Ma50b Complejidad Computacional **16 de Abril de 2005**

Pauta Control 1

Prof. Cátedra: M. Kiwi *Prof. Auxiliar: J. Soto*

PROBLEMA 1:

(i).- Sean L y L' como en el enunciado. Primero observamos que $L' \overset{1}{\leftarrow} L = L' \cap (\{0,1\}^* \overset{1}{\leftarrow} L)$. Como intersección de lenguajes regulares es regular, bastará verificar que $\{0,1\}^* \overset{1}{\leftarrow} L$ es lenguaje regular. En efecto, supongamos que L es un lenguaje regular decidido por el autómata finito determinista M. Definimos el autómata finito no–determinista $\widetilde{M}$ con el mismo conjunto de estados de M de forma que para todo estado q se tiene que $\delta_{\widetilde{M}}(q,1) = \delta_M(q,1)$, $\delta_{\widetilde{M}}(q,0) = \{q, \delta_M(q,0)\}$, y $\delta_{\widetilde{M}}(q,\varepsilon) = \delta_M(q,0)$. La idea subyacente a la construcción de $\widetilde{M}$ es que éste autómata simule M cuando lee un caracter 1, pero permitirle a $\widetilde{M}$ ya sea ignorar caracteres 0 de la entrada o "inyectar" caracteres 0 a la entrada. De esta manera, $\widetilde{M}$ va "construyendo" una secuencia ω con la misma cantidad de 1's que su entrada $\widetilde{\omega}$ y simultáneamente verifica que M aceptaría ω.

Para completar la demostración basta con probar que $\widetilde{M}$ acepta $\{0,1\}^* \overset{1}{\leftarrow} L$.

Si $\widetilde{\omega} \in \{0,1\}^* \overset{1}{\leftarrow} L$, entonces existe un $\omega \in L$ tal que $unos(\widetilde{\omega}) = unos(\omega)$. Luego, es posible obtener ω a partir de $\widetilde{\omega}$ agregando o eliminando 0's en $\widetilde{\omega}$. Suponiendo que en la entrada $\widetilde{\omega}$ la máquina $\widetilde{M}$,

- al leer un caracter 0 realiza una transición ϵ si dicho 0 debió ser agregado,
- permanece en el mismo estado en que está si dicho 0 debió ser eliminado,

se obtiene una secuencia de transiciones que dejará a $\widetilde{M}$ en el mismo estado que ω deja a M, i.e., $\widetilde{M}$ aceptará $\widetilde{\omega}$.

Por otro lado, si $\widetilde{M}$ acepta $\widetilde{\omega}$, entonces existe una camino en el diagrama de transiciones de $\widetilde{M}$ desde el estado de partida a un estado de aceptación. Sin pérdida de generalidad podemos asumir que el camino es simple, i.e., visita un nodo del diagrama de transiciones a lo más una vez. Las transiciones de dicho camino están asociadas a un caracter en $\{0,1\}$ o a una transición ε. Sea ω'_i igual a 0, 1 o ε dependiendo de que tipo de transición se trate. Sea $\omega_i = 1$ si $\omega'_i = 1$ y $\omega_i = 0$ en caso contrario. Notar que $unos(\widetilde{\omega}) = unos(\omega)$. Es fácil ver que M acepta ω, i.e., $\widetilde{\omega} \in \{0,1\}^* \overset{1}{\leftarrow} L$.

(ii).- Esta parte es una aplicación directa de la construcción usada para establecer que todo lenguaje libre de contexto es reconocido por un autómata apilador. Dicha construcción da lugar a el autómata apilador que se muestra en la Figura 1.

PROBLEMA 2:

(i.1).- Basta observar que $L_{a,b} = 0^a(0^b)^*$ y que toda expresión regular genera un lenguaje regular.

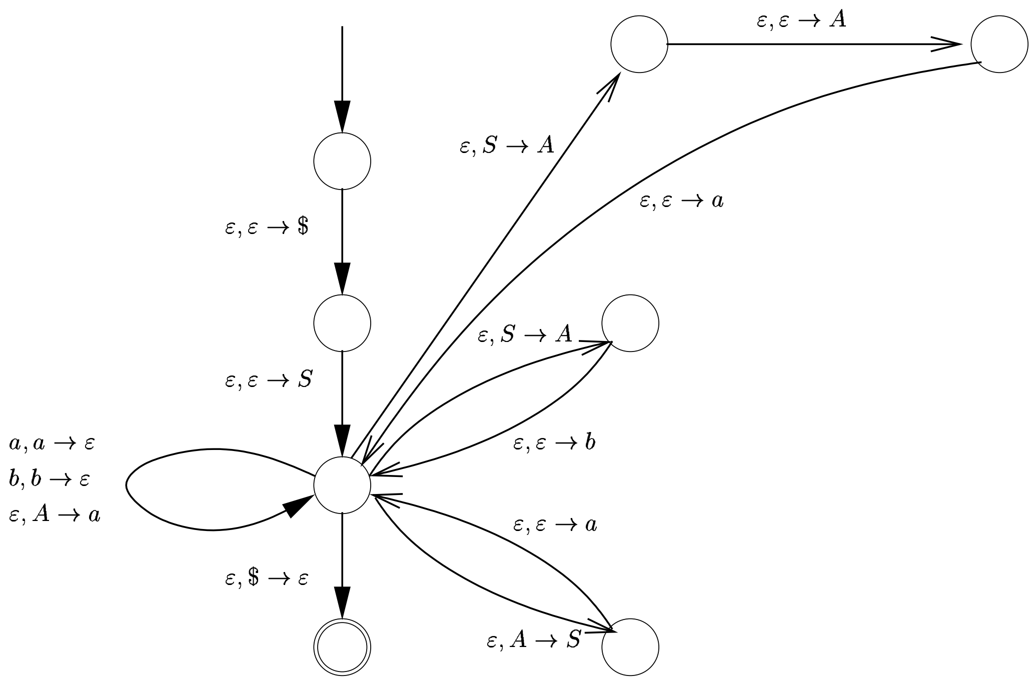

Figura 1: Autómata apilardor que reconoce L_G.

(i.2).- Por el Lema del Bombeo para lenguajes libre de contexto se tiene que si $m > p$ y $0^m \in L$ entonces existen $u, v, x, y, z \in 0^*$ tales que

$$\begin{aligned} 0^m &= uvxyz, \qquad \text{y} \qquad uv^ixy^iz \in L, \qquad \forall i \in \mathbb{N}, && (1)\\ |vxy| &\leq p\,, && (2)\\ |vy| &\neq 0\,. && (3) \end{aligned}$$

Pero como $u, v, x, y, z \in 0^*$ sigue que $0^m = uxz(vy)$, y $uv^ixy^iz = uxz(vy)^i$, para todo $i \in \mathbb{N}$. Sean n, q tales que $0^n = uxz$ y $0^q = vy$. De (1) sigue que $m = n + q$ y $0^m \in L_{n,q} \subseteq L$. De (2) sigue que $q = |vy| \leq |vxy| \leq p$, y de (3) sigue que $q \geq 1$. Es decir se tiene la conclusión deseada.

(i.3).- Sea $F = \{\omega \in 0^* \,:\, \omega \in L, |\omega| \leq p\}$. Claramente F es un conjunto finito. Para $m > p$ donde $0^m \in L$, sean $n = n(m)$ y $q = q(m)$ dos enteros (cualesquiera) que satisfacen:

$$m = n + q, \qquad 1 \leq q \leq p, \qquad \text{y} \qquad 0^m \in L_{n,q} \subseteq L\,.$$

La parte anterior garantiza la existencia de $n = n(m)$ y $q = q(m)$.

Veamos que

$$L \;=\; F \cup \bigcup_{m>p,0^m \in L} L_{n(m),q(m)}\,. \tag{4}$$

En efecto, si $0^m \in L$ y $m \leq p$ entonces $0^m \in F$. Si $0^m \in L$ y $m > p$ entonces $0^m \in L_{n(m),q(m)}$. Por otro lado, $F \subseteq L$ y de la parte anterior se tiene que $L_{n(m),q(m)} \subseteq L$ si $0^m \in L$ y $m > p$.

$M =$ En la entrada $\langle G \rangle$, donde G es GLC
    $d \leftarrow \max\{|\alpha| \,:\, A \rightarrow \alpha \in \mathcal{R}_G\}$
    $p \leftarrow |V_G|^{d+2}$
    $m \leftarrow p! + p$
    Obtiene $\langle A \rangle$ donde A es un automata apilador tal que $L_A = L_G$.
    Para $k \in \{0, \ldots, m\}$
        Simula A en 0^k
        Rechaza y para si A rechaza
    Acepta

Figura 2: Máquina de Turing que decide si $\langle G \rangle$ es tal que G es GLC y $0^* \subseteq L_G$.

A continuación dividiremos los $L_{n(m),q(m)}$ que aparecen en (4) de acuerdo al valor que toma $q(m)$. Para todo $i \in \{1, \ldots, p\}$ sea $M_i = \{m \in \mathbb{N} \,:\, 0^m \in L, m > p, q(m) = i\}$ un conjunto de índices y sea

$$Q_i \;=\; \bigcup_{m \in M_i} L_{n(m),i}\,.$$

A partir de (4) se deduce que

$$L \;=\; F \cup \bigcup_{i=1}^{p} Q_i\,.$$

La conclusión deseada se deriva del echo que Q_i es unión finita de conjuntos lineales. En efecto, como $n(m) \equiv n(m') \pmod{i}$ y $n(m) \leq n(m')$ implican que $L_{n(m),i} \subseteq L_{n(m'),i}$, sigue que Q_i es unión de a lo más $i \leq p$ conjuntos lineales distintos.

(i.4).- Sin pérdida de generalidad podemos asumir que L es un lenguaje sobre el alfabeto $\Sigma = \{0\}$. Sabemos que todo lenguaje de contexto libre sobre el alfabeto Σ es unión de un conjunto finito y una cantidad finita de conjuntos lineales. Pero todo conjunto finito es regular y además los conjuntos lineales son regulares. Sigue que L es unión finita de conjuntos regulares. Por propiedades de clausura de conjuntos regulares se concluye que L es regular.

(ii).- Sea m un entero suficientemente grande cuyo valor fijaremos más adelante. Probarémos que $0^* \subseteq L_G$ si y sólo si $0^k \in L_G$ para k entre 0 y m. Sea p una cota superior en la constante de bombeo de L_G. Por contradicción, sea $r > m + p$ el entero más pequeño tal que $0^r \notin L_G$. Suponiendo que $r > m$, entonces $0^{r-m} \in L_G$. Como $r - m > p$, el Lema del Bombeo para Lenguajes de Libre Contexto implica que existe n y q tal que $r - m = n + q$, $q \leq p$, y $0^{n+tq} \in L_G$ cualquiera sea $t \in \mathbb{N}$. Luego, si para algún t se tiene que $r = n + tq$, entonces tendríamos una contradicción. Tomando $m = p!$ y dado que $q \leq p$, sigue que existe t tal que $(t-1)q = p!$ y por lo tanto $r = n + tq$.

De todo lo anterior, sigue que para decidir si $0^* \subseteq L_G$, basta determinar una cota p en la constante de bombeo de L_G (lo que se puede hacer dada la descripción de L_G) y verificar si las $p + p!$ primeras palabras de 0^* están en L_G. Sigue que la mT de la Figura 2 decide el lenguaje L especificado en el enunciado.

PROBLEMA 3:

(i).- La observación crucial aquí es que si una mT izquierda deficiente I no desplaza su cabeza lectora una celda a la derecha en $|Q_I| \cdot |\Sigma_I \cup \{\not b\}|$ pasos, entonces nunca lo hará. Luego, en a lo más $|\omega| \cdot |Q_I| \cdot |\Sigma_I \cup \{\not b\}|$ pasos la máquina o para, o coloca su cabeza lectora sobre la celda en blanco inmediatamente a la derecha de la entrada. En los siguientes $|Q_I| \cdot |\Sigma_I|$ pasos ocurrirá alguna de las siguientes situaciones:

- La máquina para.
- La cabeza lectora permanece sobre la misma celda y escribe el blanco.
- La cabeza lectora se mueve una celda a la derecha.

En los dos últimos casos, la máquina entra en un ciclo infinito, nunca para, y por lo tanto nunca acepta.

En resumen, una mT izquierda deficiente se detiene en $(|\omega|+1) \cdot |Q_I| \cdot |\Sigma_I \cup \{\not b\}|$ pasos o entra en un ciclo infinito y nunca acepta. En cualquier caso, a una mT M que recibe como entrada $\langle I, \omega \rangle$ donde I es una mT izquierda deficiente, le bastará simular I por $(|\omega|+1) \cdot |Q_I| \cdot |\Sigma_I \cup \{\not b\}|$ pasos para determinar si I acepta ω. Sigue que A_{mTID} es decidible.

(ii).- Sean M_A y M_B mT que reconocen $\overline{A}$ y $\overline{B}$ respectivamente. Definimos la mT M tal que en la entrada ω simula alternadamente una transición de M_A y uno de M_B. La máquina M para y acepta apenas detecte que la simulación de M_A deja a esta última máquina en un estado de aceptación. Por otro lado, M para y rechaza apenas detecte que la simulación de M_B deja a esta última máquina en un estado de aceptación.

Como $\overline{A} \cup \overline{B} = \Sigma^*$ sigue que $M_A(\omega) = \mathit{acepta}$ o $M_B(\omega) = \mathit{acepta}$ cualquiera sea $\omega \in \Sigma^*$. Luego, L_M es decidible. Notar además que si $\omega \in A$, entonces $M_A(\omega) \neq \mathit{acepta}$ y $M_B(\omega) = \mathit{acepta}$. Esto implica que M rechaza ω, i.e., $\omega \notin L_M$. Análogamente, se tiene que si $\omega \in B$, entonces M acepta ω, i.e., $\omega \in L_M$. Resumiendo, $A \subseteq \overline{L_M} \subseteq \overline{B}$. Tomando $C = \overline{L_M}$ se obtiene la conclusión deseada.

(iii).- Recordemos que $\overline{EQ_{mT}}$ es igual al conjunto de secuencias $\langle M_0, M_1 \rangle$ donde M_0 y M_1 son mT sobre el mismo alfabeto tales que $L_{M_0} \neq L_{M_1}$. Luego,

$$\langle M_0, M_1 \rangle \in \overline{EQ_{mT}} \iff \exists \omega \in \Sigma^*_{M_0}, (\omega \in L_{M_0} \wedge \omega \notin L_{M_1}) \vee (\omega \notin L_{M_0} \wedge \omega \in L_{M_1}) \,.$$

Por otro lado, como L_{M_0} y L_{M_1} son recursivamente enumerables, sabemos que existen lenguajes decidibles D_0 y D_1 tales que para $b \in \{0,1\}$ se tiene que

$$\omega \in L_b \iff \exists \Pi_b, \omega \# \Pi_b \in D_b \,.$$

Sigue que $\langle M_0, M_1 \rangle \in \overline{EQ_{mT}}$ si y sólo si

$$\exists \omega \in \Sigma^*_{M_0}, \exists \Pi, \forall \Pi', (\omega\#\Pi \in D_0 \wedge \omega\#\Pi' \in \overline{D_1}) \vee (\omega\#\Pi \in \overline{D_0} \wedge \omega\#\Pi' \in D_1) \,.$$

Definiendo

$$D = \{\omega\#\Pi\#\Pi' \,:\, (\omega\#\Pi \in D_0 \wedge \omega\#\Pi' \in \overline{D_1}) \vee (\omega\#\Pi \in \overline{D_0} \wedge \omega\#\Pi' \in D_1)\} \,,$$

y observando que D es decidible, se obtiene la conclusión deseada.