>>> IF013 - Fundamentos Teóricos de Informática

>>> Licenciatura de Sistemas - UNPSJB - Sede Trelew

Name: Celia Cintas[†], Pablo Navarro[‡], Samuel Almonacid[§]

Date: August 2, 2017

[†]cintas@cenpat-conicet.gob.ar, cintas.celia@gmail.com, @RTFMCelia

[‡]pnavarro@cenpat-conicet.gob.ar, pablo1n7@gmail.com

[§]almonacid@cenpat-conicet.gob.ar, almonacid.samuel.tw@gmail.com

>>> Unidad 1

1. Autómatas finitos. Reconocedores. Traductores. Diagrama de estados. Autómatas finitos no deterministas.
2. Equivalencia entre autómatas finitos deterministas y no deterministas. Morfismos sobre autómatas. Autómata Cociente.
3. Propiedades de lenguajes aceptados por Autómatas Finitos. Expresiones y lenguajes regulares.
4. Propiedades algebraicas de los lenguajes regulares. Equivalencia entre autómatas finitos y lenguajes regulares.
5. Teorema de Kleene. Gramáticas regulares. Relación entre gramáticas regulares y autómatas finitos.
6. Usos y aplicaciones de los autómatas finitos y lenguajes regulares.

>>> Autómatas Finitos No Determinísticos

Hay dos formas posibles de entender cómo funciona un AFND.

* Cuando hay varias alternativas, el AFND elige alguna de ellas.
* Imaginarse que el AFND está en varios estados a la vez. Si luego de leer la cadena puede estar en un estado final, acepta la cadena.

En cualquier caso, es bueno por un rato no pensar en cómo implementar un AFND.

Definición

Un AFND es la 5-upla $M = (K, \Sigma, \delta, S, F)$, donde K, Σ, δ y F tienen el mismo significado que en AFD, pero $\delta : K \times \Sigma \to P(K)$.

Definición

La función de transición se puede generalizar para que acepte cadenas en Σ, es decir $\hat{\delta} : K \times \Sigma^* \to P(K)$.

$$\hat{\delta}(q, \lambda) = \{q\}$$
$$\hat{\delta}(q, xa) = \{p : \exists r \in \hat{\delta}(q, x) | p \in \delta(r, a)\} \text{ con } x \in \Sigma^* \text{ y } a \in \Sigma$$

Definición

Se dice que una cadena x es aceptada por un AFND $M = (K, \Sigma, \delta, S, F)$ si y solo si $\delta(S, x) \cap F \neq \emptyset$

Definición

Dado un AFND $M = (K, \Sigma, \delta, S, F)$, el lenguaje aceptado por M, el cual se denotará $L(M)$, es el conjunto de cadenas aceptadas por M y se define como:

$$L(M) = \{x : \delta(S, x) \cap F \neq \emptyset\}$$

Podemos extender la función de transición aún más, haciendo que mapee conjuntos de estados y cadenas en conjuntos de estados, es decir:

Definición

Función de transición $\delta : P(K) \times \Sigma^* \to P(K)$, dada por:

$$\delta(P, x) = \bigcup_{k \in P} \delta(k, x)$$

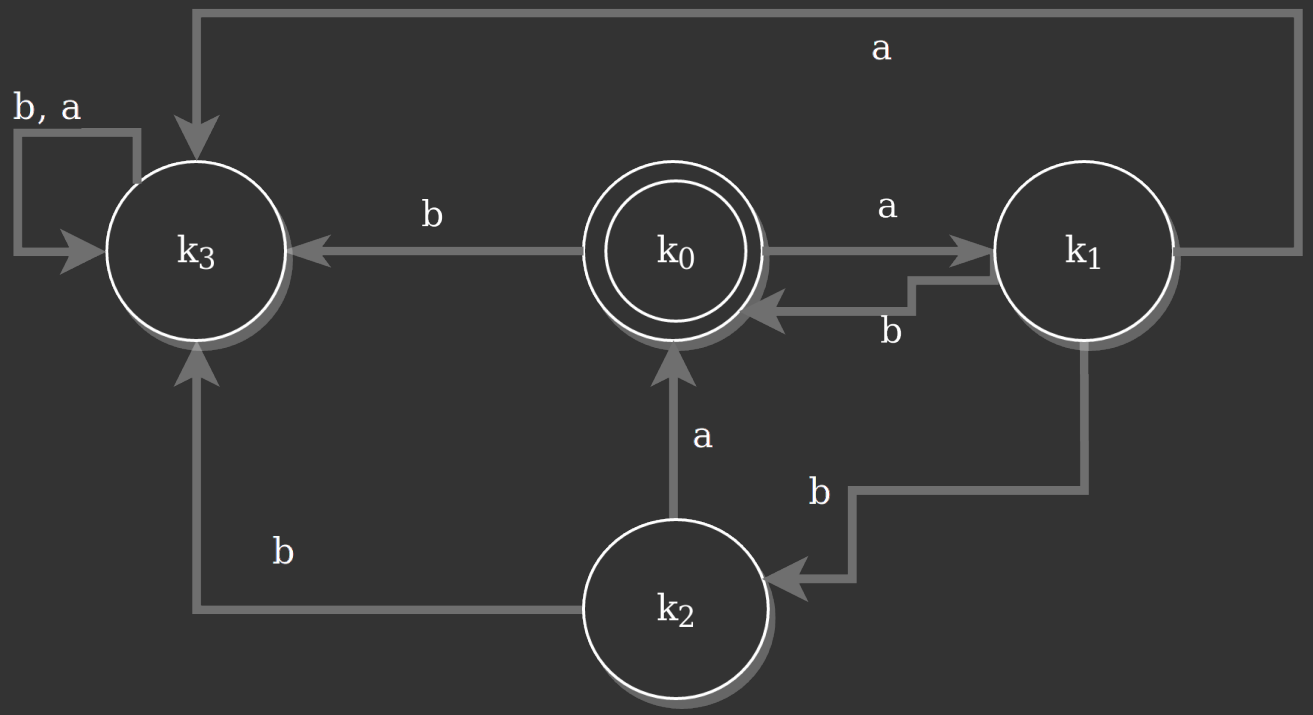
Para todo AFD existe un AFND y para cada AFND existe un AFD equivalente.

>>> Autómatas Finitos No Determinísticos (Cont.)

Ejemplo: AFND que acepte el lenguaje de las cadenas formadas por concatenación de 0 o más cadenas de ab o aba (sin importar el orden).

$$M = (K, \Sigma, \delta, S, F)$$
$$K = \{k_0, k_1, k_2, k_3\}$$
$$\Sigma = \{a, b\}$$
$$F = \{k_0\}$$
$$S = k_0$$

δ	a	b
k_0	$\{k_1\}$	$\{k_3\}$
k_1	$\{k_3\}$	$\{k_0, k_2\}$
k_2	$\{k_0\}$	$\{k_3\}$
k_3	$\{k_3\}$	$\{k_3\}$

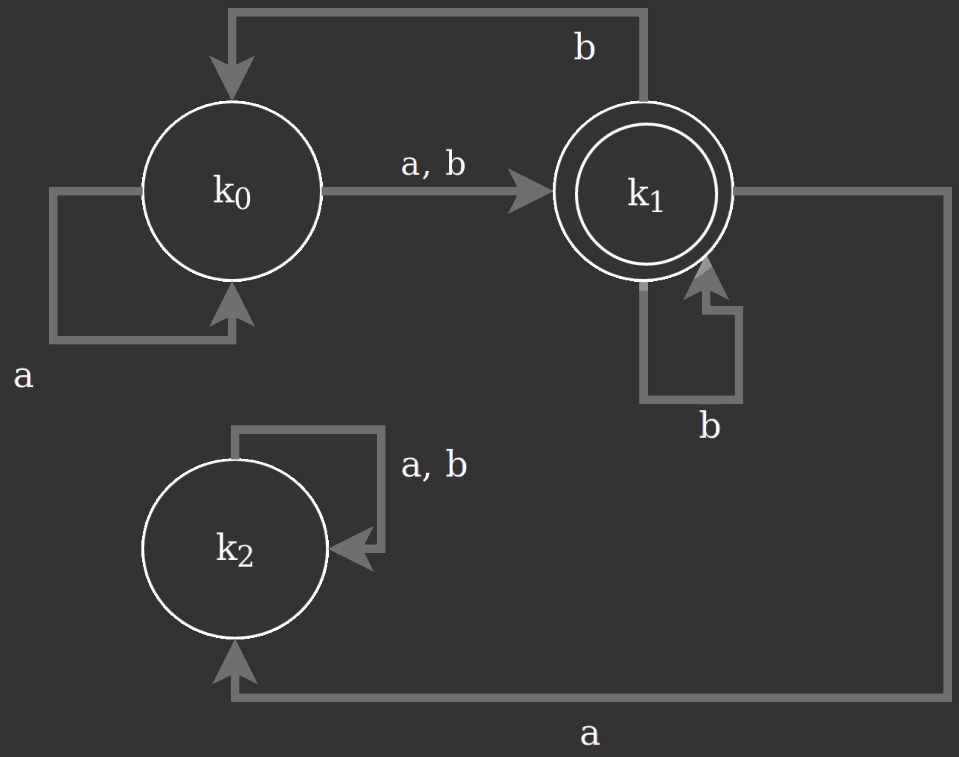


>>> Conversión de AFND a AFD

Sea un AFND $M = (K, \Sigma, \delta, S, F)$ donde:

$$K = \{k_0, k_1, k_2\}$$
$$\Sigma = \{a, b\}$$
$$F = \{k_1\}$$
$$S = k_0$$

δ	a	b
k_0	$\{k_0, k_1\}$	$\{k_1\}$
k_1	$\{k_2\}$	$\{k_0, k_1\}$
k_2	$\{k_2\}$	$\{k_2\}$



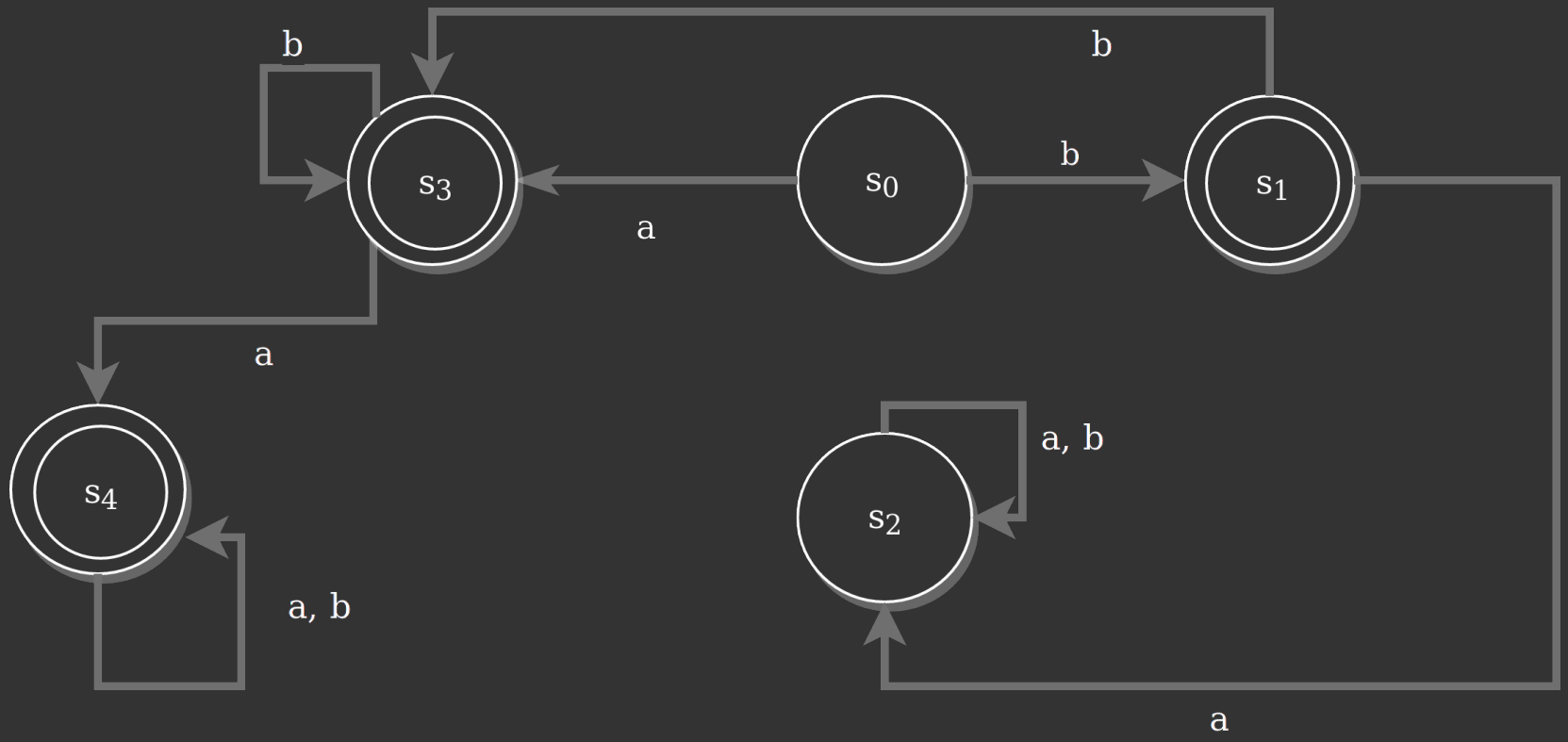
Obtengamos un AFD M' que reconozca el mismo lenguaje utilizando el teorema anterior.

Construyamos un AFD $M' = (K', \Sigma', \delta', S', F')$ donde:

$$K' = \{\{k_0\}, \{k_1\}, \{k_2\}, \{k_0, k_1\}, \{k_0, k_1, k_2\}\}$$
$$\Sigma' = \Sigma$$
$$F' = \{\{k_1\}, \{k_0, k_1\}, \{k_0, k_1, k_2\}\}$$
$$S' = \{k_0\}$$

δ	a	b	Aceptador
$s_0 = \{k_0\}$	$\{k_0, k_1\}$	$\{k_1\}$	0
$s_1 = \{k_1\}$	$\{k_2\}$	$\{k_0, k_1\}$	1
$s_2 = \{k_2\}$	$\{k_2\}$	$\{k_2\}$	0
$s_3 = \{k_0, k_1\}$	$\{k_0, k_1, k_2\}$	$\{k_0, k_1\}$	1
$s_4 = \{k_0, k_1, k_2\}$	$\{k_0, k_1, k_2\}$	$\{k_0, k_1, k_2\}$	1

>>> Conversión de AFND a AFD (cont.)



>>> Equivalencia entre AFD y AFND

Definición

Sea L un lenguaje aceptado por un autómata finito no determinista, entonces existe un autómata finito determinista que también acepta L. Agregar demostración

>>> Conversión de AFND a AFD (cont.)

lero

>>> Conversión de AFND a AFD (cont.)

lero

>>> Autómatas Finitos $AFND - \lambda$

lero

>>> Autómatas Finitos $AFND - \lambda$

lero

>>> Minimización

something here

something here

>>> Gramáticas

something here

>>> Gramáticas Regulares

something here

>>> Gramáticas Regulares y Autómatas Finitos

something here

>>> Gramáticas Regulares y Expresiones Regulares

something here

>>> Pumping Lemma para AFD

something here

>>> Gracias!