

Expresiones regulares

Conceptos básicos

- En aritmética, podemos usar las operaciones $+$ y $\times$ para construir expresiones tales como $(5 + 3) \times 4$. similarmente, podemos usar las operaciones regulares para construir expresiones llamadas **expresiones regulares**, las cuales describen lenguajes, un ejemplo es: $(0 \cup 1)0^*$.
- El valor de la expresión aritmética es un número, mientras el valor de una expresión regular es un lenguaje.
- El valor del lenguaje para la expresión $(0 \cup 1)0^*$, consiste en todas las palabras que empiezan con 0 o un 1 seguido por algún numero de ceros.
- Obtenemos este resultado partiendo a la expresión en sus partes:
Primero, el símbolo 0 y el 1 son tomados de los conjuntos $\{0\}$ y $\{1\}$. Entonces $(0 \cup 1)$ significa $(\{0\} \cup \{1\})$, el valor de esta parte es el lenguaje $\{0, 1\}$. La parte 0^* significa $\{0\}^*$, y este valor es lenguaje consistente de todas las palabras que contienen algún numero de 0's.
Segundo, tal como el símbolo de la multiplicación en álgebra, el símbolo de concatenación $\circ$ a menudo esta implícito en las expresiones regulares. Por tanto $(0 \cup 1)0^*$ es el camino corto de $(0 \cup 1) \circ 0^*$. La concatenación pega las palabras de las dos partes para obtener el valor de la expresión entera.
- Las expresiones regulares tienen un rol importante en las aplicaciones de las ciencias de la computación.
- En aplicaciones que involucran texto, los usuarios pueden necesitar buscar las palabras que satisfagan ciertos patrones, las expresiones regulares suministran un poderoso método para describir tales patrones.
- Otro ejemplo de expresión regular es $(0 \cup 1)^*$ que empieza con el lenguaje $(0 \cup 1)$ y aplica la operación $*$: El valor de esta expresión es el lenguaje consistente de todas las posibles palabras de 0's y 1's.
- Si $\Sigma = \{0, 1\}$, podemos escribir Σ como el atajo para la expresión regular $(0 \cup 1)$.
- Mas generalmente, si Σ es algún alfabeto, la expresión regular Σ describe el lenguaje consistente de todas las palabras de longitud 1 sobre el alfabeto, y Σ^* describe el lenguaje consistente de todas las palabras sobre el alfabeto.
- Similarmente Σ^*1 es el lenguaje que contiene todos las palabras que terminan con un 1.
- El lenguaje $(0\Sigma^*) \cup (\Sigma^*1)$ consta de todas las palabras que empiezan con 0 o terminan con un 1.

Definición formal de una expresión regular

- Se dice que R es una expresión si R tiene:
 1. a para alguna a en el alfabeto Σ ,
 2. ε ,
 3. $\emptyset$,
 4. $(R_1 \cup R_2)$, donde R_1 y R_2 son expresiones regulares,
 5. $(R_1 \circ R_2)$, donde R_1 y R_2 son expresiones regulares, o
 6. (R_1^*) , donde R_1 es una expresión regular.
- El punto 1 y 2, las expresiones regulares a y ε representan a los lenguajes $\{a\}$ y $\{\varepsilon\}$, respectivamente. En el punto 3, la expresión regular $\emptyset$ representa el lenguaje vacío. En los puntos 4, 5 y 6 las expresiones representan el lenguaje obtenido por tomar la unión o concatenación de los lenguajes R_1 y R_2 , o la estrella del lenguaje R_1 , respectivamente.
- No hay que confundir las expresiones regulares a y ε . La expresión ε representa el lenguaje que contiene una simple palabra llamada, la palabra vacía mientras $\emptyset$ representa el lenguaje que no contiene ninguna palabra.
- Los paréntesis en una expresión pueden ser omitidos si la evaluación de la expresión es realizada en el orden de precedencia de las operaciones, que es: estrella (*), luego concatenación y luego la unión.
- R^+ es el atajo de RR^* , en otras palabras mientras que R^* tiene todas las palabras que tienen 0 o mas concatenaciones de las palabras de R , el lenguaje R^+ tiene todas las palabras que son 1 o mas concatenaciones de las palabras de R .
- Entonces $R^+ \cup \varepsilon = R^*$.
- Cuando queremos distinguir entre una expresión regular R y el lenguaje que lo describe, escribimos $L(R)$ que sea el lenguaje de R .
- **Ejemplos:** En los siguientes casos se asume que el alfabeto Σ es $\{0,1\}$.
 1. $0^*10^* = \{w \mid w \text{ contiene un sólo } 1\}$
 2. $\Sigma^*1\Sigma^* = \{w \mid w \text{ tiene al menos un } 1\}$
 3. $\Sigma^*001\Sigma^* = \{w \mid w \text{ contiene la palabra } 001 \text{ como una subpalabra}\}$
 4. $(01^+)^* = \{w \mid w \text{ cada } 0 \text{ en } w \text{ es seguido por al menos un } 1\}$
 5. $(\Sigma\Sigma)^* = \{w \mid w \text{ tiene una palabra de longitud par}\}$ (la longitud de una palabra es el número de símbolos que contiene)
 6. $(\Sigma\Sigma\Sigma)^* = \{w \mid \text{la longitud de } w \text{ es un múltiplo de tres}\}$
 7. $01 \cup 10 = \{01, 10\}$
 8. $0\Sigma^*0 \cup 1\Sigma^*1 \cup 0 \cup 1 = \{w \mid w \text{ empieza y termina con el mismo símbolo}\}$
 9. $(0 \cup \varepsilon)1^* = 01^* \cup 1^*$ La expresión $0 \cup \varepsilon$ describe el lenguaje $\{0, \varepsilon\}$, tal que la operación de concatenación suma 0 ó ε antes de cada palabra en 1^* .

10. $(0 \cup \varepsilon)(1 \cup \varepsilon) = \{\varepsilon, 0, 1, 01\}$
11. $1 * \emptyset = \emptyset$ Concatenando el conjunto vacío a algún conjunto produce el conjunto vacío.
12. $\emptyset^* = \{\varepsilon\}$ La operación estrella pone juntos cualquier numero de palabras de el lenguaje para obtener una palabra en el resultado. Si el lenguaje es vacío, la operación estrella puede poner juntas 0 palabras, dando sólo la palabra vacía.
- Si dejamos ser a R alguna expresión regular, tenemos las siguientes identidades.
 - $R \cup \emptyset = R$, Sumando el lenguaje vacío a algún otro lenguaje, no lo cambiara.
 - $R \circ \varepsilon = R$, uniendo la palabra vacía a alguna palabra, esta no cambiara.
 - Sin embargo, intercambiando $\emptyset$ y ε en las identidades precedentes puede causar que las igualdades fallen.
 - $R \cup \varepsilon$ puede no ser igual a R . por ejemplo, si $R = 0$, entonces $L(R) = \{0\}$ pero $L(R \cup \varepsilon) = \{0, \varepsilon\}$.
 - $R \circ \emptyset$ puede no ser igual que R . por ejemplo, si $R = 0$, entonces $L(R) = \{0\}$ pero $L(R \text{ concatenado } \emptyset) = \emptyset$.
 - Las expresiones regulares son herramientas útiles en el diseño de compiladores para lenguajes de programación. Objetos elementales en un lenguaje de programación son los llamados **tokens**, tales como los nombres de variables y constantes, pueden ser descritos con expresiones regulares.
 - Por ejemplo, una constante numérica que puede incluir una parte fraccionaria y/o signo puede ser descrita como un miembro del lenguaje
$$(+ \cup - \cup \varepsilon) (D^+ \cup D^+ . D^* \cup D^+ . D^*)$$
Donde $D = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ es el alfabeto de los dígitos decimales. Ejemplos de palabras generadas son: 72, 3.14259, +7., y -.01
 - Una vez que la sintaxis de los tokens del lenguaje de programación han sido descritos con expresiones regulares, sistemas automáticos pueden generar un **analizador léxico**, la parte del compilador que inicialmente procesa el programa de entrada.