REPRESENTACION FINITA

Discutimos algunas propiedades de los lenguajes algunas de las propiedades que ya hemos discutido

Por ejemplo, las propiedades teóricas del conjunto habituales con respecto a la unión, intersección, el complemento, la diferencia, etcétera, se mantienen incluso en el contexto de algunas lenguas así que estamos interesados en algunas otras propiedades con respecto a la operación es recién instruidas

Autómata: Conjunto de estados + Control → Cambio de estados en respuesta a una entrada.

Tipo de Control:

• Determinístico: Para cada entrada, hay solo un estado al que el autómata puede ir desde el estado en que se encuentre.

• No determinístico: Un autómata finito es no-determinístico cuando se permite que el AF tenga 0 o más estados siguientes para cada par estado-entrada

Definición Formal de un Autómata

Un AF se representa como una 5-tupla: A = (Q, Σ, δ, q0, F). Donde:

1 Q: Un conjunto finito de estados.

2 Σ: Un conjunto finito de símbolos de entrada (alfabeto)

3 q0: El estado inicial/de comienzo.

4 F: cero o más estados finales/de aceptación.

5 δ: Función de transición. Esta función:

- Toma un estado y un símbolo de entrada como argumentos.
- Regresa un estado.
- Una “regla” de δ se escribe como δ (q, a) = p, donde q y p son estados y a es un símbolo de entrada.
- Intuitivamente: Si el AF está en un estado ´ q, y recibe una entrada a, entonces el AF va al estado p (nota: puede ser al mismo estado q = p).

Ejemplo:

$(L_1 U\ L_2)L_3 = L_1 L_3 U L_2 L_3$

$X€(L_1 U L_2)^{L_3}$

$X = X_1 X_2 \qquad X_1€L_1 U L_2 \qquad X_2€L_3$

$X = X_1 X_2 \qquad X_1€L_1 or\ X_1€L_2 \qquad X_2€L_3$

$X = X_1 X_2 \qquad X_1€L_1\ and\ X_2€L_3\ or\ X_1€L_2\ and\ X_2€L_3$

$X€L_1 L_3\ or\ X€L_2 L_3$

$X€L_1 L_3\ U\ X€L_2 L_3$

$X€L_1 L_3\ U\ X€L_2 L_3$

Lenguajes de un NFA

-NFA es una quíntupla (Q, Σ, δ, q0, F), donde δ es una función de Q × Σ ∪ {} al conjunto potencia de Q.

• Hay que evitar que sea parte del alfabeto Σ

El lenguaje aceptado por un NFA, A, es: L(A) = {w|δˆ (q0, w) ∩ F 6= ∅}.

Equivalencia entre un DFA y un NFA Un NFA es normalmente más fácil de definir, aunque al mismo tiempo, para cualquier NFA N existe un DFA D tal que L(D) = L(N) y viceversa. Para esto se usa la construcción de subconjunto que ´ muestra un ejemplo de cómo un autómata se puede construir a partir de otro.

Construcción de Subconjunto

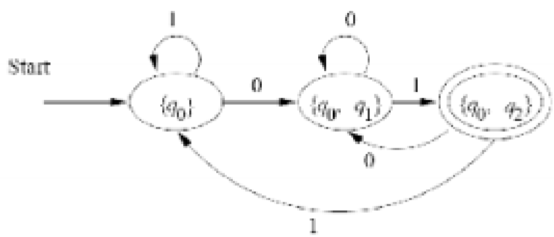
• Lo cual es un DFA (simplemente cambiando la notación). También es importante notar que no todos los estados pueden ser alcanzados. En particular, solo los estados B, E y F son accesibles, por lo que los demás los podemos eliminar.

• Una forma de no construir todos los subconjuntos para después encontrar que solo unos cuantos son accesibles, es construir la tabla solo para los estados accesibles (lazy evaluation).

EJEMPLO:

Para el ejemplo anterior: δD({q0}, 0) = {q0, q1}
δD({q0}, 1) = {q1}
δD({q0, q1}, 0) = {q0, q1}
δD({q0, q1}, 1) = {q0, q2} = δN(q0, 1) ∪ δN(q1, 1)
δD({q0, q2}, 0) = {q0, q1}
δD({q0, q2}, 1) = {q0}

Lo que nos queda:



Veremos cómo se pueden representar los lenguajes usando información infinita y qué significa la representación finita de las lenguas, hay muchos interesados en una representación finita de un idioma, si una persona es competente en un idioma en particular, no significa que produce todas las oraciones de los lenguajes. Básicamente lo que esperamos es que utilizando una cantidad finita de información queremos poder validar o construir cosas de diferencia de los idiomas, esto significa que al dar una cantidad finita de información todas las cadenas de los idiomas se enumeran o validad.

Por ejemplo, sirve el caso del compilador, el compilador puede validar cualquier programa que no sea más que una cadena del lenguaje de programación usando sólo unas instrucciones finalmente válidas