Lenguajes más pequeños sobre $\Sigma=\{a,b\}$
{}
{ ε }
{a}, {b}
{ aa }={a}{a}, {ab}={a}{b}, { bb }={b}{b}, { ba }={b}{a}
{ a , b}={a}U{b} , {ε, a}={ε}U{a}, {ε, b}={ε}U{b}

Definición de lenguajes regulares sobre un alfabeto Σ.

a. {} es un lenguaje regular.
b. {ε} es un lenguaje regular.
c. Para todo $\sigma \in \Sigma$, { σ } es un lenguaje regular. (lenguajes unitarios)
d. Si A y B son lenguajes regulares, entonces AUB, AB, A* son lenguajes regulares.
e. Ningún otro lenguaje sobre Σ es regular.

Por tanto, el conjunto de los lenguajes regulares sobre Σ está formado por el lenguaje vacío, los lenguajes unitarios incluido {ε} y todos los Lenguajes obtenidos a partir de la concatenación, unión y cerradura de estrella de lenguajes.

Ejercicio:
Verificar, aplicando la definición de lenguaje regular, que los Siguientes son lenguajes regulares sobre $\Sigma=\{a,b\}$:

$L1=\{ a^i \mid i>0\}$
L1 = { a, aa, aaa, aaaa, … }
Σ*-L1 = { ε, b, ab, bb, ba, abb, bab, bbb,… }

{a}{a}* = {a}{ε, a, aa, aaa, …} = { a, aa, aaa, aaaa, …} =L1

L1 se puede construir a partir de la estrella de un lenguaje unitario concatenada con un lenguaje unitario.
Por lo tanto L1 cumple con la definición y L1 es un lenguaje regular.

$L2=\{ a^i \mid i > n \}$ para un $n\geq 0$ fijado.
Por ejemplo n=3.
L2={ aaaa, aaaaa, aaaaaa, … }
Σ*-L2={ ε, b, a, bab, ba, aa, bbbb, baaa, aaaab, … }
En general:
$L2=\{ a^{n+1}, a^{n+2}, \ldots \}$
$\Sigma^*-L2=\{ a^0, \ldots , a^n, b, ab, \ldots \}$

$(\{a\}^*)(\{a\}^{n+1})=\{ a^{n+1}, a^{n+2}, a^{n+2}, \ldots \} = L2$

L2 se puede construir a partir de la estrella de un lenguaje unitario concatenada con la potencia (concatenación repetida) de un lenguaje unitario. Por lo tanto L2 es regular.

L3= { w1, w2, w3, …, wn } n<infinito

Para construir w_i, podemos proceder:
w_i= σ1σ2σ3σ4…σr, donde r = |w| y $σ_j$єΣ
{w_i}={σ1}{σ2}{σ3}{σ4}…{σr}

Y luego
L3= {w1}U{w2}U{w3}U…U{wn}

L3 se puede construir mediante la unión de los lenguajes que tienen una sola cadena, en los que cada uno tiene una cadena de L3, y estos lenguajes de una sola cadena se pueden Construir mediante concatenaciones de lenguajes unitarios. Por lo tanto L3 es regular.

L4={ cadenas que tienen la subcadena bb }
L4={ bb, abb, bba, bbb, abba, bbba, bbbb, … }
Σ*-L4={ ε, a, b, aa, ba, ab, bab, baba, abab, … }

{bb} = {b}{b}
{bb}({a} U {b})* = { bb, bba, bbb, bbab, bbba, bbaa, …}
({a} U {b})*{bb}({a} U {b})* = { bb, abb, bba, bbb, … } =L4

L4 se puede construir a partir de uniones, concatenaciones y estrella de lenguajes unitarios. Por lo tanto, L4 es regular.

Podemos simplificar la especificación de un lenguaje regular introduciendo una especie de abreviatura llamada
expresión regular. Convenimos en escribir

a   denota {a}
ε   denota {ε}
a U b denota {a} U {b}
ab   denota {a}{b}
a*   denota {a}*

1. Φ y ε son expresiones regulares.
2. a es una expresión regular para todo a єΣ.
3. Si r y s son expresiones regulares, entonces rUs, rs y r* también lo son.
4. Ninguna otra secuencia de símbolos es una expresión regular.

X={ 0, 00, 01, 010}
Y={ε, 1, 11}
XUY={ 0, 00, 01, 010, ε, 1, 11 }
XUY={ ε, 0, 1, 00, 01, 11, 010 }

Ejemplos de expresiones regulares:

(a U b*)c

aU b* = {a} U {b}* ={a} U { ε, b, bb, bbb, bbbb, …}
aUb* = { ε, a, b, bb, bbb, bbbb, … }
(aUb*)c = {ε, a, b, bb, bbb, …}{c} ={ c, ac, bc, bbc, bbbc, …}

L((aUb*) c) ={ c, ac, bc, bbc, bbbc, …}

28oct2020

Verificar, aplicando la definición de lenguaje regular,
que el siguiente lenguaje sobre Σ={0,1} es regular:

L={ cadenas en las que el antepenúltimo símbolo es un 0 }

L={ 000, 001, 010, 011, 1000, 00000000, 1010101000,…}
Σ*-L={ ε, 0, 1, 00, 10, 11, 100, 1111111111, 000000100,…}

({0}U{1})*{0}({0}U{1})({0}U{1})

({ 0, 1})*= { ε, 0, 1, 00, 11, 10, 01, 000, 001, 010, 011, 100, 101, 110, 111,…}

L puede ser construido a partir de lenguajes unitarios, uniones, concatenaciones
y cerradura estrella, por lo tanto L es regular.

L2={ a^i | i > n } para un n≥0 fijado.
Por ejemplo n=3.
L2={ aaaa, aaaaa, aaaaaa, … }
Σ*-L2={ ε, b, a, bab, ba, aa, bbbb, baaa, aaaab, … }
En general:
L2={ a^{n+1}, a^{n+2}, … }
Σ*-L2={ a^0, … , a^n, b, ab, … }

({a}*)($\{a\}^{n+1}$)={ a^{n+1}, a^{n+2}, a^{n+2}, … } = L2

L3={ $a^n b^n$ }
L3={ ε, ab, aabb, aaabbb, aaaabbbb, … }
Σ*-L3={ a, b, ba, aab, abb, ababab, … }

($\{a\}^n$)($\{b\}^n$) No es un argumento para concluir que L3 es regular porque
n no es fijo.

L4= { $a^n b^n$ } para n<5.

L4={ ε, ab, aabb, aaabbb, aaabbbb }
Σ*-L4 ={ a, b, ba, abb, bababa, abba, …}

{ε}U{a}{b}U{a}{a}{b}{b}U{a}{a}{a}{b}{b}{b}U{a}{a}{a}{a}{b}{b}{b}{b} =L4

Encontrar las cinco cadenas más pequeñas del lenguaje especificado
por la siguiente expresión regular. También las cinco cadenas más
pequeñas de su complemento.

(a U (ba))*(b*)

L5((a U (ba))*(b*))= { ε, a, b, ab, ba, aba, aaaaa, baba, bbbbb, abbb, … }
Σ*-L5={ bba, bbba, bbbba, abba, bbaa, bbab, … }

L5= { cadenas en las que la subcadena bb no se encuentre antes de una a }

(a U (ba))*(b*) = ({a} U ({b}{a}))*({b}*)
= ({a} U {ba})*({ε, b, bb, …}) = ({a, ba})*({b}*)
= ({ ε, a, ba, aa, baba, aba, baa, aaa, bababa, ababa, baaba, …})({b}*)
={ ε, b, a, ab, aba, abab, …} aba se forma al tomar una vez a y una vez ba de {a, ba}
aba se forma al concatenar **aba** de {a,ba}* con **ε** de {b}*.

Escribir una expresión regular para los siguientes lenguajes Σ={a,b}:

L1={ cadenas que tengan al menos una a y al menos una b }
L1={ ab, ba, abbbb, baaaa, babab, abbbba, aaaaaaaaaaaab, bbbabbbb,…}
Σ*-L1={ ε, a, b, aaaaa, bbbbb, a^n, b^n, … }

{a} = a
{b} = b

((aUb)*ab(aUb)* U ((aUb)*ba(aUb)*)

(aUb)*(ab U ba)(aUb)*

(aUb)*(a(aUb)*b U b(aUb)*a)(aUb)*

L2={ cadenas que tengan un número par de b's } (considerando el número total de b's)

L2={ ε, a^n, bb, abb, baab, bbaa, bbbb, abababaaaaab,…}
Σ*-L2={ b, aaab, bbb, abbaaabaaaa, … }

(a)*(b(a*)b)*(a)*

L3={ cadenas que tengan un número impar de b's } (considerando el número total de b's)
L3={ b, aaabaaaa, bbb, abbaaabaa, …}
Σ*-L3= { ε, aaaaa, bb, aaabaaaba, …}

((a)*(b(a*)b)*(a))* b ((a)*(b(a*)b)*(a))*

3nov2020

$(a^*b^*)^* = (\{a\}^*\{b\}^*)^* = (\{ ε, a, aa, aaa, …\}\{ε, b,bb,bbb,…\})^*$
$=(\{ ε, a, b, ab, abb, aab, … \})^* = \{a,b\}^* = Σ^* = (a \cup b)^*$

Si r y s son expresiones regulares sobre el mismo alfabeto y si
L(r) = L(s), entonces se dice que r y s son equivalentes. En el caso de
Que r y s sean equivalente se puede escribir r=s.

$(a^*b^*)^* = (a \cup b)^* =Σ^*$

$(a^*b)^* = (\{a\}^*\{b\})^* = (\{ε,a,aa,aaa,…\}\{b\})^*= \{b,ab,aab,aaab,…\}^*$
$=\{ε,b,bb,ab,bbb,bab,abb,…\}$
={ cadenas sobre Σ={a,b} que sean ε o que terminen con b }

$ε\cup(a\cup b)^*b = \{ε\}\cup \{a,b\}^*\{b\}=\{ε\} \cup \{ ε, a, b, ab,ba, aa,bb,…\}\{b\}$
$=\{ε\} \cup \{b,ab,bb,abb,bab,aab,bbb,…\}$
$=\{ε,b,ab,bb,abb,bab,aab,bbb,…\}$

$(a^*b)^* = ε\cup(a\cup b)^*b$

Sean r, s, t expresiones regulares sobre el mismo alfabeto Σ.
Entonces:

1. $r\cup s = s\cup r$.
2. $r\cup\emptyset=\emptyset\cup r=r$.
3. $r\cup r = r$.
4. $(r\cup s)\cup t = r\cup(s\cup t)$.
5. $rε = εr =r$.
6. $r\emptyset = \emptyset r = \emptyset$.
7. $(rs)t = r(st)$.
8. $r(s\cup t) = rs \cup rt$ $\qquad$ $(r\cup s)t = rt \cup st$.
9. $r^* = r^{**} = r^*r^* = (ε\cup r)^* = r^*(r\cup ε) = (r\cup ε)r^* = ε\cup rr^*$.
10. $(r\cup s)^* = (r^*\cup s^*)^* = (r^*s^*)^* = (r^*s)^*r^* = r^*(sr^*)^*$.
11. $r(sr)^* = (rs)^*r$.
12. $(r^*s)^* = ε\cup(r\cup s)^*s$.
13. $(rs^*)^* = ε\cup r(r\cup s)^*$.
14. $s(r\cup ε)^*(r\cup ε)\cup s = sr^*$.
15. $rr^* = r^*r$.

Expresiones regulares en Linux (regex)

'a..zA..Z0..9'	lenguajes unitarios
.	Cualquier carácter
[]	Rango [0-9] [a-z]
|	Indica una de varias opciones
*	Sirve para denotar que la regexp precedente se repite cero o más veces
+	Denota que la regesp precedente se repite una o más veces
()	Definen un grupo
?	Especifica que una parte de la regexp es opcional.
{n}	La regexp precedente se repite n veces
{m,n}	La regexp precedente se repite al menos m veces y máximo n veces
$	Representa el final de la cadena de caracteres o el final de la línea.
^	Como ancla, representa el inicio de la línea. En un rango, significa la negación de los lenguajes unitarios dentro del rango. Ejemplo: [^0-9] ningún dígito
\	Permite usar caracteres especiales como lenguajes unitarios. Ejemplo: \$
\< \>	Inicio, fin de palabra

Ejemplos:

'Linux'   genera la cadena 'Linux'
'CentO.' genera las cadenas {'CentOa', 'CentOA', 'CentOs',...,'CentO9'}
'200[24]' genera las cadenas {'2002', '2004' }
'199[0-9]' genera las cadenas {'1990', '1991', ..., '1999' }
'Deb|Ubu' genera las cadenas {'Deb','Ubu'}
'BA*' genera las cadenas {'B','BA','BAA','BAA',...}
'BA+' genera las cadenas {'BA','BAA','BAAA',...}
'(BA){3}' genera la cadena 'BABABA'
'Nov(\.|iembre|ember)?'
    genera las cadenas {Nov., Noviembre, November, Nov}