Complejidad Computacional
Tarea 2.1

Karla Adriana Esquivel Guzmán
Andrea Itzel González Vargas
Luis Pablo Mayo Vega
Carlos Gerardo Acosta Hernández

Entrega: 03/04/17
Facultad de Ciencias UNAM

Ejercicios

1. Demuestra que el lenguaje $\Sigma_i SAT$ es completo para Σ_i^P bajo reducciones polinomiales temporales. Recuerda que SAT es $NP-completo$.

2. Demuestra que si $3SAT$ es temporalmente reductible polinomialmente a $\overline{3SAT}$ entonces $PH = NP$.

Sabemos que $3SAT$ es $NP-completo$, entonces $\overline{3SAT} \in coNP$.
Supongamos que $3SAT$ es reductible a $\overline{3SAT}$, esto implica que $NP = coNP$. Como $\sum_1^p = NP$ y $\prod_1^p = coNP$, entonces $\sum_1^p = \prod_1^p$. Como vimos en clase, para toda $i \geq 1$ si $\sum_i^p = \prod_i^p$ entonces $PH = \sum_i^p$, o sea que la jerarquía se colapsa al nivel i. Como $\sum_1^p = \prod_1^p$ entonces $PH = \sum_1^p = NP$.
Por lo tanto si $3SAT$ es reductible a $\overline{3SAT}$ (o sea $NP = coNP$), entonces $PH = NP$.

3. Demuestra que si $P^A = NP^A$ (para algún lenguaje A), entonces $PH^A \subseteq P^A$.

Tenemos que si $P^A = NP^A$ entonces P^A es cerrado bajo el complemento $=> P^A = coNP$, ahora de manera concisa tenemos que $P^A = \Sigma_1\ P^A = \sqcap_1\ P^A$.

Ahora Demostremos por Inducción que si $P^A = \Sigma_1\ P^A = \sqcap_1\ P^A => P^A = \Sigma_{i+1}\ P^A = \sqcap_{i+1}\ P^A$
- Consideremos una $\Sigma_{i+1}\ P^A$ M$\in$TM, que consiste en una serie de ramificaciones seguidas por una serie de ramificaciones universales.
- Consideremos ahora los subarboles de la trayectoria de un calculo cuyas raices son el primer paso universal a lo largo del camino para cada uno de estos subarboles, M esta realizando un calculo $\sqcap_i$.
Por Hipotesis $\sqcap_1\ P^A = P^A$ así podemos reemplazar cada uno de estos subarboles de calculo por

un método determinista en calculo de tiempo polinomial para formar una nueva Maquina S.
- Si dejamos que a(n) sea el máximo número de pasos dados por la Maquina alterna antes de que comiencen las ramas universales y P(n) sea el numero de pasos dados por cualquiera de las maquinas P^A deterministas, que hemos sustituido por los calculos para $\sqcap_i$, entonces el tiempo en que corre S esta limitado por a(n) + P^A(a(n)).
Observemos que P^A(a(n)) es una composición de funciones, porque los subprocedimientos en P^A estan calculando entradas que pueden ser mas grandes que n (Pero deberían ser mas pequeñas o igual a a(n) ya que solo se han ejecutado a(n) pasos en el tiempo que se usan los subprocedimientos.
-Como a y p son polinomiales también lo es su composicion, Por lo tanto S esta en NP^A, por Hipotesis $P^A = NP^A =>$ S esta en P^A
-De forma similar puede ser usado para reducir una maquina $\sqcap_{i+1}$ P^A a una coNP M∈MT y Como P^A=Σ_i P^A, poniendolo asi en P^A también completamos el colapso de la jerarquía.
Por lo tanto $PH^A \subseteq PH^A$

4. Demuestra que si $EXP \subseteq P/poli$, entonces $EXP = \Sigma_2^p$.

Dem.
Sea $L \in EXP$, entonces existe una máquina de Turing *time-oblivious* M que decide L en tiempo $2^{p(n)}$ *p.a.* polinomio p. Sea $s \in \{0,1\}^n$ una cadena de entrada para M. Sabemos por la definición de M que para cada $i \in [2^{p(n)}]$ denotamos con z_i la codificación de la i-ésima "instantánea" de la ejecución de M con la entrada s. Como $EXP \subseteq P/poli$, entonces existe un circuito C de tamaño $q(n)$ (p.a. polinomio q), tal que calcula z_i a partir de una i. La correctud de lo que calcula este circuito mencionado puede ser expresado como un predicado $coNP$. Así,

$$s \in L \iff \exists C \in \{0,1\}^{q(n)} \;\; \forall i, i1, ..., ik \in \{0,1\}^{p(n)} \;\; T(s, C(i), C(i_1), ..., C(i_k)) = 1 \qquad (1)$$

donde T es una TM que verifica esas condiciones en tiempo polinomial. Se puede entonces concluir que $L \in \Sigma_2^P$, que es lo que queremos. Para probar esto, consideremos $p(n) = 2^{n^k}$. Consideremos cada entrada (i,t) en la tabla de M, codifica una cadena $z_{i,t}$, *i.e.*, el contenido de la celda i, al momento t, siempre que la cabeza lectora esté en la entrada i al momento t, y de ser así, z almacena el estado interno de M. Ahora consideremos

$$L_M = \{\langle s, i, t, z\rangle \;\; | \;\; con\;\; la\;\; entrada\;\; s\;\; tenemos\;\; z_{i,t} = z \;\;\; para\;\; M\} \qquad (2)$$

Simulando M tendremos que $L_M \in EXP \subseteq P/poli$. Utilizando circuitos de tamaño polinomial para L_M, podemos construir un circuito de tamaño polinomial C de múltiple salida, tal que $C(\langle s, i, t\rangle) = z$. Como buscábamos en (1), decimos entonces que:

$$s \in L \iff \exists C \;\; \forall i, t \;\; t.q. \;\; C(\langle s, i, t\rangle) \;\; acepta \;\; si$$
$$C(\langle s, i-t, t-1\rangle), \;\; C(\langle s, i, t-1\rangle), \;\; C(\langle s, i+1, t-t1\rangle) \;\; y \;\; C(\langle s, 1, 2^{n^k}\rangle) \;\; aceptan.$$

Por lo tanto si $EXP \subseteq P/poli$, entonces $EXP = \Sigma_2^p$.