Autómatas y Lenguajes Formales

Felipe Bracho y Carlos Velarde

Examen 3

1[3 en el libro].- ¿Cómo puede diferir la ejecución de $\rightarrow RL$ de la de $\rightarrow LR$?

$\rightarrow LR$ Puede terminar de manera anormal, si en la configuración inicial la cabeza se encuentra al principio de la cinta; mientras que $\rightarrow RL$ siempre deja la cabeza en el mismo lugar.

2a[15a].- ¿Es la unión finita de lenguajes estructurados por frase, también estructurado por frase?. Justifique su respuesta.

La unión finita de lenguajes estructurados por frase es también estructurado por frase.
DEM: Sean $G_1, G_2, ...G_n$ las gramáticas que generan a $L_1, L_2, ...L_n$. Podemos suponer, sin pérdida de generalidad, que sus símbolos no terminales son ajenos entre ellas, y que S_i, es el símbolo inicial de G_i para $1 \leq i \leq n$. Formamos una nueva gramática G, con símbolo inicial S y a la unión de las transiciones de $G_1, G_2, ...$ y G_n le añadimos $S \rightarrow S_i$ para cada $1 \leq i \leq n$. Es fácil ver que G genera exactamente $L_1 \cup L_2 ... \cup L_n$.

2b[15b].- ¿Es la unión de una colección de lenguajes estructurados por frase, también estructurado por frase?. Justifique su respuesta.

No necesariamente.

DEM: Para cada palabra w en Σ^*, el lenguaje $\{w\}$, es estructurado por frase (y generado por la grámatica con una producción: $S \to w$). De ahí que cualquier lenguaje con alfabeto en Σ, y en particular cualquier L *no* estructurado por frases (existe por el teorema 3.5), es la unión de lenguajes ($\{w\}$, con $w \in L$) estructurados por frase.

$$\bigcup_{w \in L} \{w\} = L.$$

3[21].- Demuestre que la colección de los lenguajes no estructurados por frase tiene una cardinalidad mayor que la de los estructurados por frase.

DEM: Como Σ es finito, Σ^* es numerable. De donde obtenemos por el teorema 0.1,

$$\mid \Sigma^* \mid < \mid \mathcal{P}(\Sigma^*) \mid \quad (1)$$

que $\mathcal{P}(\Sigma^*)$ es no numerable.

La construcción del teorema 3.5 demuestra que las máquinas de Turing sobre Σ forman un conjunto numerable, por lo que podemos inferir (teorema 3.4) que los lenguajes estructurados por frase, que denotaremos por $\mathcal{E}$, son también numerables. Ahora bien, $\mathcal{E} \cup \mathcal{E}^c = \mathcal{P}(\Sigma^*)$ implica que si los lenguajes no estructurados por frase ($\mathcal{E}^c$) fuesen numerables, $\mathcal{P}(\Sigma^*)$ también lo sería, pues la unión de dos numerables es también numerable (ejercicio 27 capítulo 0). Lo que contradice a (1) arriba. De lo anterior $\mathcal{E}^c$ es no numerable y $\mathcal{E} < \mathcal{E}^c$.

4[25].- ¿Los subconjuntos de un lenguaje estructurado por frase, son siempre estructurados por frase?. Justifique su respuesta.

La repuesta es: no. La razón es fácil de ver si consideramos que todos los lenguajes sobre Σ son subconjuntos de Σ^* y Σ^* mismo es estructurado por frases (es regular y reconocido por un autómata de un solo estado). El teorema 3.5 nos dice que dado Σ (que siempre es no vacío), existe L sobre Σ, lo que implica que $L \subset \Sigma^*$, tal que L es no estructurado por frase.

5[26].- Demuestre que la estrella de Kleene L^* de un lenguaje L aceptable por Máquina de Turing, es también aceptable por Máquina de Turing.

Solución 1.- Por los teoremas 3.3 y 3.4, sabemos que los lenguajes aceptados por Máquinas de Turing son precisamente los estructurados por frase, por lo que basta demostrar que dada una gramática G que genera L, podemos construir una G' que genera L^*.

Sea G tal que $\mathcal{L}(G) = L$ y $\{S', S_1, X, \#, *\}$ son símbolos ajenos a G, formamos G' con símbolo inicial S' añadiendo a las producciones de G, las siguientes:

$$
\begin{array}{ll}
(1) & S' \to \lambda \\
(2) & S' \to \#S_1* \\
(3) & S_1* \to S_1XS_1* \\
(4) & S_1 \to S \\
(5) & \{x* \to *x \mid x \in \Sigma\} \\
(6) & X* \to * \\
(7) & \#* \to \lambda
\end{array}
$$

donde S es el símbolo inicial de G.

Afirmamos que $\mathcal{L}(G') = L^*$. Es fácil ver que si $w \in L^*$ entonces, $w \in \mathcal{L}(G')$, pues w es de la forma $w_1...w_n$ con $0 \leq n$ y usando las primeras cuatro reglas podemos generar una cadena de la forma $\#SXSX...XS*$ con n ocurrencias de S. Usamos entonces las reglas de G que nos permiten generar $\#w_1Xw_2X...Xw_n*$ para que podamos correr el $*$ a la derecha con las reglas (5) y (6). Usamos finalmente la regla (7) y obtenemos $w_1...w_n = w$.

Para ver que $w \in \mathcal{L}(G')$ implica $w \in L^*$, nótese que cada regla de reescritura de G, sólo puede aplicarse a segmentos entre los delimitadores $\#$, X y $*$, pues a la derecha de $*$, sólo puede haber símbolos terminales y, por hipótesis, los símbolos $\{S', S_1, X, \#, *\}$ son ajenos a G, por lo que no pueden aparecer en ninguna regla de derivación de G.

En relación al último ejercicio, conviene hacer notar que si bien el modificar la gramática G añadiendo $S \to \lambda$ y $S \to SS$, funciona para el caso de gramáticas libres de contexto; este no es el caso para las gramáticas de estructura de frase (o sensibles al contexto). Pues las reglas podrían aplicarse a palabras que devienen de dos ocurrencias diferentes de S, como en el caso siguiente:

Sea G la gramática sobre $\{a, b\}$ con las siguientes reglas:

$$\begin{array}{ll} (1) & S \to A \\ (2) & S \to B \\ (3) & A \to a \\ (4) & B \to b \\ (5) & AB \to c \end{array}$$

En este caso $\mathcal{L}(G) = \{a, b\}$. Si formamos G' simplemente agregando $S \to \lambda$ y $S \to SS$, obtenemos $\{a, b, c\}^*$, en lugar de $\{a, b\}^*$

La solución al problema usando máquinas de Turing, también tiene aspectos que hay que considerar con cuidado; pues la máquina M que reconoce al lenguaje original L, no necesariamente va pasando al estado final cada vez que recorre un segmento en w que esté en L. M puede requerir de llegar al final de la palabra para hacer operaciones y eventualmente aceptar. Tampoco tiene necesariamente que dejar la cinta sin modificaciones, como lo hace un autómata que sólo lee de izquierda a derecha. Por estas razones, un simple bucle (o *loop*) no necesariamente funciona.

Una solución usando máquinas de Turing, es construir a partir de la máquina dada M, que acepta L, (y que podemos suponer que termina borrando su cinta y con la cabeza al inicio), una máquina M' no determinista, de dos cintas, de la manera siguiente.

M' usa la cinta 1 como entrada y la cinta 2 como cinta de trabajo para M. M' empieza con la configuración $\underline{\Delta}w$ en la primera cinta y, checa primero si el primer carácter a la derecha de la cabeza en la cinta 1 es Δ, en cuyo caso termina aceptando. Si no es Δ, pasa a un estado j. Desde j, si el el primer carácter a la derecha de la cabeza en la cinta 1 es Δ, regresa la cabeza de la cinta 2 al principio y ejecuta M en la cinta 2, y si es diferente de Δ, de manera no determinista, decide si baja el símbolo de la cinta 1 al final de la palabra en la cinta 2, lo borra en la cinta 1, recorre el contenido de la cinta 1 a la izquierda y regresa a j, o si regresa la cabeza de la cinta 2 al principio y comienza la ejecución de M en la cinta 2. Si M acepta y el símbolo inicial de la cinta 1 es Δ, M' termina aceptando. Si es diferente de Δ pasa al estado j, desde donde comienza otra vez a pasar, de manera no determinista, diferentes segmentos de la palabra restante para analizarlos.