



DE CHIAPAS

```
$bar->setFormat('debug ');
$bar->setBarCharacter('<<comment>=</comment>');
$bar->setBarWidth(50);
$bar->start();

$prod_path = public_path('imgs/products/');

$products = Product::select('id', 'image')
    ->where('cat_id', 162)
    ->where('mirror', 0)
    ->get();

foreach ($products as $product) {
    $img_cuts = explode( delimiter: '_', $product->image);
    $img_cut = $img_cuts[0].'_'.$img_cuts[1];
    $min_file = glob( pattern: $prod_path . 'bad_min/' . $img_cut . '.*.jpg');
    $big_file = glob( pattern: $prod_path . 'bad_big/' . $img_cut . '.*.jpg');
    rename(current($min_file), newname: $prod_path . 'temp_min/' . $product->image);
    rename(current($big_file), newname: $prod_path . 'temp_big/' . $product->image);

    if(!file_exists( filename: public_path('imgs/products/') . 'min/' . $prod->image))
        DB::table('products')->where('id', $prod->id)->update(['public' => 0]);
    }else{
        DB::table('products')->where('id', $prod->id)->update(['public' => 1]);
    }
}
```

UNIVERSIDAD AUTÓNOMA DE CHIAPAS

INVESTIGACIÓN

(ACTIVIDAD 1)

COMPILADORES

Facultad de Contaduría y Administración Campus I

Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software



2024
15 AGOSTO

DOCENTE
D.S.C LUIS GUTIÉRREZ ALFARO

ALUMNA
A221677 - KARLA DÍAZ AGUILAR

DEFINIR EL CONCEPTO DE EXPRESIÓN REGULAR

I. Explicar los tipos de operadores de expresiones regulares.

- La **cerradura de Kleene** que se representa por el símbolo “*”. Indica que el elemento precedente puede aparecer ninguna, una o más veces vez en la cadena.
- La **cerradura positiva** que se representa por el símbolo “+”. Indica que el elemento precedente puede aparecer una o más veces seguidas en la cadena.
- El **calificador** “?”. Indica que el elemento precedente puede aparecer ninguna, o una vez en la cadena.
- La **concatenación** por el símbolo “.”.
- La **disyunción** por el símbolo “[]”. Separa las alternativas posibles, ejemplo, " (marrón|castaño)".
- Los **paréntesis** “()”. Pueden usarse para definir un grupo de caracteres sobre los que se aplicaran otros operadores.
- Los **corchetes** “[]”. Permiten determinar una lista de caracteres, de los cuales se escogerá SOLAMENTE uno, ejemplo, [0123].

EJEMPLO:

Expresión regular que busca palabras que tengan al menos 3 letras es `\b[A-Za-z]{3,}\b`

II. Explicar el proceso de conversión de DFA a expresiones regulares.

La idea básica es eliminar sistemáticamente los estados del DFA mientras se preserva el idioma que reconoce. Este método consta de los siguientes pasos:

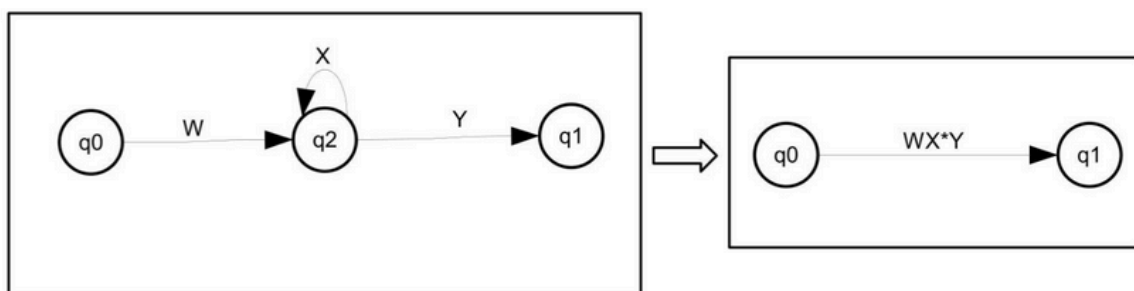
1. **Eliminar todos los estados de no aceptación:** Comience eliminando todos los estados de no aceptación del DFA. Dado que estos estados no contribuyen al lenguaje reconocido por DFA, podemos eliminarlos de manera segura sin afectar la expresión regular final.
2. **Introduzca un nuevo estado de inicio:** Cree un nuevo estado de inicio y agregue ϵ -transiciones desde este estado a cada uno de los estados de inicio originales. Esto garantiza que la expresión regular incluya todas las rutas posibles desde el nuevo estado de inicio hasta los estados de aceptación originales.
3. **Eliminar estados uno por uno:** Para cada estado restante en el DFA, lo eliminamos redirigiendo las transiciones entrantes y salientes a través de un nuevo estado. Este nuevo estado representa la expresión regular que describe las rutas entre los estados que se eliminan.
 - a. Redirigir las transiciones entrantes: para cada transición entrante al estado que se elimina, cree una nueva transición del estado de origen al estado de destino, etiquetada con la concatenación de la etiqueta de transición original y la expresión regular que representa el estado que se elimina.
 - b. Redirigir las transiciones salientes: para cada transición saliente del estado que se elimina, cree una nueva transición del estado de origen al estado de destino, etiquetada con la expresión regular que representa el estado que se elimina.
4. **Repita el paso 3 hasta que solo queden dos estados:** Continúe eliminando estados uno por uno hasta que solo queden dos estados en el DFA. Estos dos estados representan la expresión regular final que describe el lenguaje reconocido por el DFA original.

5. Combine los dos estados finales: combine los dos estados finales en un solo estado, etiquetado con la expresión regular que representa el idioma reconocido por el DFA original. Esta expresión regular es la forma equivalente del DFA original.

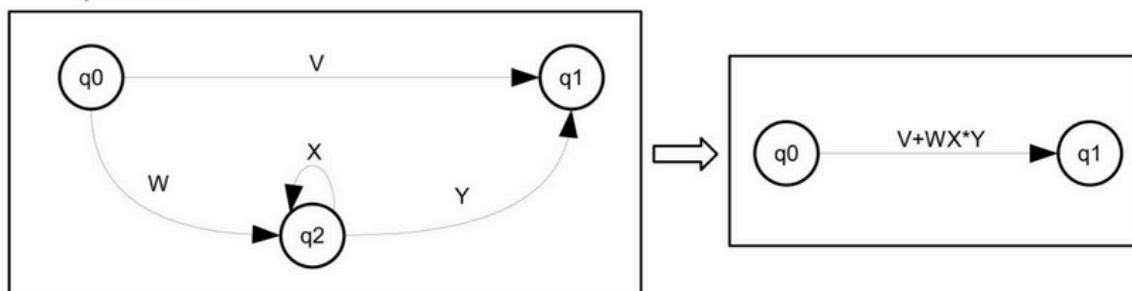
EJEMPLO:

Expresión regular que busca palabras que tengan al menos 3 letras es $\text{b}[A-Za-z]\{3,\}\text{b}$

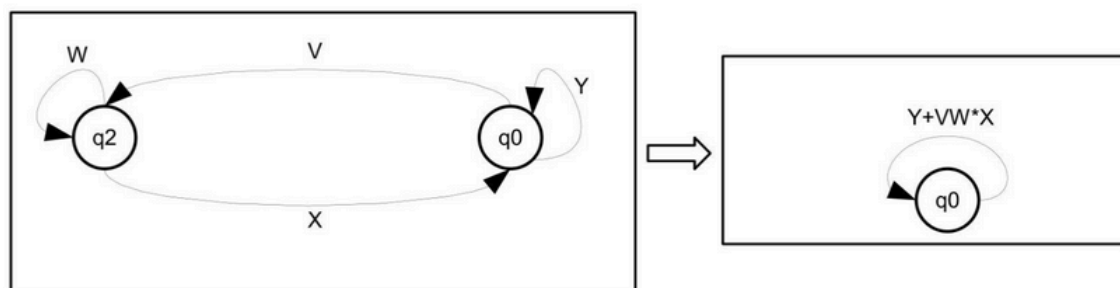
1) La concatenación:



2) La unión:



3) El retorno:



III. Explicar leyes algebraicas de expresiones regulares

Asociatividad y Conmutatividad

- Ley conmutativa para la union: $L + M = M + L$
- Ley asociativa para la union: $(L + M) + N = L + (M + N)$
- Ley asociativa para la concatenacion: $(LM)N = L(MN)$

NOTA: La concatenacion no es conmutativa, es decir $LM \neq ML$

Identidades y Aniquiladores

Una identidad para un operador es un valor tal que cuando el operador se aplica a la identidad y a algun otro valor, el resultado es el otro valor.

- 0 es la identidad para la adición: $0 + x = x + 0 = x$
- 1 es la identidad para la multiplicación: $1 \times x = x \times 1 = x$

Leyes Distributivas

Como la concatenación no es conmutativa, tenemos dos formas de la ley distributiva para la concatenación:

- Ley Distributiva Izquierda para la concatenación sobre unión: $L(M + N) = LM + LN$
- Ley Distributiva Derecha para la concatenación sobre unión: $(M + N)L = ML + NL$

Ley de Idempotencia

- Idempotencia de la Unión: $R+R=R$
- Cierre de Kleene: $R^* \cdot R^* = R^*$

(Aplicar el cierre de Kleene dos veces es equivalente a aplicarlo una sola vez)

Leyes de anulabilidad

- Concatenación con la cadena vacía: $\epsilon \cdot R = R \cdot \epsilon = R$
- Unión con la cadena vacía: $R + \emptyset = R$

Leyes de asociatividad y conmutatividad

- Asociatividad:

$$R + (S + T) = (R + S) + T$$

$$R \cdot (S \cdot T) = (R \cdot S) \cdot T$$

- Conmutación de la unión: $R + S = S + R$

EJEMPLO

- Simplificación usando Idempotencia:

$$a + a = aa + a = aa + a = a \text{ (usando la ley de idempotencia de la unión)}$$

- Asociatividad en Concatenación:

$$(a \cdot b) \cdot c = a \cdot (b \cdot c) \quad (a \cdot b) \cdot c = a \cdot (b \cdot c) \quad (a \cdot b) \cdot c = a \cdot (b \cdot c) \text{ (usando la ley de asociatividad en la concatenación)}$$

REFERENCIAS BIBLIOGRÁFICAS

Aho, A. V., Lam, M. S., Sethi, R & Ullman, J. D. (2008). *Compiladores: Principios, técnicas y herramientas*. (Segunda Edi.). (A. V. Romerio Elizondo, Trad.) México: Pearson Educación.

Academy, E. (2023, 2 agosto). ¿Cómo se puede convertir un autómata determinista de estado finito (DFA) en una expresión regular equivalente? - Academia EITCA. EITCA Academy. <https://es.eitca.org/cybersecurity/eitc-is-cctf-computational-complexity-theory-fundamentals/regular-languages/equivalence-of-regular-expressions-and-regular-languages/examination-review-equivalence-of-regular-expressions-and-regular-languages/how-can-a-deterministic-finite-state-automaton-dfa-be-converted-into-an-equivalent-regular-expression/>