



UNIVERSIDAD AUTÓNOMA DE CHIAPAS
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN C-I
LICENCIATURA EN INGENIERÍA EN DESARROLLO Y
TECNOLOGÍAS DE SOFTWARE

Actividad 1: Define los siguientes conceptos y realizar los ejercicios

Compiladores

Luis Gutierrez Alfaro

Agosto 15, 2024. Tuxtla Gutiérrez, Chiapas

Elaborado por:

A221655 Fernando Antonio Hernandez Zoma

Expresiones regulares

Es una cadena de caracteres utilizada en la detección de patrones en texto. (Benito Montoro, R., & Chen, X., 2021)

I.- Explicar los tipos de operadores de expresiones regulares.

IBM Corporation (2015) lo denota de la siguiente manera:

Barra inclinada invertida (\)

Inicia un escape de clase de carácter. Un escape de clase de carácter indica que el metacaracter siguiente debe utilizarse como carácter literal, en lugar de como un metacaracter.

Punto (.)

Coincide con cualquier carácter individual, excepto con el carácter de nueva línea (\n).

Signo de intercalación (^)

Si el carácter de intercalación aparece fuera de una clase de carácter, los caracteres que le siguen coinciden con el inicio de la serie de entrada o, para series de entrada de varias líneas, con el inicio de una línea. Una serie de entrada se considera como una serie de entrada de varias líneas si la función que utiliza la serie de entrada incluye el distintivo m.

Si el carácter de intercalación aparece como primer carácter en una clase de carácter, actúa como un signo de negación. Se produce una coincidencia si ninguno de los caracteres del grupo de caracteres aparece en la serie que se compara con la expresión regular.

Signo de dólar (\$)

Coincide con el final de la serie de entrada o, para series de entrada de varias líneas, con el final de una línea. Una serie de entrada se considera como una serie de entrada de varias líneas si la función que utiliza la serie de entrada incluye el distintivo m.

Signo de interrogación (?)

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular cero o una vez.

Asterisco (*)

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular de cero o más veces.

Signo más (+)

Coincide con el carácter o grupo de caracteres anteriores de la expresión regular una o más veces.

Barra vertical (|)

Coincide con el carácter (o grupo de caracteres) anterior o el carácter (o grupo de caracteres) siguiente.

Corchete de apertura ([) y corchete de cierre (])

Definen una clase de carácter. Por ejemplo:

- [aeiou] coincide con cualquier vocal.
- [a-z] significa cualquier letra minúscula.
- [a-p] significa cualquier letra minúscula de la a a la p.
- [0-9] significa cualquier dígito único.

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

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2000) lo explica de la siguiente manera:

Eliminación de Estados:

- Elimina un estado específico del DFA, digamos el estado sss.
- Para cada par de estados predecesor qiq_iqi y sucesor pjp_jpj del estado sss, introduce una nueva expresión regular (RE) que representa todas las posibles rutas desde qiq_iqi a pjp_jpj a través de sss. Esta RE es $Q_i S^* P_j Q_i S^* P_j$, donde:
 - $Q_i Q_i$ es la RE que representa las rutas desde qiq_iqi a sss,
 - $S^* S^* S^*$ representa las rutas dentro del estado sss (que puede incluir loops en sss),
 - $P_j P_j$ es la RE que representa las rutas desde sss a pjp_jpj.
- Agrega esta RE al arco entre qiq_iqi y pjp_jpj. Si no existe un arco entre estos dos estados, crea uno con la RE vacía \emptyset .

Repetir el Proceso:

- Repite el proceso de eliminación de estados para todos los estados del DFA, hasta que solo queden el estado inicial y los estados de aceptación.

Expresión Regular Final:

- Una vez que solo queden el estado inicial y los estados de aceptación, la RE final que describe el lenguaje del DFA será la unión de todas las rutas que van desde el estado inicial a los estados de aceptación.

Ejemplo Final:

- Para un autómata con dos estados, el proceso simplificado para obtener la RE puede ser expresado como $(R + SU^*T)^*SU^*(R + SU^*T)^*SU^*(R + SU^*T)^*SU^*$, donde:

- RRR es la RE para las rutas entre los estados,
- $SU^*SU^*SU^*$ representa las rutas dentro de un estado con loops,
- TTT es la RE para las rutas hacia otros estados.

III.- Explicar leyes algebraicas de expresiones regulares.

1. Identidades y Aniquiladores

- **Aniquilador:** Un valor que hace que, al combinarlo con otro valor mediante un operador, el resultado sea el aniquilador.
 - Para multiplicación: $\emptyset \times x = \emptyset$
 - No hay aniquilador para la suma.
- **Identidades:**
 - Para unión (+): $\emptyset + L = L$
 - Para concatenación (\cdot): $\epsilon L = L$

2. Leyes Distributivas

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

3. Ley de Idempotencia

- Un operador es **idempotente** si al aplicarlo a dos argumentos iguales el resultado es el mismo valor:
 - Para unión: $L + L = L$
 - Para multiplicación no siempre es idempotente.

4. Leyes de Cerradura

- **Idempotencia de la cerradura:** $(L^*)^* = L^*$
- **Cerradura de \emptyset :** $\emptyset^* = \epsilon$
- **Cerradura de ϵ :** $\epsilon^* = \epsilon$
- **Operaciones con L:**
 - $L^+ = LL^* = L^*L$
 - $L^* = \epsilon + L + LL + LLL + \dots$
 - $L^? = \epsilon + L$

Benito Montoro, R., & Chen, X. (2021). Comprobación de equivalencia entre especificaciones basadas en expresiones regulares.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2000). *Introduction to automata theory, languages, and computation* (2nd ed.). Addison-Wesley Publishing.