Solución auxiliar 3

CC3102 Teoría de la Computación
Profesor: Gonzalo Navarro
Auxiliar: Pablo Muñoz
13 de agosto del 2014

P1. Responda veradero o falso. Justifique

a) Todo subconjunto de un lenguaje regular es regular.

Falso. Basta tomar un lenguaje como el de parentesis balanceado. Éste es subconjunto de $\{(,)\}^*$ el cual es regular.

b) Todo lenguaje regular tiene un subconjunto propio regular.

Falso. El conjunto vacío es regular, pero no tiene subconjuntos propios.

c) Si L es regular, también lo es$\{xy : x \in L, y \notin L\}$.

Verdadero. El lenguaje descrito corresponde a $L \cdot \bar{L}$ y los lenguajes regulares son cerrados bajo complementación y concatenación.

d) Si $\{L_i \ : \ i \in \mathbb{N}\}$ es una colección infinita de lenguajes regulares, entonces $\bigcup_i L_i$ es regular.

Falso. Considere un lenguaje no regular L (*e.g.* paréntesis balanceados). Para $n \geq 0$, sea $L_n = L \cap \{0,1\}^n$. L_n contiene todas las palabras de L de largo exactamente n. Luego $L = \cup_{n \geq 0} L_n$. Sin embargo, para cada n, L_n es regular (¿Por qué?).

P2. Pruebe que si L es regular, los siguientes lenguajes son regulares.

a) $Pref(L) = \{x \ : \ \exists y, xy \in L\}$

Sea $N(Q, \Sigma, \delta, s, F)$ un AFD que reconoce L. Considere $\tilde{F} = \{q \in Q : \exists w \in \Sigma^*,\ f \in F,\ (q, w) \vdash^* (f, \varepsilon)\}$. El automata $\tilde{N}(Q, \Sigma, \delta, s, \tilde{F})$ reconoce $Pref(L)$. En efecto, si $w \in Pref(L)$, entonces existe una palabra $x \in \Sigma^*$ tal que $wx \in L$. Sea $q_w \in Q$ el estado tal que $(s, w) \vdash^* (q_w, \varepsilon)$. Entonces, hay un estado final $f \in F$ tal que $(q_w, x) \vdash^* (f, \varepsilon)$. Sigue que $q_w \in \tilde{F}$ y $\tilde{N}$ acepta w. (La recíproca es similar, escríbala).

b) $Max(L) = \{w \in L \ : \ x \neq \varepsilon \ \Rightarrow \ wx \notin L\}$

Sea $N(Q, \Sigma, \delta, s, F)$ un AFD que reconoce L. Considere el conjunto de estados finales *no-terminales* de N : $R = \{q \in Q : \exists w \in \Sigma^*,\ f \in F,\ (q, w) \vdash^* (f, \varepsilon)\}$. El automata $\tilde{N}(Q, \Sigma, \delta, s, F \backslash R)$ reconoce $Max(L)$.

<u>OBS</u>: En realidad, hay un error en la definición de R para que haga lo que queremos. ¿Cómo se relacionan R y F c.r. a inclusión de conjuntos? ¿Que pasa cuando $w = \varepsilon$ en la definición de R ? Arregle la definición y demuestre que el automata reconoce $Max(L)$.

c) $L \backslash L' = \{w \in \Sigma^* \ : \ \exists x \in L', wx \in L\}$

Observe que no hemos dicho nada sobre que tipo de lenguaje es L'. En principio puede ser cualquiera!. Se hace una demostración no constructiva del caso general, y mostramos como construir un automata en el caso que L' es regular.

No constructivo: (más simple que lo que les conté en clases) Sea $N(Q, \Sigma, \delta, s, F)$ un AFD que reconoce L. Basta considerar, similar al item (a), el conjunto de estados finales $\tilde{F} = \{q \in Q : \exists w \in L',\ f \in F,\ (q, w) \vdash^* (f, \varepsilon)\}$. El automata $\tilde{N}(Q, \Sigma, \delta, s, \tilde{F})$ reconoce $L \backslash L'$.

Observe que si bien no podemos mostrar como construirlo, el automata está bien definido, y existe.

Constructivo : Si sabemos que L' es regular, entonces hay un AFD $N'(Q', \Sigma, \delta', s', F')$ que lo reconoce. Para saber si un estado $q \in Q$ está en $\tilde{F}$, considere el automata $N_q(Q, \Sigma, \delta, q, F)$ que es identico a N pero que tiene q como estado inicial. Podemos saber si $q \in \tilde{F}$ verificando si existe una palabra $w \ \in \Sigma^*$ en $\mathcal{L}(N_q) \cap \mathcal{L}(N')$. Si la intersección es vacía sabemos que q no está en $\tilde{F}$.

P3. Construya un automata que reconozca la intersección de dos lenguajes regulares dados, sin utilizar las propiedades de clausura vista en clases.

Sean $N_i(Q_i, \Sigma, \delta_i, s_i, F_i)$ dos AFD para $i = 1, 2$. Construimos el automata *producto* de N_1 y N_2 que reconoce la intersección de los lenguajes de estos automatas. El automata producto es $N(Q_1 \times Q_2, \Sigma, \delta, (s_1, s_2), F_1 \times F_2)$. Para cada $q_i, q_i' \in Q_i$, i = 1,2, y $a \in \Sigma$, $\delta(q_1, q_2, a) = (q_1', q_2')$ ssi $\delta_i(q_i, a) = q_i'$ para $i = 1, 2$. Sigue que $\mathcal{L}(N) = \mathcal{L}(N_1) \cap \mathcal{L}(N_2)$ (demuéstrelo).

P4. ¿Es posible decidir algorítmicamente si un lenguaje regular acepta una cantidad infinita de palabras?
Si. Una forma es obtener el automata del lenguaje regular. El lenguaje es infinito si y solo si el automata contiene un ciclo alcanzable que contiene un estado desde el cual se puede alcanzar algun estado final (demuestre el *si y solo si*). Sabiendo esto, podemos construir un algoritmo que busque estos ciclos en el automata.

Otra forma es demostrar lo siguiente : Sea L regular y p su largo de bombeo. Entonces, L es infinito si y solo si L tiene una palabra cuyo largo está entre p y $2p$. Si esto es cierto, entonces basta buscar entre todas estas palabras (son finitas) y ver si el automata acepta alguna. La propiedad es cierta: en efecto, si existe una palabra con cuyo largo está entre p y $2p$, por lema de bombeo podemos aumentar la parte cíclica de w tanto como queramos. Por otro lado, si L es infinito, debe contener una palabra de tamaño mayor a $2p$. Sea w una palabra en L de menor tamaño mayor a $2p$. Por lema de bombeo, $w = xyz$ con $|xy| \leq p$ y $|y| > 0$. Por el lema, $xz \in L$. Además, $p \geq |xz| \geq 2p$, puesto que le sacamos a w a lo más p símbolos, y puesto que no puede tener más que $2p$ símbolos (por minimalidad).