MA50B Complejidad Computacional		**13 de Abril, 2002**
	Pauta Control 1	
Profesor: M. Kiwi		**Auxiliar: E. Moreno**

Problema 1:

(i).- Sea M un autómata finito que reconoce L. Para $q \in Q_M$ definimos el lenguaje

$$L_q = \{x \in L/2 \,:\, M \text{ en la entrada } x \text{ termina en el estado } q\}\,.$$

Claramente, $L/2 = \bigcup_{q \in Q_M} L_q$. Luego, como unión de lenguajes regulares es regular, bastará probar que L_q es regular cualquiera sea $q \in Q_M$. Para ello, definimos el autómata M_q cuyos estados son $Q_M \times Q_M$, su estado de partida es (S_M, q), sus estados de aceptación son (q, q_{ac}) para todo $q_{ac} \in F_M$ y su función de transición es tal que

$$(\widetilde{q}_0', \widetilde{q}_1') \in \delta_{M_q}((\widetilde{q}_0, \widetilde{q}_1), \alpha) \Longleftrightarrow \widetilde{q}_0' = \delta_M(\widetilde{q}_0, \alpha) \wedge \exists \beta \in \Sigma_M, \widetilde{q}_1' = \delta_M(\widetilde{q}_1, \beta)\,.$$

Queda argumentar que el lenguaje reconocido por M_q es efectivamente L_q. La demostración consiste en probar por inducción en n que si $x \in \Sigma_M^*$, $|x| = n$, entonces existe $y \in \Sigma_M^*$, $|y| = n$ tal que M en la entrada x termina en el estado q_0 y M partiendo del estado q y leyendo y alcanza el estado q_1 si y sólo si M_q en la entrada x puede alcanzar el estado (q_0, q_1).

(ii).- Observar que $\omega = \omega_1 \ldots \omega_l \in L$ si y sólo si $l \geq n$ y $\omega_{l+1-n} = 1$.

Sea M un autómata finito determinista que reconoce L. Supongamos que $|Q_M| < 2^n$. Por el principio del palomar sigue que deben haber dos secuencias $\alpha, \beta \in \{0,1\}^n$ distintas y un estado $q \in Q_M$ tales que M al leer las entradas α o β queda en el estado q. Sea $i \in \{1, \ldots, n\}$ el mayor índice tal que $\alpha_i \neq \beta_i$. Sin pérdida de generalidad asumimos que $\alpha_i = 1$ y $\beta_i = 0$. Sea γ una secuencia cualquiera de i bits. Sigue que $\alpha' = \alpha\gamma \in L$ pero $\beta' = \beta\gamma \notin L$, sin embargo M en la entrada α' y β' termina en el mismo estado (pues una vez leídos los n primeros caracteres de α' o β' – es decir α o β – el autómata M alcanza el estado q y α' y β' coinciden a partir del (n+1)-ésimo carácter).

(iii.1).- Supongamos primero que L es un lenguaje de libre contexto. Sea $\omega \in L$ la palabra de menor largo en L (en caso de no haber una sola tomamos cualquiera con esta propiedad). Si $|\omega| \leq p$ estaríamos listos. Supongamos entonces que $|\omega| > p$. Por el Lema del Bombeo para lenguajes de libre contexto, existen $u, v, x, y, z \in \Sigma_L^*$ tales que $\omega = uvxyz$, $|vy| > 0$ y $uv^n xy^n z \in L$ cualquiera sea $n \in \mathbb{N}$. En particular, tomando $n = 0$, obtenemos que $uxz \in L$ y que $|uxz| < |vy| + |uxz| = |\omega|$, contradiciendo así la minimalidad de $\omega \in L$.

El converso es obvio.

(iii.2).- Supongamos primero que L es un lenguaje de libre contexto que contiene una infinidad de palabras. Sea $\omega \in L$ una palabra de largo mínimo entre las palabras de largo al menos p (observar que ω existe pues de lo contrario L sería finito). Si $|\omega| \leq 2p$ estamos listos. De lo contrario, como $|\omega| > 2p \geq p$, por el Lema del Bombeo para lenguajes de libre contexto, existen $u, v, x, y, z \in \Sigma_L^*$ tales que $\omega = uvxyz$, $|vy| > 0$, $|vxy| \leq p$ y $uv^n xy^n z \in L$ cualquiera sea $n \in \mathbb{N}$. Luego, $uxz \in L$, $|\omega| = |uxz| + |vy| > |uxz| = |\omega| - |vy| \geq |\omega| - |vxy| > 2p - p = p$. Es decir, $uxz \in L$ y $2p \geq |uxz| > p$ lo que contradiciendo la minimalidad de $\omega \in L$, $|\omega| > p$.

El converso es una consecuencia directa de la aplicación del Lema del Bombeo para lenguajes de libre contexto, pues a partir de $\omega \in L$, $|\omega| > p$ se deduce la existencia en L de una infinidad de palabras.

Problema 2:

(i).- Las palabras que están en L tienen la forma 01, 010011, 01001100001111, etc. Notar que cada una de estas palabras se obtiene de la anterior duplicando cada caracter (ya sea 0 o 1) y anteponiendo 01 a la palabra así obtenida. Diseñaremos una gramática que construye $\omega = \omega^{(1)}\omega^{(2)}\cdots\omega^{(n)}$, $|\omega^{(i)}| = 0^{2^{i-1}}1^{2^{i-1}}$, $i = 1, \ldots, n$ en n etapas. Durante la i-ésima etapa generaremos la palabra $\omega^{(1)}\omega^{(2)}\cdots\omega^{(i+1)}$ a partir de la palabra $\omega^{(1)}\omega^{(2)}\cdots\omega^{(i)}$.

Específicamente, sea $G = (V, \Sigma, R, S)$ donde $\Sigma = \{0, 1\}$ y el conjunto R contiene las siguientes reglas de producción (las variables de G se indican con letras mayúsculas):

$$\left.\begin{array}{rcl} S & \longrightarrow & 01 \end{array}\right\} \quad \text{Genera las palabras en } L \text{ de largo 2.}$$

$$\left.\begin{array}{rcl} S & \longrightarrow & R\#0\$ \end{array}\right\} \quad \begin{array}{l}\text{Inicia generación de palabras en } L \text{ de largo al}\\ \text{menos 2 (\$ se interpreta como el último 1).}\end{array}$$

$$\left.\begin{array}{rcl} R0 & \longrightarrow & 00R \\ R1 & \longrightarrow & 11R \\ R\# & \longrightarrow & \#01R \\ R\$ & \longrightarrow & 1\$L \end{array}\right\} \quad \text{Movimiento a la derecha ``duplicando'' 0's y 1's.}$$

$$\left.\begin{array}{rcl} 0L & \longrightarrow & L0 \\ 1L & \longrightarrow & L1 \\ \$L & \longrightarrow & L\$ \\ \#L & \longrightarrow & R\# \end{array}\right\} \quad \text{Movimiento a la izquierda.}$$

$$\left.\begin{array}{rcl} R\# & \longrightarrow & F\# \end{array}\right\} \quad \text{Comienza fase final de la generación.}$$

$$\left.\begin{array}{rcl} F\# & \longrightarrow & 01F \\ F0 & \longrightarrow & 00F \\ F1 & \longrightarrow & 11F \\ F\$ & \longrightarrow & 11 \end{array}\right\} \quad \text{Fase final de la generación.}$$

Observación 1 *La necesidad de reglas de producción para generar palabras de largo* 2 *y* *las aparentemente demasiado involucradas reglas de producción necesarias para terminar de generar las palabras, se explican por la necesidad de que para que $x \to y$ sea una regla de producción se debe tener que $|x| \leq |y|$.*

(ii).- Sea L un lenguaje reconocido por un autómata de doble sentido D. Observar que cualquiera sea la entrada, si en dos instantes de tiempo distintos el autómata D se encuentra en el mismo estado y con su cabeza lectora sobre la misma celda, entonces el autómata se encuentra en un loop del cual no saldrá — por lo tanto, no aceptará. Fijemos entonces la entrada ω del autómata D y observamos la secuencia $(q_1, \ldots, q_l)$ de estados de D al cruzar una misma frontera entre dos celdas de su cinta (asumimos como convención que D primero cambia su estado y luego mueve su cabeza lectora). Los estados q_i con i impar son aquellos en que D esta leyendo la celda a la derecha de la frontera. Por el contrario, los estados q_i con i par son aquellos en que D tiene su cabeza lectora sobre la celda a la izquierda de la mencionada frontera. Si un mismo estado se repite en una posición par o en una posición impar, el autómata D no aceptará. Luego, basta considerar secuencias de a lo más $2|Q_D|$ estados. En una secuencia más larga de estados, por el principio del palomar, dos estados se repetirían en posiciones de idéntica paridad y ω no sería aceptada por D.

La discusión del párrafo anterior motiva el considerar un autómata finito determinista D' cuyos estados son posibles secuencias de estados de D de largo a lo más $2|Q_D|$. El estado de partida de D' es aquel asociado a la secuencia de largo 1 dada por S_D (el estado de partida de D). Sin pérdida de generalidad asumimos que si D acepta lo hace cuando su cabeza lectora esta sobre el símbolo #. Un estado de D' será de aceptación si el último estado de la secuencia asociada es un estado de aceptación de D. Finalmente, la función de transición de D' será tal que del estado $(q_1, \ldots, q_l)$ podrá haber una transición al estado $(q'_1, \ldots, q'_{l'})$ cuando se lee el símbolo $\alpha \in \Sigma_D$ si $(q_1, \ldots, q_l)$ y $(q'_1, \ldots, q'_{l'})$ pueden ser secuencias de cruces de las fronteras izquierda y derecha respectivamente de una celda que contenga el símbolo α.

Sólo queda observar que en la entrada ω, el autómata D' acepta si y sólo si existen $|\omega|+1$ secuencias de cruce, que juntas representan la evolución del autómata de doble sentido D en la entrada ω.

Problema 3:

(i).- Sea $L = \{\omega \in \Sigma^* : \exists \pi, \langle \omega, \pi \rangle \in D\}$ donde D es decidible. Sea M una mT determinista M que decide D. Sin pérdida de generalidad podemos asumir que M tiene una sola cinta. Sea S la mT no–determinista tal que

S = "En la entrada ω,

(i) Mueve su cabeza a la derecha hasta la primera celda en blanco

(ii) Escribe # y mueve la cabeza a la derecha

(iii) Usando no–determinismo

– adivina un bit y lo escribe en su cinta,

– mueve su cabeza a la derecha,

– adivina si debe continuar o pasar a (v).

(iv) Recodifica el contenido de su cinta, i.e., $\omega\#\pi$, como $\langle\omega,\pi\rangle$.

(v) Mueve su cabeza a la izquierda hasta el comienzo de su cinta.

(vi) Simula M tomando como entrada el contenido de la cinta.

(vii) Acepta si M acepta, y rechaza si M rechaza.

Claramente, S acepta ω si y sólo si $\exists\pi$ tal que $\langle\omega,\pi\rangle$ es aceptada por M. Esto último equivale a decir que $\exists\pi, \langle\omega,\pi\rangle \in D$, i.e., $\omega \in L$. En resumen S reconoce L.

Supongamos ahora que L es reconocible. Sigue que existe una mT M tal que $L = L(M)$. Sea D' el lenguaje formado por las palabras de la forma

$$\omega\#C_0\#C_1\#\ldots\#C_m\#\,,$$

donde cada C_i corresponde a una posible descripción instantánea de M. En particular C_0 y C_m son descripciones instantáneas de M al partir en la entrada ω y al aceptar respectivamente. Más aun, C_{i+1} es una configuración que M puede alcanzar en una transición.

Observar que $L = \{\omega : \exists\pi, \omega\pi \in D'\}$. Recodificando adecuadamente las palabras $\omega\pi \in D'$ como $\langle\omega,\pi\rangle \in \Sigma^*$ se obtiene la conclusión deseada una vez que se comprueba que D es efectivamente decidible (omitido).

(ii).- Sea M la mT no–determinista con k cintas y a tiempo $f(n)$. Sea N otra mT no–determinista de dos cintas. Supondremos que las dos cintas de N tienen k tracks, i.e., que cada celda esta subdividida en k sub–celdas. Cada sub–celda puede almacenar un símbolo de la forma $\sigma \in \Sigma_M$ (sin pérdida de generalidad asumimos que N recibe la entrada $\omega = \omega_1 \ldots \omega_n$ de manera que en la primera sub–celda de la celda i esta ω_i y el resto de las sub–celdas de la celda i están en blanco).

Lo primero que hace N es, para $t = 1, \ldots, f(|\omega|)$ adivinar cuales símbolos estarían leyendo cada una de las k cabezas lectoras de la mT M en el instante de tiempo t en una rama de aceptación de M en la entrada ω. La máquina N almacena en la i–ésima sub–celda de la celda t de la segunda cinta el símbolo supuestamente bajo la i–ésima cabeza lectora de M en el instante t. Esto puede hacerse en un sólo barrido de la segunda cabeza lectora de N sobre la correspondiente cinta. Una vez realizada esta etapa, N regresa su segunda cabeza lectora a la primera celda de la correspondiente cinta. Todo lo anterior toma tiempo $2f(|\omega|)$.

La segunda parte de la ejecución de N constará de k sub–etapas. Cada sub–etapa tomará $2f(|\omega|)$ instantes de tiempo. Los últimos $f(|\omega|)$ instantes sólo se usarán para regresar ambas cabezas lectoras a la primera celda de sus correspondientes cintas. En el instante $t = 1, \ldots, f(|\omega|)$ de la sub-etapa i, la segunda cabeza lectora se encontrará sobre la celda t de su cinta, la máquina N realizará los siguientes pasos:

1. Verificará que el contenido de la sub–celda i de la celda bajo la segunda cabeza lectora corresponde al contenido de la sub–celda i de la celda bajo la primera cabeza lectora. En caso contrario rechaza.

2. Asumirá que los símbolos leídos por cada una de las k cabezas lectoras de M en el instante t son aquellos almacenados en las k sub–celdas de la celda bajo la segunda cabeza lectora. De acuerdo a estos valores determinará en que dirección D hubiese desplazado M su i–ésima cabeza lectora, el símbolo α que M hubiese escrito en su i–ésima cinta y el estado q en el que M hubiese quedado. La máquina N escribirá α en la i–ésima sub-celda de la celda bajo su primera cabeza, moverá dicha cabeza en la dirección D, desplazará su segunda cabeza una celda a la derecha y simulará una transición de M al estado q.

3. Si q es un estado de aceptación (respectivamente de rechazo) de M, entonces N acepta (respectivamente rechaza).

Observar que cada una de las k sub–etapas de la segunda parte de la ejecución de N efectivamente toman $2f(|\omega|)$ instantes de tiempo.

Queda verificar que N decide el mismo lenguaje que M y que es a tiempo $O(f(n))$. Lo último es evidente del hecho que a N le toma $2(k+1)f(n)$ pasos simular $f(n)$ pasos de M y dado que k es una constante fija independiente del largo de la entrada.

La igualdad de los lenguajes decididos por N y M se obtiene mostrando (por inducción) que al instante $t \in \{1, \ldots, f(|\omega|)\}$ de la i–ésima sub–etapa (de la segunda parte) de la ejecución de N, si los contenidos de la sub–celda t' del track $i' > i$ y de las sub-celdas $t' > t$ del track $i' = i$ de la segunda cinta son los símbolos leídos por la cabeza i' de M en el instante t', entonces los contenidos de la sub–celda t' del track $i' < i$ y de las sub-celdas $t' \leq t$ del track $i' = i$ de la segunda cinta son los símbolos leídos por la cabeza i' de M en el instante t' (omitido).