Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2223 — Teoría de Autómatas y Lenguajes Formales — 2' 2018

# PAUTA INTERROGACIÓN 1

## Pregunta 1

La pregunta pedía demostrar el lema del bombeo. Dada la formulación del lema propuesta, una demostración valida y revisada en clases es la siguiente.

Sea $L$ un lenguaje regular cualquiera. Por definición, debe existir un DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ tal que $L = \mathcal{L}(\mathcal{A})$. Fijaremos $N = |Q|$.

Ahora, para toda palabra $xyz \in L$ tal que $|y| \geq N$, como pertenece a $L$, entonces debe existir una ejecución de aceptación $\rho$ de $\mathcal{A}$ sobre $xyz$:

$$\rho: \; q_0 \underbrace{\xrightarrow{a_1} \ldots \xrightarrow{a_l}}_{x} q_l \overbrace{\xrightarrow{a_{l+1}} \ldots \xrightarrow{a_k}}^{y} q_k \underbrace{\xrightarrow{a_{k+1}} \ldots \xrightarrow{a_n}}_{z} q_n$$

La sub-ejecución $q_l \xrightarrow{a_{l+1}} \ldots \xrightarrow{a_k} q_k$ lee $|y|$ letras, por lo que considera $|y|+1$ estados. Esta cantidad de estados es mayor a $|Q|$ ya que $|y|+1 \geq N+1 > N = |Q|$. Luego, por el principio de palomar, al menos dos de esos $|y|+1$ estados, son iguales. Es decir, existen $r < s$ entre $l$ y $k$, tal que $q_r = q_s$.

A partir de estos, es posible realizar la subdivisión $y = uvw$, donde $v$ está limitado por los indices $r$ y $s$:

$$q_l \underbrace{\xrightarrow{a_{l+1}} \ldots \xrightarrow{a_r}}_{u} q_r \overbrace{\xrightarrow{a_{r+1}} \ldots \xrightarrow{a_s}}^{v} q_s \underbrace{\xrightarrow{a_{s+1}} \ldots \xrightarrow{a_k}}_{w} q_k$$

Donde $v \neq \epsilon$ ya que $r < s$. Ya que los estados que rodean a $v$ en esta sub-ejecución son el mismo, es posible recortar o repetir la sub-ejecución de $v$ y generar ejecuciones de aceptación validas de $\mathcal{A}$. En particular, si consideramos $i = 0$, la ejecución al recortar esa subsección sigue siendo valida y de aceptación:

$$q_0 \underbrace{\xrightarrow{a_1} \ldots \xrightarrow{a_l}}_{x} q_l \underbrace{\xrightarrow{a_{l+1}} \ldots \xrightarrow{a_r}}_{u} q_r = q_s \underbrace{\xrightarrow{a_{s+1}} \ldots \xrightarrow{a_k}}_{w} q_k \underbrace{\xrightarrow{a_{k+1}} \ldots \xrightarrow{a_n}}_{z} q_n$$

Y si consideramos $i > 0$, la ejecución al repetir $i$ veces esa subsección sigue siendo valida y de aceptación:

$$q_0 \underbrace{\xrightarrow{a_1} \ldots \xrightarrow{a_l}}_{x} q_l \underbrace{\xrightarrow{a_{l+1}} \ldots \xrightarrow{a_r}}_{u} \underbrace{q_r \overbrace{\xrightarrow{a_{r+1}} \ldots \xrightarrow{a_s}}^{v} q_s = q_r \to \ldots \to q_s = q_r \overbrace{\xrightarrow{a_{r+1}} \ldots \xrightarrow{a_s}}^{v} q_s}_{i \text{ veces}} = q_r \underbrace{\xrightarrow{a_{s+1}} \ldots \xrightarrow{a_k}}_{w} q_k \underbrace{\xrightarrow{a_{k+1}} \ldots \xrightarrow{a_n}}_{z} q_n$$

Lo cual muestra que para todo $i \geq 0$ se cumple $xuv^iwz \in L$. Al mostrar que para todo $L$ regular, existe $N > 0$, y para toda palabra $xyz$ existe subdivisión que cumple lo que dice el lema, queda demostrado.

Dado lo anterior, la distribución de puntaje es el siguiente:

- **(1 puntos)** Por reconocer la existencia del DFA $\mathcal{A}$ y fijar $N = |Q|$ (o cualquier valor mayor a el).
- **(0.5 puntos)** Por definir la ejecución de aceptación para una palabra cualquiera $xyz$ con $|y| \geq N$.
- **(2 puntos)** Por utilizar como argumento el principio del palomar para encontrar dos estados iguales en la sub-ejecución de $y$.
- **(1 puntos)** Por establecer, a partir de lo anterior, la subdivisión $y = uvw$, donde $v$ empieza y termina con el par de estados encontrados. **0.5 puntos** por la subdivisión y **0.5 puntos** por argumentar $v \neq \varepsilon$.
- **(1 puntos)** Por mostrar la construcción de ejecuciones de aceptación al repetir $i > 0$ veces la sub-ejecución de $v$ que muestran palabras que pertenecen a $L$.
- **(0.5 puntos)** Por incluir y explicar el caso cuando se recorta la sub-ejecución de $v$, que también pertenece al lenguaje $L$.

# Pregunta 2

### Pregunta 2.1

La solución consistía en entregar una expresión regular que cumpla con las restricciones. Un ejemplo sería:

- Caso normal: $(1 + 01)^* \cdot 001 \cdot (1 + 01)^* \cdot 00 \cdot (1 + 10)^*$
- Case borde: $(1 + 01)^* \cdot 000 \cdot (1 + 10)^*$

Y luego la expresión final es la disyunción de ambas.

Dado lo anterior, el puntaje asignado es el siguiente:

- **(2.5 puntos)** Por caso normal.
- **(0.5 puntos)** Por caso borde.

### Pregunta 2.2

Para esta pregunta, se esperaba que el alumno describiera lo que hacía la expresión regular, y después explicara por que lo que dijo esta correcto. Es decir, el alumno debía mencionar que la regex aceptaba **todas** las palabras con cantidad par de 1s y una cantidad par de 0s simultaneamente, y despues explicar por que la condición es suficiente y necesaria.

Dado lo anterior, el puntaje asignado es el siguiente:

- **(1 punto)** Por decir que las palabras aceptadas deben tener una cantidad par de 0s.
- **(1 punto)** Por decir que las palabras aceptadas deben tener una cantidad par de 1s.
- **(1 punto)** Por explicar por que las condiciones son suficientes y necesarias.

# Pregunta 3

### Pregunta 3.1

Para responder esta pregunta bastaba demostrar el contrapositivo del lema de bombeo. Una posible demostración es como sigue:

$\forall N \geq 0$

Se elige la palabra $a^N b^N \in S$, con $x = \varepsilon, y = a^N, z = b^N$

$\forall\ u \cdot v \cdot w = y = a^N$ con $v \neq \varepsilon$

Se elige $i = 3$, con lo que $x \cdot u \cdot v^3 \cdot w \cdot z \notin S$

Había que justificar que $a^N b^N \in S$, mientas que $x \cdot u \cdot v^i \cdot w \cdot z \notin S$. Hecho esto se demuestra que no se cumple el lema de bombeo, lo que implica que $S$ no es regular.

Dado lo anterior, el puntaje asignado es el siguiente:

- **(1 punto)** Por escoger y dividir correctamente la palabra a bombear.
- **(1 punto)** Por bombear correctamente (escoger el valor de $i$ y mostrar que $x \cdot u \cdot v^i \cdot w \cdot z \notin S$).
- **(1 punto)** Por la justificación de cada uno de los pasos.

**Pregunta 3.2**

Dado que $L$ es un lenguaje regular existe un autómata finito no determinista $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$ que lo acepta. Luego podemos construir un autómata $\mathcal{A}^{mr}$ que acepte $L^{mr}$. Si hacemos eso, quedaría demostrado que $L^{mr}$ es un lenguaje regular por definición.

Una posible manera de construir $\mathcal{A}^{mr}$ es $\mathcal{A}^{mr} = (Q', \Sigma \cup \{\#\}, \Delta', I, F')$ tal que:

$$
\begin{aligned}
Q' &= Q \cup Q \times \Sigma \cup \Sigma \cup \{q_f\} \\
F' &= \{q_f\} \\
\Delta' &= \Delta \cup \{(p, \varepsilon, (q, a)) | (p, a, q) \in \Delta\} \\
&\quad \cup \{((p, a), b, (q, a)) | (p, b, q) \in \Delta \wedge a \in \Sigma\} \\
&\quad \cup \{((p, a), \#, a) | p \in F \wedge a \in \Sigma\} \\
&\quad \cup \{(a, a, q_f) | a \in \Sigma\}
\end{aligned}
$$

Luego se debe demostrar que $\mathcal{L}(\mathcal{A}^{mr}) = L^{mr}$.

Dado lo anterior, el puntaje asignado es el siguiente:

- **(2 puntos)** Por construir correctamente el autómata $\mathcal{A}^{mr}$.
- **(1 punto)** Por demostar correctamente que $\mathcal{L}(\mathcal{A}^{mr}) = L^{mr}$.

# Pregunta 4

Para una posible solución podemos introducir el símbolo $\preceq_p$ tal que, con $u, v \in \Sigma^*$, $u \preceq_p v$ ssi $u$ es prefijo de $v$. Con esta intuición, decimos que dos lenguajes $L$ y $R$,

$$L \preceq_p R \ \leftrightarrow\ \forall v \in R.\ \exists u \in L.\ u \preceq_p v,$$

es decir $L \preceq_p R$ ssi $L$ es prefijo de $R$. Es claro entonces que

$$\neg(L \preceq_p R) \ \leftrightarrow\ \exists v \in R.\ \forall u \in L.\ \neg u \preceq_p v.$$

A partir de esta noción podemos definir un nuevo lenguaje a partir de $L$ y $R$:

$$L \not\preceq_p R = \{v \in R \mid \forall u \in L.\ \neg u \preceq_p v\}.$$

La idea será crear un autómata $\mathcal{C}$ a partir de $\mathcal{A}$ y $\mathcal{B}$ tal que $\mathcal{L}(\mathcal{C}) = \mathcal{L}(\mathcal{A}) \not\preceq_p \mathcal{L}(\mathcal{B})$, usar el algoritmo de *non emptiness* para verificar si $\mathcal{L}(\mathcal{C}) \neq \varnothing$ y en base a eso entregar una respuesta. Así dados

$$\mathcal{A} = (Q_{\mathcal{A}}, \Sigma, \delta_{\mathcal{A}}, q_0^{\mathcal{A}}, F_{\mathcal{A}})$$
$$\mathcal{B} = (Q_{\mathcal{B}}, \Sigma, \delta_{\mathcal{B}}, q_0^{\mathcal{B}}, F_{\mathcal{B}}),$$

definimos un nuevo autómata $\mathcal{C}$:

$$\mathcal{C} = (Q_{\mathcal{A}} \times Q_{\mathcal{B}}, \Sigma, \delta, (q_0^{\mathcal{A}}, q_0^{\mathcal{B}}), Q_{\mathcal{A}} \times F_{\mathcal{B}}),$$

tal que

$$\delta((p_1, q_1), a) = (\delta_{\mathcal{A}}(p_1, a), \delta_{\mathcal{B}}(p_2, a)) \leftrightarrow p_1 \notin F_{\mathcal{A}}.$$

Luego de esto se debe demostrar que $\mathcal{L}(\mathcal{C}) = \mathcal{L}(\mathcal{A}) \not\preceq_p \mathcal{L}(\mathcal{B})$. El algoritmo que se pide es el siguiente:

```
function PREF_LENG(A, B)
    C ← construir C tal que L(C) = L(A) ⋠_p L(B)
    if non_empty(C) then
        return FALSE
    end if
    return TRUE
end function
```

Finalmente se debe explicar la correctitud de este algoritmo.

Dado lo anterior, el puntaje asignado es el siguiente:

- **(3 puntos)** Por construir correctamente el autómata $\mathcal{C}$ a partir de los autómatas $\mathcal{A}$ y $\mathcal{B}$.
- **(1 punto)** Por demostrar que $\mathcal{L}(\mathcal{C}) = \mathcal{L}(\mathcal{A}) \not\preceq_p \mathcal{L}(\mathcal{B})$.
- **(1 punto)** Por escribir correctamente el algoritmo a partir del autómata $\mathcal{C}$.
- **(1 punto)** Por explicar la correctitud del algoritmo.