Expresiones regulares y derivadas

Teoría de Lenguajes

1^{er} cuatrimestre de 2020

Plan de la clase

1. Expresiones regulares
   - Definiciones y propiedades
   - Ejemplos
   - Otras propiedades
   - Extensiones
2. Conversión de autómata finito a expresión regular
   - Definiciones y propiedades
   - Ejemplo
   - Conversión
3. Conversión de expresión regular a autómata finito
   - Derivada de un lenguaje
   - Derivada de una expresión regular
   - Ejemplos
   - Conversión

Expresiones regulares

Las expresiones regulares son expresiones que se utilizan para *denotar* o *definir* lenguajes regulares. No sirven para denotar lenguajes menos restrictivos que los regulares (libres de contexto y demás).
Ejemplo: *.txt
Las usan grep, awk, sed, perl, muchos editores de texto.
Se definen a partir de conjuntos simples y los que se pueden obtener de ellos a partir de las operaciones de union, concatenación y clausura.
Cada expresión regular e va a denotar un lenguaje regular $\mathrm{L}(\mathrm{e})$.

Expresiones regulares

Se definen recursivamente de la siguiente manera:

1. La expresión regular $\emptyset$ define el lenguaje vacío.
2. La expresión regular λ define el lenguaje $\{\lambda\}$.
3. La expresión regular a define el lenguaje $\{\mathrm{a}\}$.
4. Sean:

   v la expresión regular que define el lenguaje $\mathrm{L_v}$
   w la expresión regular que define el lenguaje $\mathrm{L_w}$

   entonces:

   1. la expresión regular $(\mathrm{v.w})$ define el lenguaje $\mathrm{L_v.L_w}$
   2. la expresión regular $(\mathrm{v|w})$ define el lenguaje $\mathrm{L_v} \cup \mathrm{L_w}$
   3. la expresión regular v^* define el lenguaje $\mathrm{L_v^*}$
   4. la expresión regular v^+ define el lenguaje $\mathrm{L_v^+}$

Precedencia: 1) * , 2) . , 3) |
El operador . se puede omitir.

Ejemplos

- La expresión regular $(\mathrm{a}|\mathrm{b})^+$ define el lenguaje de todas las cadenas de as y bs de longitud mayor o igual que 1.
- La expresión regular $(1|\ldots|9)(0|1|\ldots|9)^*|0$ define todos los números naturales sin ceros no significativos.
- Cadenas de 0s y 1s terminadas en 01:

$$(((0|1)^*0)1) = (0|1)^*01$$

- Cadenas que tienen como subcadena a 000:

$$(((((0|1)^*0)0)0)(0|1)^*) = (0|1)^*000(0|1)^*$$

Semántica

Definición

- $\mathrm{L(u.v)} = \mathrm{L(u).L(v)}$
- $\mathrm{L(u|v)} = \mathrm{L(u)} \cup \mathrm{L(v)}$
- $\mathrm{L(u^*)} = (\mathrm{L(u)})^*$

Definición

Dos expresiones regulares e y e' se dicen equivalentes, si $\mathrm{L(e)} = \mathrm{L(e')}$. Se suele escribir $\mathrm{e} = \mathrm{e}'$.

Propiedades

La unión y la concatenación son asociativas:

Asociatividad

$$(u|v)|w = u|(v|w) = u|v|w$$

$$(u.v).w = u.(v.w) = uvw$$

Ejemplos

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

Dar una expresión regular que defina:

1. Cadenas que terminan en 01: $(0|1)^*01$
2. Cadenas con 0s y 1s alternados: $(0|\lambda)(10)^*(1|\lambda)$
3. Complemento del primero (cadenas que NO terminan en 01): $((0|1)^*(0|11))|1|\lambda$
4. Cadenas con cantidad par de ceros: $1^*(01^*01^*)^*$ o equivalentemente: $(1|01^*0)^*$

Aclaraciones

- Por lo general no vamos a pedir que se demuestre que la ER denota el lenguaje pedido, pero sí tiene que haber una justificación de la misma.
- Como dijimos, el poder expresivo de las ER y los AF son el mismo. Pero hay veces que un formalismo es más intuitivo para definir uno que el otro. Por ejemplo, el ejemplo 4 es muy fácil de aceptar con AF.
- *Recordar que aceptar un lenguaje* L *(lo mismo, denotar un lenguaje) significa exactamente* L*, no alguno que lo incluya.*

Ejercicios

1. Constantes reales con signo y notación exponencial.
2. Nombres de archivo con path (para DOS/Win, para Unix).
3. Direcciones de e-mail (sólo de Argentina o Uruguay).

Conmutatividad y distributividad

Conmutatividad de $|$

$$u|v = v|u$$

(notar que . no es conmutativo)

Distributividad de . respecto de $|$

$$u(v|w) = uv|uw$$

$$(u|v)w = uw|vw$$

Neutro y absorbente

Elemento neutro de .

$$u.\lambda = \lambda.u = u$$

Elemento absorbente de .

$$u.\emptyset = \emptyset.u = \emptyset$$

Elemento neutro de |

$$u|\emptyset = \emptyset|u = u$$

Otros operadores

Los operadores . | * y + se comportan de la forma que vimos.
Si u es una expresión regular, n y m naturales:

- $u? = (u|\lambda)$
- $u\{n\} = u^n$
- $u\{n,\} = u^n.u^*$
- $u\{n,m\} = u^n|u^{n+1}|...|u^m$
- $() = \lambda$
- . = cualquier carácter
- ^ = comienzo de línea
- $ = fin de línea
- \seguida de alguno de los caracteres ^.[$()—*+?{\ representa ese carácter literal

Conjuntos de caracteres

- $[\mathrm{abc}] = \mathrm{a}|\mathrm{b}|\mathrm{c}$
- $[\mathrm{a}-\mathrm{z}] = \mathrm{a}|...|\mathrm{z}$
- ^ al comienzo de la lista indica complemento [^abc] = cualquier caracter menos a, b o c.

Hay clases de caracteres predefinidas: (en posix) *alnum*, *alpha*, *digit*, *space*...

Conversión de autómata finito a expresión regular

Sea
$A = (Q = \{q_0, q_1, \ldots, q_n\}, \Sigma = \{a_1, a_2, \ldots, a_m\}, F \subseteq Q, q_0, \delta_A)$
un autómata finito determinístico.
δ_A es la función de transición y está definida en:

$$\delta_A : Q \times \Sigma \to Q$$

Esta función se puede extender para cadenas de cualquier longitud sobre Σ, definiendo la función δ:

$$\delta : Q \times \Sigma^* \to Q \text{ tal que } \begin{cases} \delta(q, ax) & = & \delta(\delta_A(q, a), x), \text{ para } x \in \Sigma^* \\ \delta(q, \lambda) & = & q \end{cases}$$

Generalización

El lenguaje aceptado por el autómata se define como:

$$\mathrm{L(A)} = \{\mathrm{x} \in \Sigma^* \mid \delta(\mathrm{q_0}, \mathrm{x}) \in \mathrm{F}\}$$

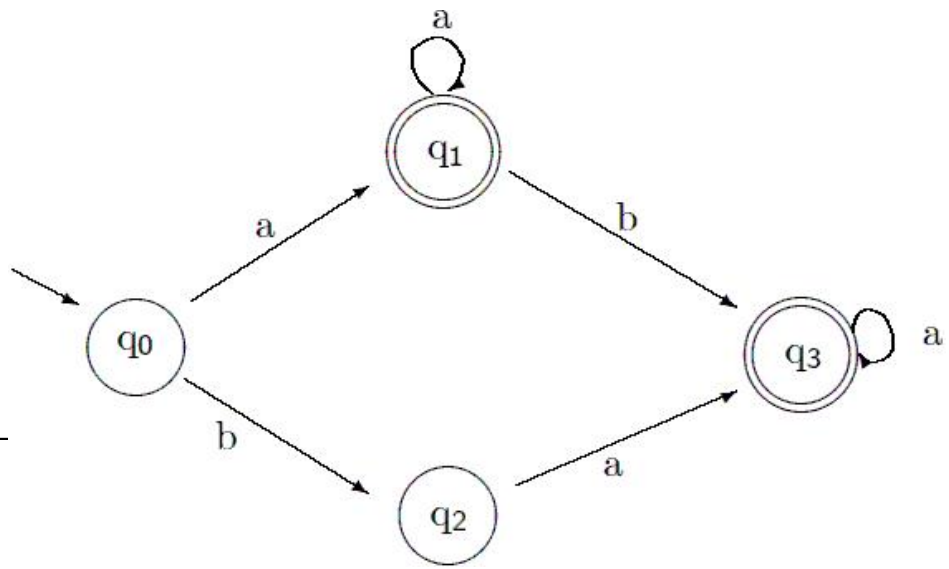
De la misma forma que definimos el lenguaje L como el conjunto de cadenas aceptadas por el autómata partiendo del estado $\mathrm{q_0}$, podemos definir un lenguaje $\mathrm{L_i}$ para un estado $\mathrm{q_i}$ dado del autómata, que corresponde al lenguaje aceptado por el autómata si partiéramos, no de $\mathrm{q_0}$ sino de ese estado $\mathrm{q_i}$:

$$\mathrm{L_i} = \{\mathrm{x} \in \Sigma^* \mid \delta(\mathrm{q_i}, \mathrm{x}) \in \mathrm{F}\}$$

Ejemplo

Dado el autómata $A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \{q_1, q_3\}, q_0, \delta_1)$

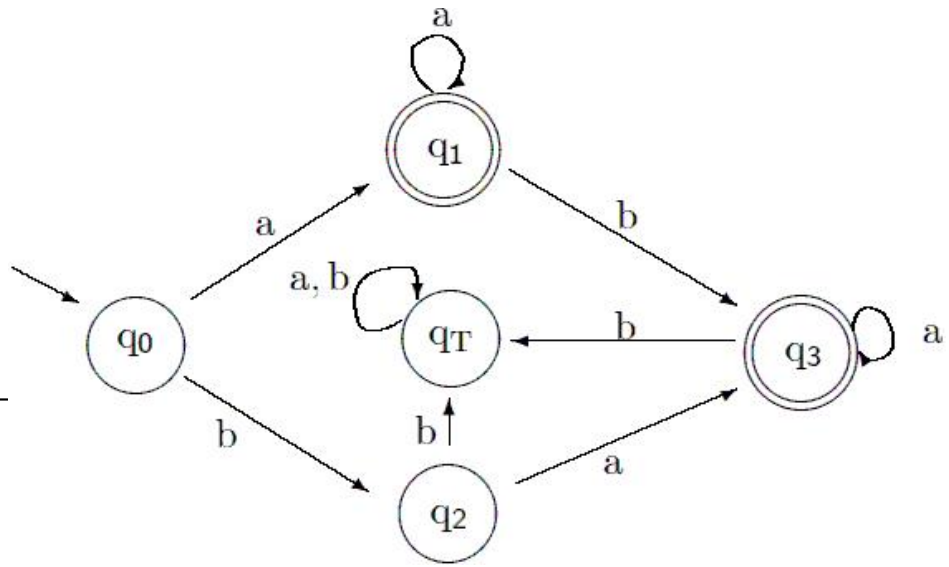
δ_1	a	b
0	1	2
1	1	3
2	3	
3	3	



Ejemplo completado con trampa

Dado el autómata $A = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \{q_1, q_3\}, q_0, \delta_1)$

δ_1	a	b
0	1	2
1	1	3
2	3	T
3	3	T
T	T	T



Autómata completado con el estado trampa. Trampa es un estado no final, adonde van las transiciones no definidas, y que cicla sobre el mismo con cada símbolo de Σ.

Observación

Partiendo del estado q_0, tendremos el lenguaje $\mathrm{L} = \mathrm{L}_0$ aceptado por el autómata. Ahora, si en vez de partir del estado q_0 partiéramos del estado q_1, tendríamos el lenguaje L_1, y así con los demás. En particular, el estado trampa define el lenguaje vacío, ya que partiendo de él no se puede reconocer ninguna cadena, i.e. $\mathrm{L}_\mathrm{T} = \emptyset$.

Ecuaciones

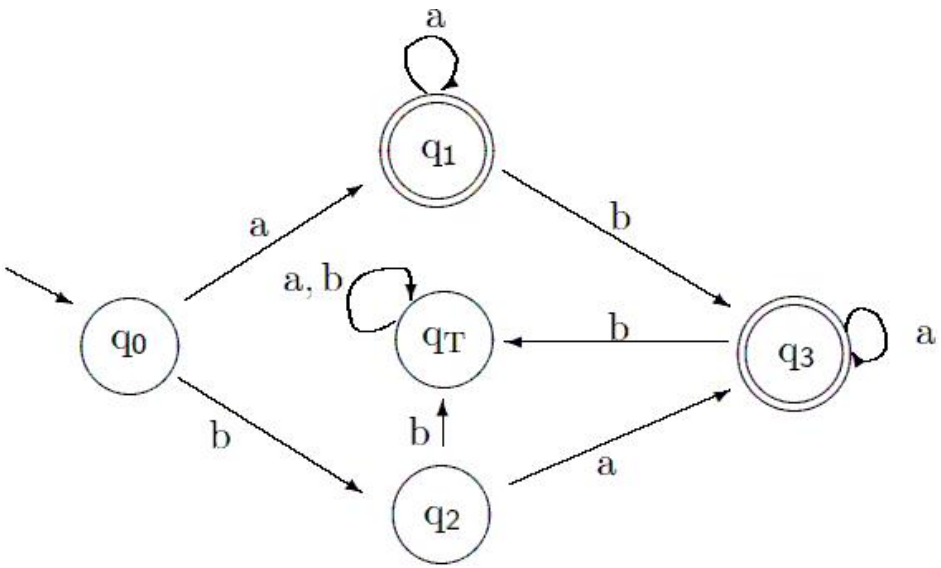
En el autómata definido antes, veamos que para cada estado q_i dado $(1 \leq i \leq n)$, podemos considerar las siguientes igualdades, donde $\Sigma = \{a_1, \ldots, a_m\}$:

$$\begin{aligned} \delta(q_i, a_1) &= p_{k_1} \\ \delta(q_i, a_2) &= p_{k_2} \\ &\ldots \\ \delta(q_i, a_m) &= p_{k_m} \end{aligned} \tag{1}$$

y a partir de ellas expresar el lenguaje L_i mediante la siguiente ecuación:

$$L_i = a_1.L_{k_1} | a_2.L_{k_2} | \ldots | a_m.L_{k_m} | \epsilon(L_i) \tag{2}$$

Ecuaciones



En el autómata del ejemplo, estas ecuaciones serán:

$$
\begin{aligned}
L_0 &= a.L_1|b.L_2 \\
L_1 &= a.L_1|b.L_3|\lambda \\
L_2 &= a.L_3|b.L_T \\
L_3 &= a.L_3|\lambda \\
L_T &= a.L_T|b.L_T|\emptyset
\end{aligned}
$$

Ahora, hay que resolverlas.

Ecuaciones entre lenguajes

Imaginen ejemplos como

$$\begin{array}{ll} \mathrm{L} = \mathrm{aL} & \mathrm{L} = \mathrm{La} \\ \mathrm{L} = \mathrm{La}|\mathrm{aL} & \mathrm{L} = \mathrm{a}|\mathrm{bL} \\ \mathrm{L} = \mathrm{LL} & \mathrm{L} = \mathrm{LaL} \\ \mathrm{L} = \mathrm{L}^* & \mathrm{L} = \mathrm{L} \\ \ldots & \end{array}$$

¿Cómo se resuelven? Es decir, dada una ecuación, ¿cómo encontramos un lenguaje L tal que al sustituirlo a ambos lados de la igualdad esta se cumpla? (Más allá de la solución trivial $\emptyset$ que sirve en muchos casos...) Es complicado dar un método general. Pero nos interesan particularmente las ecuaciones de la forma $\mathrm{L} = \alpha\mathrm{L}|\beta$ donde α, β son expresiones regulares sobre Σ.

Propiedad

Podemos resolver el sistema de ecuaciones anterior (asociado al autómata) usando la siguiente propiedad de los lenguajes definidos sobre Σ (junto a otras propiedades de e.r.):

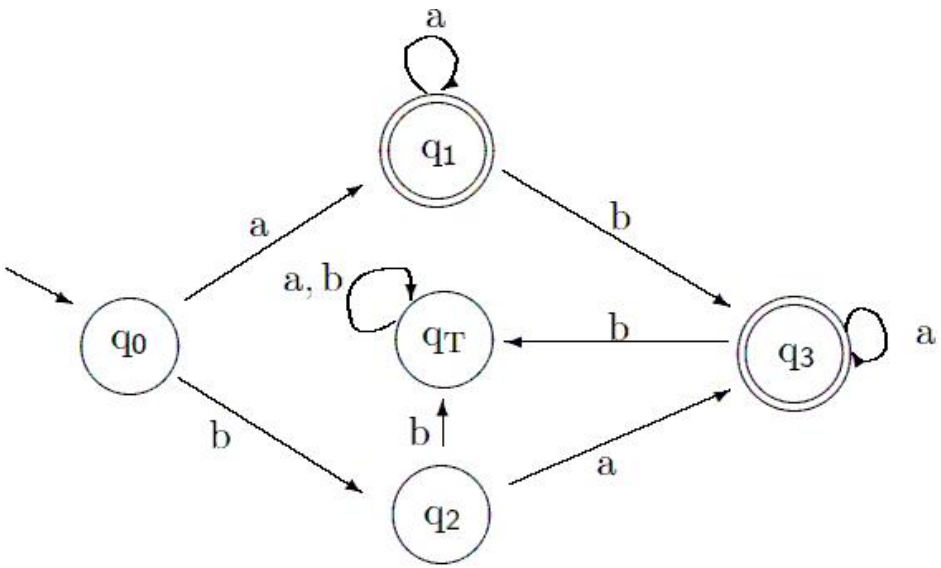
Proposición

Dados $\mathrm{R}, \alpha, \beta$ lenguajes sobre Σ,

$$\text{si } \mathrm{R} = \alpha.\mathrm{R}|\beta \wedge \lambda \notin \alpha, \text{ entonces } \mathrm{R} = \alpha^*.\beta$$

Si no se cumpliera la hipótesis $\lambda \notin \alpha$, podría haber más soluciones... incluso no regulares.

Conversión



$L_T = (a|b).L_T|\emptyset$, y como $\lambda \notin \{a, b\}$, entonces $L_T = (a|b)^*.\emptyset = \emptyset$
$L_3 = a.L_3|\lambda$, y como $\lambda \notin \{a\}$, entonces $L_3 = a^*.\lambda = a^*$
$L_2 = a.L_3|b.L_T = a.a^*|\emptyset = a^+$
$L_1 = a.L_1|b.L_3|\lambda = a.L_1|(b.a^*|\lambda)$, y como $\lambda \notin \{a\}$,
entonces $L_1 = a^*.(b.a^*|\lambda)$
$L_0 = a.L_1|b.L_2 = a^+(b.a^*|\lambda)|ba^+$

Derivada de un lenguaje

Dado un lenguaje L sobre un alfabeto Σ definimos

Derivada

1. $\partial\lambda(\mathrm{L}) = \mathrm{L}$
2. $\partial\mathrm{a}(\mathrm{L}) = \{\alpha : \mathrm{a}\alpha \in \mathrm{L}\}$ donde $\mathrm{a} \in \Sigma, \alpha \in \Sigma^*$
3. $\partial\mathrm{w}(\mathrm{L}) = \partial\mathrm{u}(\partial\mathrm{a}(\mathrm{L}))$ si $\mathrm{w} = \mathrm{a}.\mathrm{u}, \mathrm{a} \in \Sigma, \mathrm{u} \in \Sigma^*$

Notaremos $\partial\mathrm{ab}\ \mathrm{L} = \partial\mathrm{b}(\partial\mathrm{a}(\mathrm{L}))$

Ejemplos

Por ejemplo, si
$\mathrm{L} = \{\mathrm{a}^{3\mathrm{n}} : \mathrm{n} \in \mathbf{N}_{\geq 1}\} = \{\mathrm{aaa}, \mathrm{aaaaaa}, \mathrm{aaaaaaaaa}, \dots\}$, entonces:

$$\begin{aligned} \partial \mathrm{a}\ \mathrm{L} &= \{\mathrm{a}^{3\mathrm{n}-1} : \mathrm{n} \in \mathbf{N}_{\geq 1}\} = \{\mathrm{aa}, \mathrm{aaaaa}, \mathrm{aaaaaaaa}, \dots\} \\ \partial \mathrm{b}\ \mathrm{L} &= \emptyset \\ \partial \mathrm{aaa}\ \mathrm{L} &= \mathrm{L} \cup \{\lambda\} \\ \partial \mathrm{aaaa}\ \mathrm{L} &= \partial \mathrm{aL} \end{aligned}$$

Derivada de una expresión regular

Podemos definir la derivada $\partial w(u)$ de una expresión regular u con respecto a una cadena $w \in \Sigma^*$, mediante las siguientes reglas:

1. $\partial\lambda(u) = u$
2. Para $a \in \Sigma$:

Derivadas

$$\begin{aligned} \partial a(\emptyset) &= \emptyset \\ \partial a(\lambda) &= \emptyset \\ \partial a(b) &= \begin{cases} \lambda & \text{si} \quad a = b \\ \emptyset & \text{si} \quad a \neq b \end{cases} \end{aligned}$$

Derivada de una expresión regular

Si u y v son expresiones regulares, entonces:

Derivada

$$\begin{aligned} \partial \mathrm{a}(\mathrm{u}|\mathrm{v}) &= \partial \mathrm{a}(\mathrm{u})|\partial \mathrm{a}(\mathrm{v}) \\ \partial \mathrm{a}(\mathrm{u}.\mathrm{v}) &= \partial \mathrm{a}(\mathrm{u}).\mathrm{v}|\epsilon(\mathrm{u}).\partial \mathrm{a}(\mathrm{v}) \\ \partial \mathrm{a}(\mathrm{u}^*) &= \partial \mathrm{a}(\mathrm{u}).\mathrm{u}^* \end{aligned}$$

donde:

$\epsilon(\bullet)$

$$\epsilon(\mathrm{u}) = \begin{cases} \emptyset & \text{si} \quad \lambda \notin \mathrm{L_u} \\ \lambda & \text{si} \quad \lambda \in \mathrm{L_u} \end{cases}$$

Para $\mathrm{a} \in \Sigma \wedge \mathrm{w} \in \Sigma^*$:

$$\partial \mathrm{aw}(\mathrm{u}) = \partial \mathrm{w}(\partial \mathrm{a}(\mathrm{u}))$$

Ejemplos

$$\partial a(a) = \lambda$$
$$\partial a(a|b) = \partial a(a)|\partial a(b) = \lambda|\emptyset = \lambda$$
$$\partial a((a|b).c) = \partial a(a|b).c|\epsilon(a|b).\partial a(c) = \lambda.c|\emptyset.\emptyset = c$$
$$\partial a(a^*) = a^*$$
$$\partial a(a^+) = \partial a(a.a^*) = \partial a(a).a^*|\epsilon(a).\partial a(a^*) = \lambda.a^*|\emptyset.a^* = a^*$$
$$\partial a(ba^{20}) = \cdots = \emptyset$$

Conversión

Podemos formular un método para, dada una expresión regular que define un lenguaje, encontrar un autómata finito que acepta ese lenguaje.
Primero observemos que las ecuaciones (1) se pueden formular en términos de las derivadas del lenguaje L_i:

Ecuaciones

$$\begin{aligned} \partial a_1(L_i) &= L_{k_1} \\ \partial a_2(L_i) &= L_{k_2} \\ &\dots \\ \partial a_m(L_i) &= L_{k_m} \end{aligned}$$

Reemplazando estos valores en (2) nos queda:

$$L_i = a_1.\partial a_1(L_i)|a_2.\partial a_2(L_i)|\dots|a_m.\partial a_m(L_i)|\epsilon(L_i)$$

Observaciones

Si tenemos la expresión regular que define L, tenemos la expresión regular que define $\mathrm{L_0}$. Si derivamos esta expresión regular con respecto a un símbolo $\mathrm{a_j} \in \Sigma$, obtendremos la expresión regular que define el lenguaje aceptado por el autómata comenzando por el estado $\delta(\mathrm{q_0}, \mathrm{a_j})$ (puede ser el mismo $\mathrm{L_0}$ u otro lenguaje $\mathrm{L_k}$). Así iremos obteniendo una cantidad finita de expresiones regulares, que iremos adoptando como estados del autómata.

Conversión

Veamos cómo obtenemos el autómata que corresponde a la expresión regular $\mathrm{a}^+(\mathrm{b}.\mathrm{a}^*|\lambda)|\mathrm{ba}^+ = \mathrm{L}_0$:

$$
\begin{aligned}
\partial\mathrm{a}\ \mathrm{L}_0 &= \partial\mathrm{a}(\mathrm{a}^+(\mathrm{b}.\mathrm{a}^*|\lambda))|\partial\mathrm{a}(\mathrm{ba}^+) = \\
&\quad \mathrm{a}^*(\mathrm{b}.\mathrm{a}^*|\lambda)|\emptyset.\partial\mathrm{a}(\mathrm{ba}^*|\lambda)|\emptyset = \mathrm{a}^*(\mathrm{b}.\mathrm{a}^*|\lambda) = \mathrm{L}_1 \\
\partial\mathrm{b}\ \mathrm{L}_0 &= \partial\mathrm{b}(\mathrm{a}^+(\mathrm{b}.\mathrm{a}^*|\lambda)|\mathrm{ba}^+) = \\
&\quad \partial\mathrm{b}(\mathrm{a}^+).(\mathrm{b}.\mathrm{a}^*|\lambda)|\emptyset.\partial\mathrm{b}(\mathrm{b}.\mathrm{a}^*|\lambda)|\partial\mathrm{b}(\mathrm{ba}^+) = \mathrm{a}^+ = \mathrm{L}_2 \\
\partial\mathrm{a}\ \mathrm{L}_1 &= \partial\mathrm{a}(\mathrm{a}^*(\mathrm{b}.\mathrm{a}^*|\lambda)) = \partial\mathrm{a}(\mathrm{a}^*).(\mathrm{b}.\mathrm{a}^*|\lambda)|\lambda.\partial\mathrm{a}(\mathrm{b}.\mathrm{a}^*|\lambda) = \\
&\quad \mathrm{a}^*(\mathrm{b}.\mathrm{a}^*|\lambda)|\lambda.\emptyset = \mathrm{a}^*(\mathrm{b}.\mathrm{a}^*|\lambda) = \mathrm{L}_1 \\
\partial\mathrm{b}\ \mathrm{L}_1 &= \partial\mathrm{b}(\mathrm{a}^*(\mathrm{b}.\mathrm{a}^*|\lambda)) = \emptyset.(\mathrm{b}.\mathrm{a}^*|\lambda)|\lambda.\partial\mathrm{b}(\mathrm{b}.\mathrm{a}^*|\lambda) = \mathrm{a}^* = \mathrm{L}_3 \\
\partial\mathrm{a}\ \mathrm{L}_2 &= \partial\mathrm{a}(\mathrm{a}^+) = \mathrm{a}^* = \mathrm{L}_3 \\
\partial\mathrm{b}\ \mathrm{L}_2 &= \partial\mathrm{b}(\mathrm{a}^+) = \emptyset = \mathrm{L}_\mathrm{T} \\
\partial\mathrm{a}\ \mathrm{L}_3 &= \partial\mathrm{a}(\mathrm{a}^*) = \mathrm{a}^* = \mathrm{L}_3 \\
\partial\mathrm{b}\ \mathrm{L}_3 &= \partial\mathrm{b}(\mathrm{a}^*) = \emptyset = \mathrm{L}_\mathrm{T} \\
\partial\mathrm{a}\ \mathrm{L}_\mathrm{T} &= \partial\mathrm{a}\ \emptyset = \emptyset \\
\partial\mathrm{b}\ \mathrm{L}_\mathrm{T} &= \partial\mathrm{b}\ \emptyset = \emptyset
\end{aligned}
$$

Conversión

$$
\begin{array}{lcccl}
\partial a\ L_0 & = & a^*(b.a^*|\lambda) & = & L_1 \\
\partial b\ L_0 & = & a^+ & = & L_2 \\
\partial a\ L_1 & = & a^*(b.a^*|\lambda) & = & L_1 \\
\partial b\ L_1 & = & a^* & = & L_3 \\
\partial a\ L_2 & = & a^* & = & L_3 \\
\partial b\ L_2 & = & \emptyset & = & L_T \\
\partial a\ L_3 & = & a^* & = & L_3 \\
\partial b\ L_3 & = & \emptyset & = & L_T \\
\partial a\ L_T & = & \emptyset & = & L_T \\
\partial b\ L_T & = & \emptyset & = & L_T
\end{array}
$$

Conclusión

Con las derivadas calculadas podemos construir el autómata correspondiente. Los estados serán las expresiones regulares $\mathrm{L_i}$ obtenidas, y $\delta(\mathrm{L_i}, \mathrm{a}) = \mathrm{L_j}$ cada vez que $\partial \mathrm{a}\ \mathrm{L_i} = \mathrm{L_j}$. Serán estados finales aquellos $\mathrm{L_i}$ tales que $\lambda \in \mathrm{L_i}$.
El estado trampa $\mathrm{L_T}$ siempre corresponde a la expresión regular $\emptyset$.
En el caso del ejemplo, son finales $\mathrm{L_1}$ y $\mathrm{L_3}$.