Compiladores

Análisis Léxico

Tokens

Expresiones regulares

- Basado en las operaciones entre lenguajes, cómo definiría las operaciones necesarias para formar:
  - Un identificador

$$letra\_ (letra\_ \mid dígito)^*$$

  - Un entero
  - Un número con decimales

Expresiones regulares

- Una expresión regular r denota un Lenguaje L(r) con cierto alfabeto Σ
- Reglas
  - ϵ es una expresión regular y L (ϵ) es {ϵ}
    - es decir, un lenguaje con la cadena vacía como único elemento
  - Si a es elemento de Σ entonces a es una expresión regular L(a) = {a}

Expresiones regulares

- Entonces, sean r y s expresiones regulares de L(r) y L(s) respectivamente
  - (r) es una expresión regular que denota el lenguaje L(r)
  - (r)|(s) es una expresión regular que denota el lenguaje L(r)UL(s)
  - (r)(s) es una expresión regular que denota el lenguaje L(r)L(s)
  - (r)* es una expresión regular que denota el lenguaje (L(r))*

Expresiones regulares

- Observaciones
  - * tiene mayor prioridad y es asociativo por la izquierda
  - Concatenación tiene la segunda posición en prioridad y es asociativa por la izquierda
  - | tiene la precedencia más baja por lo tanto

$$(\mathbf{a})|((\mathbf{b})^*(\mathbf{c}))$$

$$\mathbf{a}|\mathbf{b}^*\mathbf{c}$$

- (a) (c)* se puede sustituir ac*

Ejercicio

- La expresión regular a|b denota {a,b} por tanto:
- (a|b)(a|b)

- a*

- (a|b)*

- a|a*b

Propiedades algebraicas

- Si un lenguaje puede definirse mediante una expresión regular se llama **conjunto regular**.
- Si dos expresiones regulares r y s denotan el mismo conjunto regular entonces son equivalentes r= s en (a|b) = (b|a)

Propiedades algebraicas

LEY	DESCRIPCIÓN
$r\|s = s\|r$	$\|$ es conmutativo
$r\|(s\|t) = (r\|s)\|t$	$\|$ es asociativo
$r(st) = (rs)t$	La concatenación es asociativa
$r(s\|t) = rs\|rt;\ (s\|t)r = sr\|tr$	La concatenación se distribuye sobre $\|$
$\epsilon r = r\epsilon = r$	ϵ es la identidad para la concatenación
$r^* = (r\|\epsilon)^*$	ϵ se garantiza en un cerradura
$r^{**} = r^*$	* es idempotente

Figura 3.7: Leyes algebraicas para las expresiones regulares

Definiciones regulares

Definiciones regulares

- Si Σ es un alfabeto de símbolos básicos se nombrará **definición regular** a la secuencia de definiciones de la forma:

$$\begin{aligned} d_1 &\rightarrow r_1 \\ d_2 &\rightarrow r_2 \\ &\cdots \\ d_n &\rightarrow r_n \end{aligned}$$

- Donde:
  - d_1 es un nuevo símbolo que no está en Σ y es diferente de cualquier d
  - r_1 es una expresión regular sobre el alfabeto Σ U $\{d_1, d_2, d_3 \ldots\}$

Definiciones regulares

- Ejemplos:

$$
\begin{aligned}
\mathit{digito} &\rightarrow \texttt{0} \mid \texttt{1} \mid \cdots \mid \texttt{9} \\
\mathit{digitos} &\rightarrow \mathit{digito}\ \mathit{digito}^* \\
\mathit{fraccionOpcional} &\rightarrow \texttt{.}\ \mathit{digitos} \mid \epsilon \\
\mathit{exponenteOpcional} &\rightarrow (\ \texttt{E}\ (\ \texttt{+} \mid \texttt{-} \mid \epsilon\ )\ \mathit{digitos}\ ) \mid \epsilon \\
\mathit{numero} &\rightarrow \mathit{digitos}\ \mathit{fraccionOpcional}\ \mathit{exponenteOpcional}
\end{aligned}
$$

Definiciones regulares

Extensiones:

- Una o más
  - r+ equivale a rr* = r*r
- Cero o una
  - r? Equivale a r|ϵ
- Clases de caracteres
  - a|b|c puede reemplazarse por [abc]
  - Por lo tanto a|b|c…|z puede representarse como [a-z]

$$
\begin{aligned}
\textit{letra\_} &\rightarrow \texttt{[A-Za-z\_]} \\
\textit{digito} &\rightarrow \texttt{[0-9]} \\
\textit{id} &\rightarrow \textit{letra\_}\ (\textit{letra} \mid \textit{digito})^*
\end{aligned}
$$

identificadores

$$
\begin{aligned}
\textit{digito} &\rightarrow \texttt{[0-9]} \\
\textit{digitos} &\rightarrow \textit{digito}^+ \\
\textit{numero} &\rightarrow \textit{digitos}\ (\texttt{.}\ \textit{digitos})?\ (\texttt{E}\ \texttt{[+-]}?\ \textit{digitos})?
\end{aligned}
$$

números

Ejercicios

- Describa los lenguajes que generan las siguientes **expresiones regulares**:
  - a(a|b)*a
  - ((ϵ|a)b*)*
  - (a|b)*a(a|b)(a|b)
  - a*ba*ba*ba*
  - (aa|bb)*((ab|ba)(aa|bb)*(ab|ba)(aa|bb)*)*

Ejercicios

- Escriba las **definiciones regulares** de:
  - Los “dígitos” en un número hexadecimal (elija mayúsculas o minúsculas para los “dígitos” mayores a 9) .
  - Todas las cadenas de dígitos de longitud 3

Ejercicios

! Ejercicio 3.3.12: SQL permite una forma rudimentaria de patrones, en los cuales dos caracteres tienen un significado especial: el guión bajo (_) representa a cualquier carácter y el signo de por ciento (%) representa a cualquier cadena de 0 o más caracteres. Además, el programador puede definir cualquier carácter, por decir *e*, para que sea el carácter de escape, de manera que si colocamos a *e* antes de *e* antes de _, % o cualquier *e*, obtenemos el carácter que va después de su significado literal. Muestre cómo expresar cualquier patrón de SQL como una expresión regular, dado que sabemos cuál es el carácter de escape.

Ejercicios

https://jarroba.com/busqueda-de-patrones-expresiones-regulares/

https://regexr.com/

Referencias

- Aho,
- Imágenes
  - Aho/Setti Text book