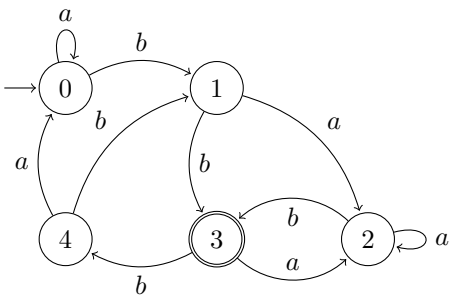
Pontificia Universidad Católica de Chile
Departamento de Ciencia de la Computación
IIC2223 – Teoría de Autómatas y Lenguajes Formales
Segundo semestre de 2024
Profesor: Cristian Riveros
Ayudante: Amaranta Salas

Ayudantia 6

Minimización de autómatas y Repaso I1

Problema 1

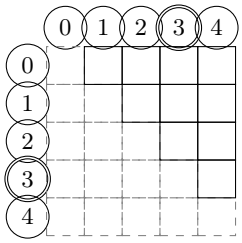
Minimize el siguiente autómata con el algoritmo visto en clases y dibuje el autómata mínimo:



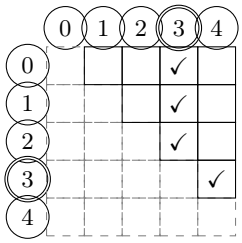
Solución

Usando el algoritmo de minimización visto en clases:

1. Construya una tabla con los pares $\{p, q\}$ inicialmente sin marcar.



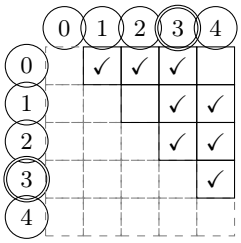
2. Marque $\{p, q\}$ si $p \in F$ y $q \notin F$ o viceversa.



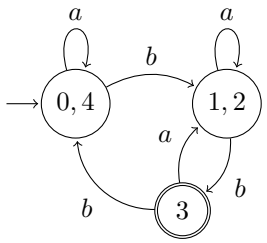
3. Repita este paso hasta que no hayan mas cambios:

- Si $\{p, q\}$ no están marcados y $\{\delta(p, a), \delta(q, a)\}$ estan marcados para algún $a \in \Sigma$, entonces marque $\{p, q\}$.

De manera directa: (0,1), (0,2), (1,4) y (2,4) si se lee una b llegan al mismo par de estados (1,3) que está marcado.



Los dos estados sin marcar llegan con ambos símbolos a un par de estados iguales y no se marcan. Finalmente, obtenemos el siguiente autómata:



Problema 2

Sea $\Sigma = \{0, 1\}$ y R una expresión regular sobre Σ^* dada por $R = \Sigma^* \cdot 0010 \cdot \Sigma^*$. Construya una expresión regular para el lenguaje $L = \Sigma^* \backslash \mathcal{L}(R)$. Explique su construcción

Solución

Armamos la expresión en tres pasos, lo que va antes de lo que se repite, lo que se repite, y con lo que puede finalizar la expresión. Recordar que queremos evitar el que aparezca 0010 en el string.

Paso 1: antes de la expresión

Primero vemos lo que se lea antes del patrón a repetir:

$$0^?1^*$$

Paso 2a: casos que empiezan con 0, pero no 00, y no contienen 0010

Ojo que tenemos cuidado con el caso de que repetir la expresión causa que se pueda tener un 0010, como sería el caso de $(010^*)^*$.

$$011^+0^*$$

Paso 2b: casos que empiezan con 00, pero no 001, y no contienen 0010

$$000^+11^+0^*$$

Paso 2c: casos que empiezan con 001, pero no 0010, y no contienen 0010

$$0011^+0^*$$

Paso 2d: casos con 0, y no 1

$$0^*$$

Paso 2e: casos que empiezen con 1

Por como estamos armando la expresión, recordamos que hay que evitar que se forme un 0010 al juntarlo con las otras expresiones del paso 2. Notar que el que vaya 0 después ya se ha visto en el paso 2, por lo que no necesitamos repetir eso

$$11^+0^*$$

Paso 3: Después de la expresión

Esta parte no se repite, ya que es el cómo termina el string. Vemos casos en los que el que antes se repitiese hiciece que no los podamos poner. Notar que no podemos poner 10 porque la expresión de los casos que se repiten $(2, 3, 4)$ hace que se pueda formar un 0010

$$0^*1^*$$

Paso 4: Juntar y simplificar expresiones

(simplificamos para que sean mas fáciles los pasos siguientes) Juntando todo, tenemos:

$$0^?1^*(011^+0^* + 000^+11^+0^* + 0011^+0^* + 0^* + 11^+0^*)^*(0^*1^*)$$

Notar que las expresiones de los pasos $2a, 2b, 2c$ tienen bastante en común así que las juntamos, para que quede mas corta la excritura:

$$0^?1^*((\varepsilon + 00^+ + 0)011^+0^* + 0^* + 11^+0^*)^*(0^*1^*)$$

Notamos que el paróentesis interior es 0^*:

$$0^?1^*(00^*11^+0^* + 0^* + 11^+0^*)^*(0^*1^*)$$

Reescribimos el 00^* como 0^+, y sacamos el paréntesis final que no aporta

$$0^?1^*(0^+11^+0^* + 0^* + 11^+0^*)^*0^*1^*$$

Juntamos los terminos con 11^+

$$0^?1^*(0^*11^+0^* + 0^*)^*0^*1^*$$

Notamos que el 0000^+ no aporta tanto, ya que ya se lee 0^* antes de un 1 en los pasos 2 y 3, por lo que podemos usar 0^* en la expresión:

$$0^?1^*(0^*11^+0^* + 0^*)^*0^*1^*$$

Podemos juntar los 0^* rebundantes:

$$0^?1^*(0^*(11^+)^?)^*0^*1^*$$

Finalmente el 0^* del paso 3 es rebundante, así que lo sacamos:

$$0^?1^*(0^*(11^+)^?)^*1^?$$

Adicional: Probamos casos que dada la construcción pueden hacer que falle la expresión

Se usa la intuición en esto, por ejemplo ya que dejamos los casos en el paso 2e y el paso 3 para el final, es bien posible que esos hayan quedado menos bien hecho, así que nos podemos enfocar en esos casos. Notar que con la construcción nos enfocamos con los casos que partan principalmente con 0^*, y eso restringió la construcción de las últimas expresiones, en particular las que son con $0^?10$

- 010, es aceptado
- 100, es aceptado
- 1010, no es aceptado, tendremos que ver ese caso, en particular los $(10)^+$. Notar que tenemos los casos en los que hay 00, y los que hay 11, así que tendremos que agregar este caso especial en donde se repite la expresión, evitando que se mezcle con los que contienen 00. Nos enfocaremos en los que sean de la forma $(1^+0)^*$, ya que el patrón no puede contener un 0010, entonces no puede tener un 00 y luego un $(10)^*$, por lo que no podemos hacer (10^+) con tanta facilidad
- 1010, no es aceptado, habrá que modificar la expresión

Paso 5: Agregamos casos borde

El problema principal es que tenemos puesto que pueden partir y terminar con $0*$, lo que nos limita en gran cantidad el agregar el caso $(10)^+$. Habrá que modificar la expresión considerando eso. Tenemos la expresión

$$0^?1^*(0^*(11^+)^?)^*1^?$$

Agregamos el caso de que se puede leer $(1^+0)^*$ al inicio de la expresión, considerando que puede haber una cantidad arbitraria de 1. Notar que cubrimos también el caso de que empieze con $(01^+)^*$:

$$(0^?1^+)^*(0^*(11^+)^?)^*1^?$$

Agregamos el $(10)^*$ a la parte que se repite, notamos los casos que pueden pasar

- $(0^+11^+)^*$, ya cubierto
- 00^+10, no válido porque contiene 0010
- $1(10)^*$, tenemos que agregar este caso
- (Casos mezclando (1^+0) con 0^* y 11^+ ya están cubiertos con la expresión, así que si se mezclan bien, podemos usarlos para no ver tantos casos)

Agregamos el caso a la parte que se repite. Vemos el caso del 11, ya que si hay 00, no se puede repetir el patrón:

$$(0^?1^+)^*(0^*(11^+(01^+)^*)^?)^*1^?$$

Notamos que con los casos que se repite, ya estamos considerando los que terminan con $0(1^+0)^*$

Problema 3

$$L = \{u_1\#u_2\#\ldots\#u_k \mid u_1, u_2, \ldots, u_k \in \{a,b\}^* \wedge k \geq 2 \wedge \forall i \neq j \in \{1,\ldots,k\}\ \ u_i \neq u_j\}$$

Solución

Podemos demostrarlo usando el lema de bombeo. Sea $w = \varepsilon\#a\#aa\#\ldots\#a^N$, es claro que $w \in L$ para todo $N > 0$ y que podemos tomar:

$$x = \varepsilon\#a\#aa\#\ldots\#a^{N-1}\#, y = a^N, z = \varepsilon$$

Luego, toda división de y tendrá la forma:

$$y = u \cdot v \cdot w = a^j a^k a^l, \text{ con } j + k + l = N$$

Y tomando $i = 0$ tendremos:

$$x \cdot u \cdot v^i \cdot w \cdot z = \varepsilon\#a\#aa\#\ldots\#a^{j+l} \notin L$$

Pues $j + l < N$, debido a la construcción de la palabra w, sabemos que existe una palabra u_i en x tal que sea igual a a^{j+l}, por lo que ya no se cumple que $\forall i \neq j \in \{1, \ldots, k\}\ \ u_i \neq u_j$.