

Proyecto – BOT

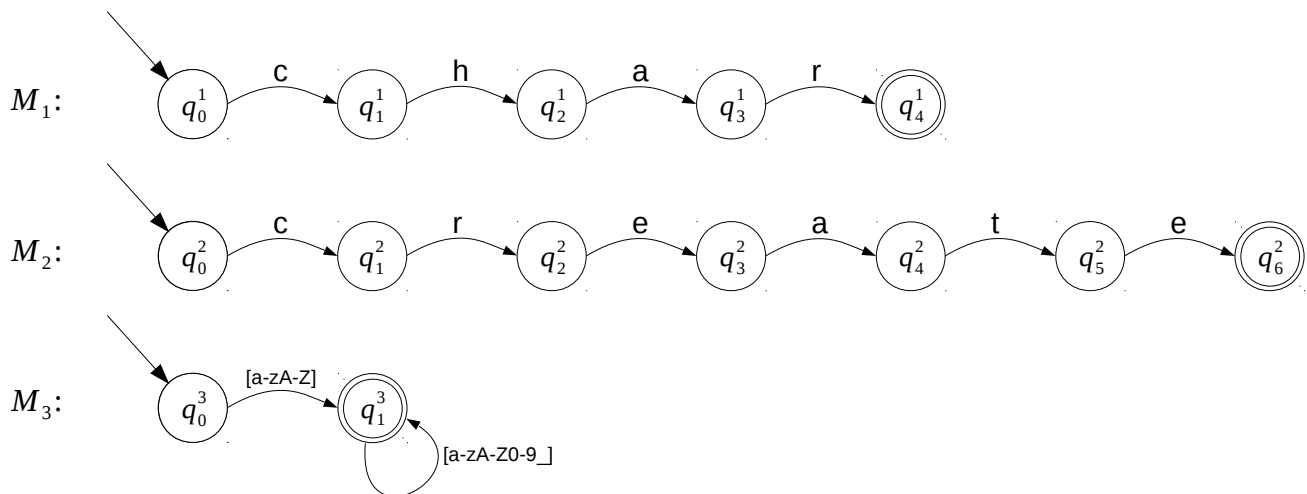
Revisión teórico-práctica

1. **Dé tres expresiones regulares E_1 , E_2 y E_3 que correspondan respectivamente al reconocimiento de la palabra clave *create*, de la palabra clave *char* y de identificadores:**

Dado un alfabeto Σ que está formado por todos los caracteres ASCII, podemos decir que $E_1 = \text{create}$, $E_2 = \text{char}$ y $E_3 = [a-zA-Z][a-zA-Z0-9_]*$ son expresiones regulares que reconocen la palabra clave *create*, de la palabra clave *char* y los identificadores de nuestro lenguaje, respectivamente.

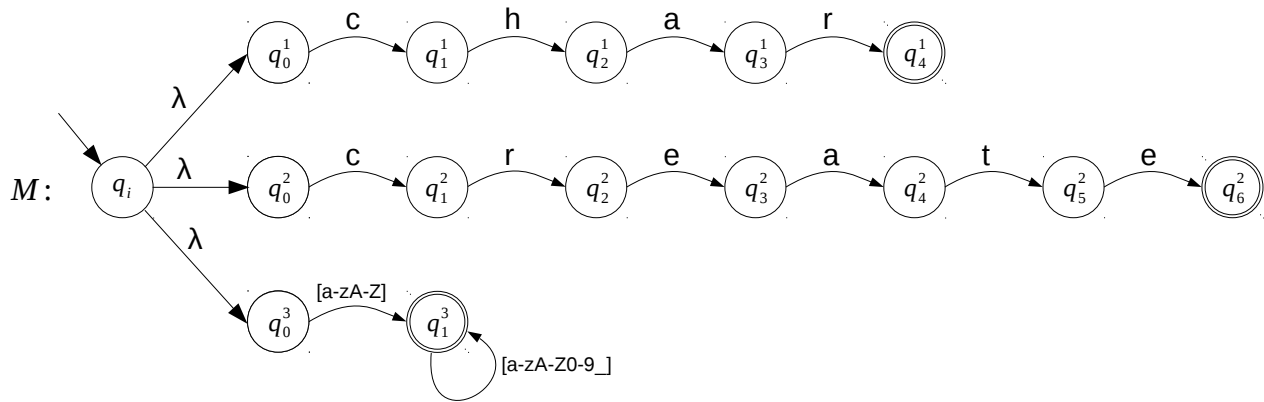
2. **Dé los diagramas de transición (i.e. representación gráfica) de los tres autómatas finitos (posiblemente no-determinísticos) M_1 , M_2 y M_3 que reconocen respectivamente a los lenguajes denotados por E_1 , E_2 y E_3 , esto corresponde al paso (1) del algoritmo A, con R'_i y R_i refiriéndose a nuestras E_i :**

Podemos representar gráficamente los autómatas finitos determinísticos M_1 , M_2 y al autómata finito no-determinístico M_3 que reconocen las expresiones E_1 , E_2 y E_3 mediante digrafos de la siguiente manera:



3. **Construya un autómata finito no-determinístico M que reconozca la unión de los lenguajes $L(M_1)$, $L(M_2)$ y $L(M_3)$ tal como se indica en el paso (2) de A.**

El siguiente autómata M es un autómata finito no-determinístico y reconoce la unión de los lenguajes $L(M_1)$, $L(M_2)$ y $L(M_3)$:



Donde se representa la frase vacía con el símbolo λ , que nos permite unir los tres autómatas obtenidos anteriormente en un único autómata que reconozca los tres lenguajes dados.

Nótese que el no-determinismo de M no está dado únicamente por el no-determinismo de M_3 , al unir M_1 y M_2 , ambos reconocen frases cuya primera letra empieza por "c".

4. **Note que, a efectos de implementar un analizador lexicográfico, es importante que el autómata M sepa reportar a cuál de los tres lenguajes pertenece cada palabra que él reconozca. Esto significa que M debe poder identificar a cuál de los tres lenguajes corresponde cada estado final. Indique a qué lenguaje corresponde cada uno de los estados finales de su autómata M :**

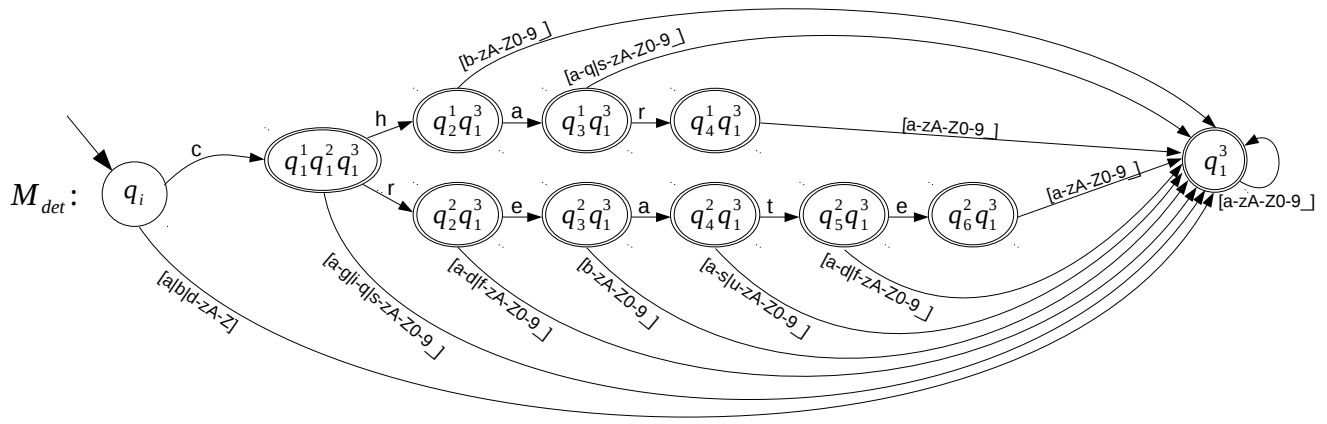
Los estados finales q_4^1 , q_6^2 y q_1^3 corresponden a los lenguajes $L(M_1)$, $L(M_2)$ y $L(M_3)$, respectivamente.

5. **La asignación de estados finales a lenguajes de su respuesta a la pregunta 4 debe crear conflictos, pues hay elementos que pertenecen a más de uno de los tres lenguajes $L(M_1)$, $L(M_2)$ y $L(M_3)$. Cada conflicto corresponde a una palabra w que pertenece a más de un lenguaje, digamos L_x y L_y . Indique cuáles son los conflictos de su autómata M , especificando las palabras que lo generan, y los lenguajes y estados finales involucrados:**

Podemos observar que la frase $w_1 = \text{char}$ pertenece tanto al lenguaje $L(M_1)$ como a $L(M_3)$. De manera similar, la frase $w_2 = \text{create}$ pertenece tanto al lenguaje $L(M_2)$ como a $L(M_3)$. Esto sucede debido a que $\text{sem}(E_1) \cup \text{sem}(E_2) \subset \text{sem}(E_3)$, luego $L(M_3)$ contendrá las mismas frases de $L(M_1)$ y $L(M_2)$.

6. **Construya un autómata finito determinístico M_{det} equivalente a M , i.e. que reconozca el mismo lenguaje M (viz. $L(M_1) \cup L(M_2) \cup L(M_3)$). Esto corresponde al paso (3) de A:**

Aplicando el paso (3) del algoritmo A, obtenemos un autómata finito determinístico equivalente a M , que llamaremos M_{det} , el cual reconoce el lenguaje $L = L(M_1) \cup L(M_2) \cup L(M_3)$.



7. ¿Cómo se reflejan los conflictos de su respuesta a la pregunta 5 en su autómata M_{det} ?

Básicamente, en el autómata M_{det} todos los estados son de aceptación excepto el estado q_i , dado que la frase vacía no es reconocida por la unión de los tres lenguajes dados. Los estados $q_4^1 q_1^3$ y $q_6^2 q_1^3$ reconocen las frases $w_1 = \text{char}$ y $w_2 = \text{create}$, respectivamente, por lo que el conflicto de saber a qué lenguaje pertenecen las frases w_1 y w_2 persiste en M_{det} , sabiendo que solamente pueden pertenecer a un lenguaje.

8. Los conflictos deben ser resueltos mediante un orden lineal que priorice a los lenguajes involucrados. En nuestro caso, establecemos que el orden de prioridad viene dado por la secuencia $\langle L(E_1), L(E_2), L(E_3) \rangle$ de manera decreciente, i.e. el primer lenguaje tiene más prioridad que los otros dos y el segundo más que el tercero. De acuerdo con esto, asocie un lenguaje (solo uno) a cada estado final de M_{det} tal como lo hizo en la pregunta 4 para M :

Dado el orden de prioridad anterior, podemos asociar los lenguajes $L(E_1)$ y $L(E_2)$ a los estados de aceptación $q_4^1 q_1^3$ y $q_6^2 q_1^3$, respectivamente, mientras que el resto de los estados de aceptación del autómata M_{det} les asociamos al lenguaje $L(E_3)$.

9. Tal como se indica en el paso (4) de A, construya un autómata finito determinístico mínimo M_{min} equivalente a M (y, por lo tanto, también equivalente a M_{det}). Explique por qué se debe usar la partición inicial de estados especificada en el paso (4) de A, e indique a cuál de los tres lenguajes corresponde cada estado final de M_{min} :

Para hallar el autómata M_{min} debemos tomar inicialmente una partición del conjunto de estados del autómata M_{det} , la cual será

$$\Pi_0 = \{ \{q_i\}, \{q_1^1 q_1^2 q_1^3, q_2^1 q_1^3, q_3^1 q_1^3, q_1^3, q_2^2 q_1^3, q_3^2 q_1^3, q_4^2 q_1^3, q_5^2 q_1^3\}, \{q_4^1 q_1^3\}, \{q_6^2 q_1^3\} \}$$

En lugar de tomar la partición inicial como el conjunto de estados finales y el conjunto de estados no finales, tomamos Π_0 dado que queremos saber a qué lenguaje $L(E_1)$, $L(E_2)$ o $L(E_3)$ pertenece cada frase que reconoce M_{det} , dado el orden de prioridad establecido anteriormente.

Aplicando clases de equivalencia obtenemos que

$$\Pi_1 = \{ \{q_i\}, \{q_1^1 q_1^2 q_1^3, q_2^1 q_1^3, q_1^3, q_2^2 q_1^3, q_3^2 q_1^3, q_4^2 q_1^3\}, \{q_4^1 q_1^3\}, \{q_6^2 q_1^3\}, \{q_3^1 q_1^3\}, \{q_5^2 q_1^3\} \}$$

$$\Pi_2 = \{ \{q_i\}, \{q_1^1 q_1^2 q_1^3, q_1^3, q_2^2 q_1^3, q_3^2 q_1^3\}, \{q_4^1 q_1^3\}, \{q_6^2 q_1^3\}, \{q_3^1 q_1^3\}, \{q_5^2 q_1^3\}, \{q_2^1 q_1^3\}, \{q_4^2 q_1^3\} \}$$

$$\Pi_3 = \{ \{q_i\}, \{q_1^3, q_2^2 q_1^3\}, \{q_4^1 q_1^3\}, \{q_6^2 q_1^3\}, \{q_3^1 q_1^3\}, \{q_5^2 q_1^3\}, \{q_2^1 q_1^3\}, \{q_4^2 q_1^3\}, \{q_1^1 q_1^2 q_1^3\}, \{q_3^2 q_1^3\} \}$$

$$\Pi_4 = \{ \{q_i\}, \{q_1^3\}, \{q_2^2 q_1^3\}, \{q_4^1 q_1^3\}, \{q_6^2 q_1^3\}, \{q_3^1 q_1^3\}, \{q_5^2 q_1^3\}, \{q_2^1 q_1^3\}, \{q_4^2 q_1^3\}, \{q_1^1 q_1^2 q_1^3\}, \{q_3^2 q_1^3\} \}$$

Finalmente, obtenemos que Π_4 es la partición de los singletones de los estados del autómata M_{det} , por lo tanto el autómata finito determinístico mínimo M_{min} es M_{det} .

10. ¿Cómo relaciona usted el desarrollo de las preguntas 1-9 a la implementación de su analizador lexicográfico construido con la herramienta escogida?:

El algoritmo A anteriormente utilizado, también conocido como Algoritmo de Hopcroft, nos permite obtener un autómata finito determinístico mínimo dada una expresión regular, generando a partir de esta última un autómata finito no-determinístico, luego convirtiéndolo en determinístico y finalmente minimizándolo.

Este algoritmo es el que implementa la herramienta Flex, para C y C++, que es básicamente un generador de analizadores lexicográficos, donde toma como entrada un texto escrito en los lenguajes anteriormente mencionados, y tokeniza dicho texto dadas ciertas expresiones regulares aplicando el Algoritmo de Hopcroft.