in assignment } A\}$, $T = \{x \mid x = 0 \text{ in assignment } A\}$. 考虑割 (S, T), 注意到 $x_i, \neg x_i$ 不能同时在 S/T 中, 这一部分的割的大小为 $n \cdot (3m)^2 = 9nm^2$. 而对于每个子句 C_j, 由 $\neq$-assignment 的定义, C_j 中至少有一个变量取值为 0, 那么每个子句对应的三角形有两条边在割中, 这一部分的割的大小为 $2m$. 因此总的割的大小为 $9nm^2+2m$. 即 $\langle G, 9nm^2 + 2m\rangle \in$ **MAX-CUT**.

2.) $\Leftarrow$: 若 $\langle G, 9nm^2 + 2m\rangle \in$ **MAX-CUT**. 那么 G 的割的大小至少为 $9nm^2 + 2m$. 首先说明标号相同的节点分别位于 S, T 中, 否则考虑最简单的情况, 如果标号为 x_i 的节点中有 $3m - 1$ 个在 S 中, 1 个在 T 中, 那么此时割的大小不超过 $2m + (n - 1)9m^2 + 3m(3m - 1) < 9nm^2 + 2m$, 矛盾. 再说明对于每个子句产生的三角形, 不可能有存在割使得三角形的三个边都在割中. 进而其他条件给定, 如果割只包含一条边, 那么总的割大小严格小于 $9nm^2 + 2m$, 矛盾. 因此每个子句对应的三角形恰有两条边在割中. 那么我们令 $x_i = 1$ 当且仅当 x_i 对应的节点在 S 中, 就得到了一个合法的 $\neq$-assignment. 因此 $\psi \in \neq$SAT.

综上所述, $\neq$SAT $\leq_p$ **MAX-CUT**, 即 **MAX-CUT** 是 **NP**-complete. $\triangleleft$

Problem 2.(16 points). A language is called unary if every string in it is the form 1^i (the string of i ones) for some $i > 0$. Show that if there exists an **NP**-complete unary language, then **NP** = **P**.

Hint: If there is a n^c time reduction from 3SAT to a unary language L, then this reduction can only map size n instances of 3SAT to some string of the form 1^i where $i \leq n^c$. Use this observation to obtain a polynomial-time algorithm for 3SAT using the downward self-reducibility argument of Theorem 2.18 in our textbook. $\blacktriangleleft$

Answer. 思路: 用类似于课本中 Theorem 2.18 的递归迭代方法, 来证明 SAT $\in$ **P**, 进而 **NP** = **P**.

设 L 是一个 **NP**-complete 的 unary 语言, 那么 L 中的每个字符串都是形如 1^i 的 (不妨令 $I = \{i\}$). 设存在一个时间复杂度为 n^c 的规约 f 将 SAT 映射到 L, 即

$$\forall \psi \in \text{SAT}, f(\psi) = 1^i, \quad \text{where size of } \psi \text{ is } n,\ i \leq n^c.$$

考虑如下的算法 A:

Input: a CNF ψ_n with $|\psi_n| = n$, and the array I represents whether $1^i \in L$ or not.
Output: 1 if ψ_n is satisfiable, 0 otherwise.

1: **if** $n = 1$ **then**
2: Compute ψ directly and return the result.
3: Compute $f(\psi_n) = 1^i$ in n^c time.
4: **if** $I[i]$ is already computed **then**
5: **return** $I[i]$.
6: **else**
7: Recursively compute $A(\psi_n \mid x_1 = 0)$, where $\psi_{n'} = (\psi_n \mid x_1 = 0)$ means the simplified formula with assignment $x_1 = 0$. Notice that $|\psi_{n'}| < |\psi_n|$.
8: **if** $A(\psi_n \mid x_1 = 0) = 1$ **then**

9: $I[i] \leftarrow 1$, then **return** 1.
10: **else**
11: $r \leftarrow A(\psi_n \mid x_1 = 1)$
12: $I[i] \leftarrow r$, then **return** r.

时间复杂度分析: 对每个 i, 我们会创建 2 个分支 (这是由于 1^i 是否属于 L 我们通过数组 I 存储下来, 通过两次调用 $A(x_1 = 0/1)$ 即可得到, 后续直接查询即可), 由于 $i \leq n^c$, 那么最多有 $2n^c$ 个分支. 而每个分支中只需要调用一次规约函数 f(复制 I 后传参), 故时间复杂度为 $O(n^c)$. 那么总的时间复杂度:

$$T(n) \leq 2n^c \cdot O(n^c) = O(n^{2c}).$$

故 SAT $\in$ **P**, 进而 **NP** = **P**. 证毕. ◁

Problem 3.(16 points). Prove that the following language SPACETM is **PSPACE**-complete:

$$\textbf{SPACETM} = \{\langle M, w, 1^n\rangle \mid \text{ DTM } M \text{ accepts } w \text{ in space } n \}.$$

◀

Answer. 先证明 **SPACETM** $\in$ **PSPACE**. 考虑如下的算法:

Algorithm 3 Universal DTM U to simulate M on input w

Input: $\langle M, w, 1^n\rangle$

1: U simulates M on input w. If M uses more than n space, U rejects immediately.
2: **if** M accepts w in space $O(n)$ **then**
3: **return** Accept.
4: **else**
5: **return** Reject.

那么 U 的空间复杂度是 Poly($|\langle M, w, 1^n\rangle|$) 的, 故 **SPACETM** $\in$ **PSPACE**.

下面证明 **SPACETM** 是 **PSPACE**-hard 的. $\forall A \in$ **PSPACE**, 那么存在时间复杂度为 $p(n)$ 的 DTM M 使得 $A = L(M)$, 其中 $p(n)$ 是多项式函数, n 为输入长度. 考虑如下的映射:

$$f : w \to \left\langle M, w, 1^{p(|w|)} \right\rangle.$$

f 显然是多项式时间的. 且有:

$$w \in A \iff f(w) \in \textbf{SPACETM}$$

因此 $\forall A \in$ **PSPACE**, $A \leq_p$ **SPACETM**, 即 **SPACETM** 是 **PSPACE**-hard 的.

综上所述, **SPACETM** 是 **PSPACE**-complete 的. 证毕. ◁

Problem 4.(16 points). The class **DP** is defined as the ser of language L for which there are two language $L_1 \in$ **NP**, $L_2 \in$ **coNP** such that $L = L_1 \cap L_2$. (Don't confuse **DP** with **NP** $\cap$ **coNP**, which may seem superficially similar.) Show that:

a.) (6 points) EXACT INDSET $\in$ **DP**.

b.) (10 points) Every language in **DP** is polynomial-time reducible to EXACT INDSET.

◀

Answer. **Note that:**

$$\text{EXACT INDSET} = \{\langle G, k\rangle \mid \text{the largest independent set in } G \text{ has size exactly } k\}$$
$$\text{INDSET} = \{\langle G, k\rangle \mid \text{the largest independent set in } G \text{ has size at least } k\}$$

a.) 令 $L = \{\langle G, k\rangle \mid G \text{ has no independent set of size} \geq k+1\}$, 那么:

$$\text{EXACT INDSET} = \text{INDSET} \cap L$$

由于 $\neg L = \{\langle G, k\rangle \mid G \text{ has an independent set of size } k+1\} \in \mathbf{NP}$, 故 $L \in \mathbf{coNP}$, 又由于 INDSET $\in$ **NP**, 故 EXACT INDSET $\in$ **DP**.

b.) 对 $\forall A \in \mathbf{DP}, \exists A_1 \in \mathbf{NP}, A_2 \in \mathbf{coNP}$, $s.t.$ $A = A_1 \cap A_2$. 由于 INDSET $\in$ **NP**-complete, 类似课上的证明同理可以得到 $L \in$ **coNP**-complete 的. 那么存在多项式时间的规约 φ_1, φ_2 使得

$$w \in A \iff w \in A_1 \wedge w \in A_2 \iff \varphi_1(w) \in \text{3SAT} \wedge \varphi_2(w) \in \neg\text{3SAT}$$

而从 3SAT 可以构造合法规约 $\psi : \text{3SAT} \to \text{EXACT INDSET}$, 类似证明 3SAT $\leq_p$ INDSET 的方法构造 $G = (V, E)$. 令

V : 每个字句 $C_i \in \psi$ 对应含 7 个点的团, 分别对应 C_i 中 7 种可行的赋值方式.

E : 在团内部两两加边, 在互斥的赋值对应的点之间加边

可以使得 $\forall w \in$ 3SAT with k clauses $\iff \psi(w) = \langle G, k\rangle \in$ EXACT INDSET. 同理可以构造出合法归约 $\widehat{\psi} : \neg\text{3SAT} \to \text{EXACT INDSET}$.

故对 $\forall w \in A$, 存在多项式时间规约 f_1, f_2(把上面讨论的两次规约步骤复合一下即可), 使得

$$f_1(w) = \langle G_1, k_1\rangle \in \text{EXACT INDSET}, \quad f_2(w) = \langle G_2, k_2\rangle \in \text{EXACT INDSET}.$$

考虑如下的规约 f:

$$f(w) = \langle (n \cdot G_1) \cup G_2, nk_1 + k_2\rangle .$$

其中 n 是一个足够大的常数, $n \cdot G_1 \cup G_2$ 表示 n 个图 G_1 和 1 个 G_2 合并成一个新的图. 那么 f 是多项式时间的, 且有 $w \in A \iff f(w) \in$ EXACT INDSET. 故任意的 $A \in$ **DP** 都可以多项式时间规约到 EXACT INDSET, 即 EXACT INDSET 是 **DP**-complete 的. 证毕. ◁

Problem 5.(16 points). Show that the following language is **NL**-complete:

$$\{\langle G\rangle \mid G \text{ is strongly connected digraph}\}.$$

◀

Answer. 不妨设 $A = \{\langle G\rangle \mid G \text{ is strongly connected digraph}\}$. 首先证明 $A \in \mathbf{NL}$, 对任意的一个图 $G = (V, E)$, 设 $V = \{1, 2, \cdots, n\}$, 考虑如下的算法:

Input: Directed graph $G = (V, E)$

Output: Accept if G is strongly connected, reject otherwise.

1: **for** i from 1 to n **do**

2: Check whether $\langle G, i, i+1\rangle \in PATH$ (let $i+1 = 1$ if $i = n$).

3: **if** All the above checks are true **then**

4: **return Accept**.

5: **else**

6: **return Reject**.

注意到 $PATH \in \mathbf{NL}$, 那么第 2 行可以在 $O(\log n)$ 的空间内完成. 循环时重复使用同一空间则总的空间复杂度为 $O(\log n)$, 因此 $A \in \mathbf{NL}$.

下面证明 A 是 **NL**-hard 的. 对任意有向图 $G = (V, E)$, 如下构造出一个新的图 $G' = (V', E')$:

$$\text{令 } V' = V, E' = E, \text{ 选取两个点} s, t, \ \forall v \in V/\{s, t\}, \text{ 向 } E' \text{ 中添加边 } (t, v), (v, s).$$

那么我们可以证明如下引理:

Lemma 1. *G' 是强连通图当且仅当 $\langle G, s, t\rangle \in PATH$.*

证明. 1). $\Rightarrow$: 如果 G' 是一个强连通图, 那么存在路径 $s \rightsquigarrow t$, 注意到由 G 构造 G' 时只添加了 $t \to u \to s$, 那么 $\forall e \in E'/E, e \notin \{s \rightsquigarrow t\}$. 因此 $s \rightsquigarrow t$ 是 G 中的一条路径, 即 $\langle G, s, t\rangle \in PATH$.

2). $\Leftarrow$: 若 $\langle G, s, t\rangle \in PATH$, 那么 $\forall u, v \in G'$, 在 G' 中存在有向边 $u \to s, t \to v$ 与路径 $s \rightsquigarrow t$, 因此存在路径 $u \to s \rightsquigarrow t \to v$ 连接 u 和 v, 由于 u, v 是任意的, 故 G' 是强连通图. □

回到本题, 给定任意的 $\langle G, s, t\rangle$, 上述的构造过程 $f : G \to G'$ 只需要空间 $O(\log |\langle G, s, t\rangle|)$ 的工作带 (按顺序遍历点, 只在工作带上记录当前的点, 在输出纸带上添加相应的边), 由引理:

$$\langle G, s, t\rangle \in PATH \iff G' = f(\langle G, s, t\rangle) \text{ is strongly connected.}$$

故 f 是一个对数空间规约, 那么 $PATH \leq_l A$, 即 A 是 **NL**-hard 的.

综上所述, A 是 **NL**-complete. ◁

Problem 6.(20 points). Prove that in the certificate deinifion of **NL**, if we allow the verifier machine to move its head back and forth on the certificate, then the class being defined changes to **NP**. ◀

Answer. 记 $\mathbf{NL}^*$ 为允许验证机在证书上来回移动的复杂性类.

1). 首先证明 $\mathbf{NL}^* \subseteq \mathbf{NP}$. 对于 $\forall A \in \mathbf{NL}^*$, 存在空间复杂度为 $O(\log n)$ 的验证机 M, 以及多项式时间函数 $p(n)$, 使得对于

$$\forall x \in \Sigma^*, \ x \in A \iff \exists u \in \{0,1\}^{p(|x|)} \ s.t. \ M(x, u) = 1.$$

注意到任意一个空间复杂度为 $O(\log n)$ 的验证机可以在多项式时间内被模拟 (遍历空间的所有可能状态即可). 那么存在一个多项式时间的 DTM M', 使得对于

$$\forall x \in \Sigma^*,\ x \in A \iff \exists u \in \{0,1\}^{p(|x|)}\ s.t.\ M'(x,u) = 1.$$

因此 $A \in \mathbf{NP}$, 即 $\mathbf{NL}^* \subseteq \mathbf{NP}$.

2). 接着证明 $\mathbf{NP} \subseteq \mathbf{NL}^*$. $\forall A \in \mathbf{NP}$, 设存在一个 NDTM N 使得 $L(N) = A$. 那么:

$$\begin{aligned} x \in A \iff & \exists \text{ configurations } C_1, C_2, \cdots, C_T,\ s.t. \\ & (1)\ C_1 \text{ is the start configuration of } N \\ & (2)\ C_T \text{ is the accept configuration of } N \\ & (3)\ \forall i \in [1, T-1],\ C_i \vdash C_{i+1} \text{ under the transition function of } N. \end{aligned}$$

那么我们可以如下构造一个空间复杂度为 $O(\log n)$ 的验证机 M, 其中 $\langle\cdot\rangle$ 表示编码:

Input: string x and certificate $u = \langle C_1 C_2 \cdots C_T \rangle$

Output: 1 if N accepts x, 0 otherwise.

1: Check if C_1 is the start configuration of N.

2: Check if C_T is the accept configuration of N.

3: **for** $i = 1$ to $T - 1$ **do**

4: Check if $C_i \vdash C_{i+1}$ under the transition function of N.

首先证书 $u = \langle C_1 C_2 \cdots C_T \rangle$ 的长度显然是 $\mathrm{Poly}(n)$ 的. 第 1,2 步按顺序检查格局中的每个字符, 这样做的空间复杂度是 $O(\log n)$ 的. 第 3 行可以每次检查两个格局之间的转移是否合法 (一次检查一个字符), 这样做的空间复杂度也是 $O(\log n)$ 的. 因此 $A \in \mathbf{NL}^*$, 即 $\mathbf{NP} \subseteq \mathbf{NL}^*$.

综上所述, $\mathbf{NL}^* = \mathbf{NP}$.

除了课上给出的这种证明, 这里尝试一个其他证明 (麻烦助教一并看一下):

设 $\mathbf{NL}^*$ 定义中的验证机为 M, M 的空间复杂度为 $O(\log |x|)$, 那么 M 最多运行 $2^{O(\log |x|)} = O(|x|^c)$ 步 (其中 c 是一个常数). 我们允许 M 在证书上来回移动, 那么每次最多访问整个证书纸带 (由定义, 设证书长度为多项式 $p(|x|)$), 那么 M 每一步的时间复杂度为为 $O(p(|x|))$, 故整个验证过程的时间复杂度为 $O(|x|^c \cdot p(|x|))$. 因此, M 是多项式时间验证机, 且:

$$x \in A \iff \exists u \in \{0,1\}^{p(|x|)}\ s.t.\ M(x,u) = 1.$$

这即是 $\mathbf{NP}$ 的定义, 故 $\mathbf{NL}^* = \mathbf{NP}$.

这种证明不太正确。。。 ◁