UNIVERSIDAD TECNICA FEDERICO SANTA MARIA | DEPARTAMENTO DE INFORMÁTICA

INF155 - Informática Teórica

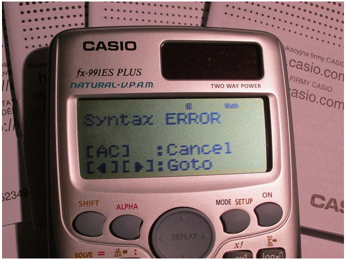
Ayudantía #1

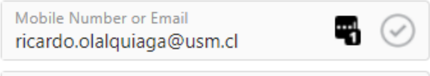
NP-Complete Warriors

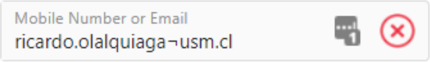
Semestre 2022-2

Motivación

¿Para qué sirven las expresiones regulares?







Conceptos

- **Alfabeto:** conjunto finito de símbolos atómicos. Denotados con letras griegas mayúsculas: $\Sigma, \Gamma, \Delta, \ldots$
- **Palabra (*string*):** una palabra sobre un alfabeto Σ es una secuencia finita de símbolos de dicho alfabeto. Denotados con letras griegas minúsculas: $\alpha, \beta, \gamma, \ldots$
- El **largo** de una palabra σ cualquiera se denota como $|\sigma|$ (también suele denotarse como $\lg(\sigma)$). Corresponde a la cantidad de símbolos de la misma.
- La secuencia de símbolos vacía (o la palabra vacía), cuyo largo es 0 se denota con la letra griega ε.
- Un **lenguaje** sobre el alfabeto Σ es un conjunto de palabras sobre Σ. Puede ser finito o infinito.

Conceptos - Ejemplo

Consideremos el siguiente alfabeto:

$$\Sigma = \{0, 1\}$$

Una palabra σ podría ser 000111. El largo de dicha palabra se denota como:

$$|\sigma| = 6 \text{ (ya que tiene 6 símbolos)}$$

Un lenguaje sobre Σ podría ser:

$$\mathscr{L} = \{001, 010, 100\}$$

que corresponde al lenguaje de palabras sobre Σ que tienen largo 3 y que contienen un *1*.

Operaciones entre palabras

Sea Σ un alfabeto, y sean

$$\alpha = a_1 a_2 a_3 \dots a_m$$
$$\beta = b_1 b_2 b_3 \dots b_n$$

palabras sobre Σ. Se define la **concatenación** de α y β como:

$$\alpha \cdot \beta = \alpha\beta = a_1 a_2 a_3 \dots a_m b_1 b_2 b_3 \dots b_n$$

La concatenación:

- En general, **no es conmutativa**, es decir, $\alpha\beta \neq \beta\alpha$.
- Es asociativa, es decir $(\alpha\beta)\gamma = \alpha(\beta\gamma)$
- Se cumple que $\varepsilon \cdot \alpha = \alpha \cdot \varepsilon = \alpha$
- Se cumple que $|\alpha\beta| = |\alpha| + |\beta|$

Operaciones entre palabras

Para $n \in \mathbb{N}_0$ se define la operación de **potencia** como se muestra a continuación:

$$\alpha^n = \begin{cases} \varepsilon & si \quad n = 0 \\ \alpha^{n-1} \cdot \alpha & si \quad n \geq 1 \end{cases}$$

Operaciones entre palabras - Ejemplo

Consideremos las siguientes palabras sobre $\Sigma = \{a, b, c\}$:

$$\sigma = aaabbbccc$$
$$\omega = abc$$

Entonces:

$$\sigma \cdot \omega = aaabbbcccabc$$
$$\omega \cdot \sigma = abcaaabbbccc$$

Note que $\sigma \cdot \omega \neq \omega \cdot \sigma$, y que $|\sigma \cdot \omega| = |\omega \cdot \sigma| = |\omega| + |\sigma| = 12$.

Operaciones entre palabras - Ejemplo

Consideremos la palabra $\omega = abc$. Luego, se tiene que:

$$\omega^3 = \omega \cdot \omega \cdot \omega = abcabcabc$$

Note que $|\omega^3| = 3 \cdot |\omega| = 9$.

Operaciones entre lenguajes

Sean $\mathscr{L}_1$ y $\mathscr{L}_2$ lenguajes sobre Σ. Se definen las siguientes operaciones:

Unión o alternancia

$$\mathscr{L}_1 \mid \mathscr{L}_2 = \{\alpha : \alpha \in \mathscr{L}_1 \cup \mathscr{L}_2\}$$

Interpretación: escojo una palabra de $\mathscr{L}_1$ o una palabra de $\mathscr{L}_2$.

Concatenación

$$\mathscr{L}_1 \cdot \mathscr{L}_2 = \{\alpha \cdot \beta : \alpha \in \mathscr{L}_1 \wedge \beta \in \mathscr{L}_2\}$$

Interpretación: escojo una palabra de $\mathscr{L}_1$ y le concateno una palabra de $\mathscr{L}_2$.

Observación: en general, $\mathscr{L}_1 \cdot \mathscr{L}_2 \neq \mathscr{L}_2 \cdot \mathscr{L}_1$.

Operaciones entre lenguajes

Potencia

$$\mathscr{L}_1^n = \begin{cases} \{\varepsilon\} & si \quad n = 0 \\ \mathscr{L}_1^{n-1} \cdot \mathscr{L}_1 & si \quad n \geq 1 \end{cases}$$

Interpretación: escojo n palabras de $\mathscr{L}_1$ y las concateno de forma consecutiva.

Estrella de Kleene

$$\mathscr{L}_1^* = \bigcup_{n \geq 0} \mathscr{L}_1^n$$

Interpretación: escojo 0 o más palabras de $\mathscr{L}_1$ y las concateno de forma consecutiva.

Operaciones entre lenguajes

Kleene plus

$$\mathscr{L}_1^+ = \bigcup_{n\geq 1} \mathscr{L}_1^n$$

Interpretación: escojo 1 o más palabras de $\mathscr{L}_1$ y las concateno de forma consecutiva.

Aplicando las definiciones anteriores, se explica que:

- $\mathscr{L}_1^+ = \mathscr{L}_1 \cdot \mathscr{L}_1^* = \mathscr{L}_1^* \cdot \mathscr{L}_1$
- $\mathscr{L}_1^* = \mathscr{L}_1^+ \cup \{\varepsilon\}$

Operaciones entre lenguajes

Dado que los lenguajes son conjuntos, heredan sus operaciones. Sean $\mathscr{L}_1$, $\mathscr{L}_2$ y $\mathscr{L}_3$ lenguajes sobre Σ. Entonces, se cumple que:

- $\mathscr{L}_1 \mid \mathscr{L}_1 = \mathscr{L}_1$
- $\mathscr{L}_1 \mid \mathscr{L}_2 = \mathscr{L}_2 \mid \mathscr{L}_1$
- $\mathscr{L}_1 \mid (\mathscr{L}_2 \mid \mathscr{L}_3) = (\mathscr{L}_1 \mid \mathscr{L}_2) \mid \mathscr{L}_3$
- $\varnothing \mid \mathscr{L}_1 = \mathscr{L}_1 \mid \varnothing = \mathscr{L}_1$
- $\{\varepsilon\} \cdot \mathscr{L}_1 = \mathscr{L}_1 \cdot \{\varepsilon\} = \mathscr{L}_1$
- $\varnothing \cdot \mathscr{L}_1 = \mathscr{L}_1 \cdot \varnothing = \varnothing$
- $(\mathscr{L}_1 \cdot \mathscr{L}_2) \cdot \mathscr{L}_3 = \mathscr{L}_1 \cdot (\mathscr{L}_2 \cdot \mathscr{L}_3)$
- $(\mathscr{L}_1 \mid \mathscr{L}_2) \cdot \mathscr{L}_3 = (\mathscr{L}_1 \cdot \mathscr{L}_3) \mid (\mathscr{L}_2 \cdot \mathscr{L}_3)$
- $\mathscr{L}_1 \cdot (\mathscr{L}_2 \mid \mathscr{L}_3) = (\mathscr{L}_1 \cdot \mathscr{L}_2) \mid (\mathscr{L}_1 \cdot \mathscr{L}_3)$

Operaciones entre lenguajes

Algunas observaciones:

- No confundir $\varnothing$, ε y $\{\varepsilon\}$.
- Nuestra definición dice que $\varnothing^0 = \{\varepsilon\}$, por lo que se tiene que:
  - $\varnothing^* = \{\varepsilon\}$
  - $\varnothing^+ = \varnothing$
- La precedencia de las operaciones que se utiliza es la siguiente:
  1. Kleene, Kleene plus, Potencias.
  2. Concatenaciones.
  3. Uniones.

Operaciones entre lenguajes - Ejemplo

Sea $\Sigma = \{a, b, c, d\}$. Consideremos los lenguajes $\mathscr{L}_1$ y $\mathscr{L}_2$ definidos sobre Σ como:

$$\mathscr{L}_1 = \{aaa, ccc\}$$
$$\mathscr{L}_2 = \{bbb, ddd\}$$

Entonces:

- $\mathscr{L}_1 \cup \mathscr{L}_2 = \{aaa, bbb, ddd, ccc\}$
- $\mathscr{L}_1 \cap \mathscr{L}_2 = \varnothing$
- $\mathscr{L}_1^* = \{\varepsilon, aaa, ccc, aaaccc, cccaaa, \ldots\}$
- $\mathscr{L}_2^+ = \{bbb, ddd, bbbddd, dddbbb, \ldots\}$

Expresiones Regulares (RE)

Sea Σ un alfabeto. Definimos *expresiones regulares* (RE) y a los lenguajes que denotan como se muestra a continuación:

- $\varnothing$ denota $\varnothing$
- ε denota $\{\varepsilon\}$
- Para $a \in \Sigma$, a denota $\{a\}$.

Sean R y S expresiones regulares que denotan a $\mathscr{L}(R)$ y $\mathscr{L}(S)$, respectivamente. Entonces:

- $R \cdot S$ denota $\mathscr{L}(R) \cdot \mathscr{L}(S)$.
- $R \mid S$ denota $\mathscr{L}(R) \cup \mathscr{L}(S)$.
- R^* denota $\mathscr{L}(R)^*$.
- R^+ denota $\mathscr{L}(R)^+ = \mathscr{L}(R) \cdot \mathscr{L}(R)^*$.

Extraoficialmente, R^n denota $\mathscr{L}(R)^n$ (con $n \in \mathbb{N}_0$).

Expresiones Regulares (RE)

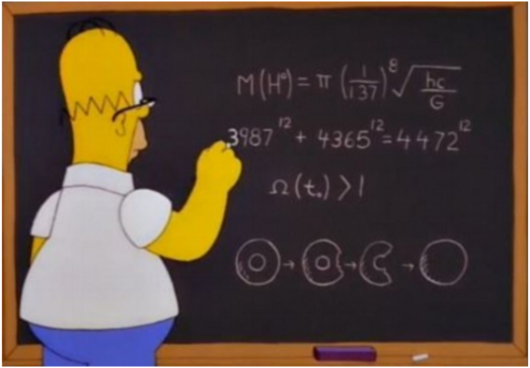
Algunas observaciones:

- Para un alfabeto Σ utilizaremos Σ^* para denotar al lenguaje de todas las palabras que se pueden formar con los símbolos de Σ.
- La precedencia de las operaciones que se utiliza es la siguiente:
  1. Kleene, Kleene plus, Potencias.
  2. Concatenaciones.
  3. Uniones.

Algunas propiedades:

- $\mathscr{L}((R^*)^*) = \mathscr{L}(R^*)$
- $\mathscr{L}(R^*) = \mathscr{L}(R^+ \mid \varepsilon)$
- $\mathscr{L}(R^*R^*) = \mathscr{L}(R^*)$
- $\mathscr{L}(R^+R^*) = \mathscr{L}(R^*R^+) = \mathscr{L}(R^+)$

Ejercicios



RE

Sea $\Sigma = \{a, b, c\}$. Tenemos las siguientes RE posibles:

RE 1

Palabras sobre Σ que contienen una cantidad par de a's.

RE 2

Palabras de largo impar, cuyo último símbolo es una c.

RE 3

Palabras que contienen exactamente un símbolo b.

RE - Soluciones propuestas

Sea $\Sigma = \{a, b, c\}$. Tenemos las siguientes RE posibles:

RE 1

$$((b \mid c)^* a (b \mid c)^* a (b \mid c)^*)^*$$

Palabras sobre Σ que contienen una cantidad par de a's.

RE 2

$$(\Sigma\Sigma)^* c$$

$$((a \mid b \mid c)(a \mid b \mid c))^* c$$

Palabras de largo impar, cuyo último símbolo es una c.

RE 3

$$(a \mid c)^* b (a \mid c)^*$$

Palabras que contienen exactamente un símbolo b.

Ejercicio 1

1. Sean $\boldsymbol{u}$ y $\boldsymbol{v}$ dos *strings*, y $\boldsymbol{n}$ y $\boldsymbol{m}$ dos números naturales. Demuestre que:
$$|u^n v^m| = n|u| + m|v|$$
2. Encuentre el lenguaje $\mathscr{L}$ tal que para todo $\mathscr{L}_\Sigma$ siempre se cumple que:
$$\mathscr{L} \cdot \mathscr{L}_\Sigma = \mathscr{L}_\Sigma \cdot \mathscr{L}$$

Solución Ejercicio 1.1

Hay 2 formas equivalentes para enfrentar este ejercicio.

- **Forma 1:** Sea $\sigma = u^n v^m$ la palabra en cuestión. Luego:

$$\sigma = u \cdot u \cdot u \cdots u \cdot v \cdot v \cdots v$$

El string u se repite n veces y el string v se repite m veces. Luego:

$$|\sigma| = |u^n v^m| = n \cdot |u| + m \cdot |v|$$

$\square$

- **Forma 2:** aplicando noción de logaritmo.

$$\log(u^n v^m) = \log(u^n) + \log(v^m) = n \cdot \log(u) + m \cdot \log(v) = n \cdot |u| + m \cdot |v|$$

$\square$

Solución Ejercicio 1.2

La demostración es sencilla. Si recordamos la definición de concatenación, sabemos que para un lenguaje $\mathscr{L}_\Sigma$ cualquiera se cumple que:

$$\{\varepsilon\} \cdot \mathscr{L}_\Sigma = \mathscr{L}_\Sigma \cdot \{\varepsilon\} = \mathscr{L}_\Sigma$$

Luego, es claro que el lenguaje $\mathscr{L}$ debe ser $\{\varepsilon\}$.

$\square$

Ejercicio 2

Se define el reverso de un lenguaje como $\mathscr{L}^R = \{\omega^R \mid \omega \in \mathscr{L}\}$, demuestre que se cumplen las siguientes afirmaciones:

a. $(\mathscr{L}_1 \cdot \mathscr{L}_2)^R = \mathscr{L}_2^R \cdot \mathscr{L}_1^R$
b. $(\mathscr{L}^R)^* = (\mathscr{L}^*)^R$
c. $(\mathscr{L}_1 \cup \mathscr{L}_2)^R = \mathscr{L}_1^R \cup \mathscr{L}_2^R$

Solución Ejercicio 2

Antes de comenzar a desarrollar los ejercicios planteados, conviene realizar las siguientes definiciones:

- Sea $\alpha \in \mathscr{L}_1$ tal que $\alpha = a_1 a_2 \ldots a_m$
- De forma análoga se define $\beta \in \mathscr{L}_2$ tal que $\beta = b_1 b_2 \ldots b_n$

Es decir, α y β representan a una palabra cualquiera de los respectivos lenguajes.

a. Demostración directa aplicando la definición de concatenación:

$$\begin{aligned} \mathscr{L}_1 \cdot \mathscr{L}_2 &= \alpha\beta = a_1 a_2 \ldots a_m b_1 b_2 \ldots b_n \qquad /()^R \\ (\mathscr{L}_1 \cdot \mathscr{L}_2)^R &= (\alpha\beta)^R = b_n \ldots b_2 b_1 a_m \ldots a_2 a_1 \\ &= \beta^R \alpha^R = \mathscr{L}_2^R \cdot \mathscr{L}_1^R \end{aligned}$$

$\square$

Solución Ejercicio 2

b. Sin pérdida de generalidad, suponemos que $\mathscr{L} = \mathscr{L}_1$. Luego:

$$\begin{aligned}\mathscr{L}^* &= \varepsilon \mid \alpha \mid \alpha\alpha \mid \alpha\alpha\alpha \ldots \qquad / ()^R \\ (\mathscr{L}^*)^R &= \varepsilon \mid a_m \cdots a_2 a_1 \mid a_m \cdots a_2 a_1 a_m \cdots a_2 a_1 \mid \ldots \\ &= \varepsilon \mid \alpha^R \mid \alpha^R \alpha^R \mid \alpha^R \alpha^R \alpha^R \mid \ldots \\ &= \varepsilon \mid \alpha^R \mid (\alpha\alpha)^R \mid (\alpha\alpha\alpha)^R \mid \cdots = (\mathscr{L}^R)^*\end{aligned}$$

$\square$

c. Dado que la unión nos permite escoger una palabra de un lenguaje o del otro, la demostración es sencilla:

$$\begin{aligned}\mathscr{L}_1 \cup \mathscr{L}_2 &= \alpha \cup \beta \qquad ()^R \\ (\mathscr{L}_1 \cup \mathscr{L}_2)^R &= \alpha^R \cup \beta^R = \mathscr{L}_1^R \cup \mathscr{L}_2^R\end{aligned}$$

$\square$

Ejercicio 3

Una expresión regular se llama *ambigua* si hay alguna palabra que se puede describir de dos o más maneras. Determine cuales de las siguientes RE son ambiguas:

a. $a((ab)^*cd)^* \mid a(ababcd^*)^*a^*$

b. $aab^*(ab)^* \mid ab^* \mid abba^*$

c. $aaba^* \mid aba \mid aabba^+a \mid a$

Solución Ejercicio 3

a. La RE derecha pareciese ser un *caso particular* de la RE izquierda. Si obtenemos la palabra mínima en ambos casos (es decir, escogemos 0 repeticiones en el operador *), obtenemos la palabra 'a' en ambos casos. Luego, la RE es ambigua.
b. Siguiendo el razonamiento anterior, consideramos sólo la RE del centro y la de la derecha para realizar la comparación, ya que la primera RE es distinta. Si en la RE del centro se repite dos veces la letra b y en la RE de la derecha escogemos 0 repeticiones de la letra a, entonces obtenemos la palabra 'abb' en ambas. Luego, la RE es ambigua.

Solución Ejercicio 3

c. Nuevamente obtenemos la palabra mínima para cada una de las sub-expresiones regulares:
   - aab
   - aba
   - $aabbaa$
   - a

   En todas obtenemos palabras distintas, y no hay forma equivalente de generarlas a partir de otras RE. Luego, la RE no es ambigua.

¿Dudas?

INF155 - Informática Teórica

Ayudantía #1

NP-Complete Warriors

Semestre 2022-2