UNIDAD II: LENGUAJES REGULARES

6.1 Concepto de AFND

El funcionamiento de un AFND es muy similar al de un AFD. Sin embargo, mientras en un AFD sólo existe una posible acción en cada momento, en un AFND se puede elegir entre varias alternativas de transición.

6.1.1 Definición

Un *autómata finito no determinista* (AFND) es una quíntupla $A=(Q,\Sigma,f,q_0,F)$, donde:

- Q es un conjunto de estados
- Σ es el alfabeto de entrada
- $q_0 \in Q$ es el estado inicial
- $F \subseteq Q$ es el conjunto de estados finales
- $f: Q\times(\Sigma\cup\{\lambda\}) \rightarrow 2^Q$ es la función de transición

Ejemplo:

$AFND_1=(\{q_0,q_1,q_2\},\{0,1\},f, q_0,\{q_2\})$, con f:

$f(q_0,0)=\{q_0,q_1\}$	$f(q_1,0)=\varnothing$	$f(q_2,0)=\varnothing$
$f(q_0,1)=\{q_0\}$	$f(q_1,1)=\{q_2\}$	$f(q_2,1)=\varnothing$
$f(q_0,\lambda)=\varnothing$	$f(q_1,\lambda)=\{q_0,q_1\}$	$f(q_2,\lambda)=\varnothing$

AFND's pueden representarse mediante *tablas de transiciones*:

	$AFND_1$	0	1	λ
$\rightarrow$	q_0	$\{q_0,q_1\}$	$\{q_0\}$	
	q_1		$\{q_2\}$	$\{q_0,q_1\}$
*	q_2			

Notas:

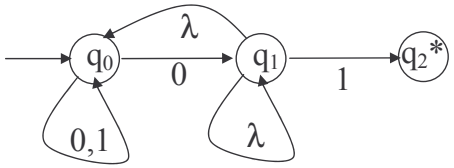
- El resultado de la función de transición puede ser el conjunto vacío (pares estado/símbolo terminal para las que el autómata no puede transitar). Se suelen omitir estas entradas en las tablas de transiciones: $f(q_0, \lambda)$, …
- El autómata puede transitar sin leer ningún símbolo de entrada: $f(q_1, \lambda)=\{q_0,q_1\}$
- El *no determinismo* del autómata proviene del hecho de que en cada momento (para cada símbolo de entrada y cada estado) pueden existir varias posibilidades de transición (o ninguna).

Definición (diagrama de transición de estados):
Sea un AFND $A=(Q,\Sigma,f,q_0,F)$. El *diagrama de transición de estados* de dicho autómata es el grafo que cumple las siguientes propiedades:

- El conjunto de nodos del grafo es Q
- El nodo correspondiente al estado inicial (y sólo él) está marcado con una flecha
- Los |F| nodos correspondientes a los estados finales (y sólo ellos) están marcados mediante "*" o mediante un doble círculo
- Existe un arco desde el nodo q_i a q_j etiquetado mediante el símbolo a si y sólo si $q_j \in f(q_i,a)$.

Ejemplo:

1. Grafo de transición para $AFND_1$ del ejemplo anterior:



6.1.2 Lenguaje reconocido por un AFND

Intuitivamente, un AFND acepta todas las palabras para las que **puede transitar** desde el estado inicial a un estado final.

Ejemplos:

1. El autómata $AFND_1$ del ejemplo anterior acepta las palabras $\{x \in \{0,1\}^* | x \text{ termina en } 01\}$.

2. ¿Qué palabras acepta el siguiente $AFND_2$?

	$AFND_2$	a	λ
→	q_0	$\{q_1,q_4\}$	
	q_1	$\{q_2\}$	
	q_2	$\{q_3\}$	
*	q_3		
	q_4	$\{q_5\}$	
*	q_5	$\{q_4\}$	

3. El siguiente autómata acepta la palabra a, ya que acepta aλ.

	$AFND_3$	a	b	λ
→*	p	{q}		
	q	{q,r,s}	{p,r}	{s}
	r		{p,s}	{r,s}
*	s			{r}

Definición (clausura respecto a λ):
La clausura de un estado p con respecto a λ, $CLAUS_\lambda(p)$, se define recursivamente de la siguiente manera:

1. $p \in CLAUS_\lambda(p)$
2. Si $q \in CLAUS_\lambda(p)$, y $r \in f(q,\lambda)$, entonces $r \in CLAUS_\lambda(p)$

Ejemplo:

1. $CLAUS_\lambda$ para los estados del $AFND_3$ (del ejemplo anterior):

AFND$_3$	$CLAUS_\lambda$
p	{p}
q	{q,s,r}
r	{r,s}
s	{r,s}

Nota: El conjunto $CLAUS_\lambda(p)$ se puede obtener mediante *árboles de transición*.

Definición (función de transición extendida):
La *función de transición extendida* $f^*: Q\times\Sigma^* \to 2^Q$, se define recursivamente de la siguiente manera:

- $f^*(p,\lambda)=CLAUS_\lambda(p)$, para todo $p \in Q$
- $f^*(p,ax)$, para todo $p \in Q$, $a\in\Sigma$ y $x\in\Sigma^*$, se construye de la siguiente forma:
  - Si $CLAUS_\lambda(p)=\{p_1,...,p_n\}$
  - y $\bigcup_{i=1}^{n} f(p_i,a)=\{r_1,...,r_m\}$
  - entonces $f^*(p,ax)=\bigcup_{i=1}^{m} f^*(r_i,x)$

Nota: Si $f(p,a)$ con $p\in Q$ y $a\in\Sigma\cup\{\lambda\}$ no está definido, se considera que $f(p,a)=\{\}=\varnothing$.

Ejemplos:

1. Considerando el $AFND_3$ de los ejemplos de antes, ¿qué valores tienen $f^*(p,b)$, $f^*(q,b)$, $f^*(p,a)$, $f^*(p,ba)$, $f^*(p,aab)$, $f^*(p,aa)$ y $f^*(r,\lambda)$?

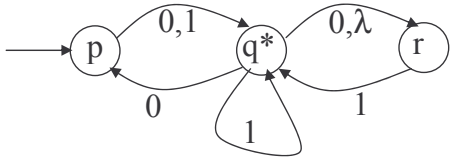
Definición (lenguaje reconocido):
Sea un AFND $A=(Q,\Sigma,f,q_0,F)$. El lenguaje *reconocido* por A está formado por el conjunto de palabras que pueden hacer transitar al autómata desde el estado q_0 haste un estado final:

$$L(A)=\{x \mid x\in\Sigma^* \text{ y } f^*(q_0,x)\cap F\neq\varnothing\}.$$

Definición (AFNDs equivalentes):
Dos AFNDs A_1 y A_2 son equivalentes ($A_1\equiv A_2$), si reconocen el mismo lenguaje: $L(A_1)=L(A_2)$.

Ejemplos:

1. ¿Qué lenguaje reconoce el $AFND_3$?
2. ¿Que lenguaje reconoce el siguiente AFND?



3. Define un AFND con no más de 5 estados que reconoce $L=\{abab^n \mid n\geq 0\}\cup\{aba^n \mid n\geq 0\}$
4. Define un AFND con 3 estados que reconoce L={ab, abc}*
5. ¿Es posible reconocer el lenguaje de 4. con menos estados?
6. Define un AFND que reconoce L={a}* y es tal que al borrar una transición en su grafo el lenguaje aceptado sería {a}.

6.2 Teoremas de equivalencia

6.2.1 Equivalencia entre AFND y AFD

Teorema 1:
Dado un AFND $A=(Q,\Sigma,f,q_0,F)$, existe un AFD A' tal que $L(A)=L(A')$.

Demostración:
Consideramos el AFD $A'=(2^Q,\Sigma,f',q_0',F')$, donde:

a) $q_0'=f^*(q_0,\lambda)=CLAUS_\lambda(q_0)$
b) $F'=\{c|\ c\in 2^Q \text{ y } c\cap F\neq\varnothing\}$
c) $f'(c_i,a)=\bigcup_{p\in c_i} f^*(p,a)$

Se puede observar que para todo $x\in\Sigma^*$: $x\in L(A)$ sii $x\in L(A')$:

$x\in L(A) \Leftrightarrow f^*(q_0,x)\cap F\neq\varnothing$ (lenguaje reconocido de AFNDs)
$\Leftrightarrow f^*(q_0,x)\in F'$ (por definición de F')
$\Leftrightarrow f'^*(q_0',x)\in F'$ (por definición de q_0' y f')
$\Leftrightarrow x\in L(A')$ (lenguaje reconocido de AFDs)

Por tanto, $A'\equiv A$.

Ejemplo: (autómata del ejemplo 2 anterior)

AFND:

	A	0	1	λ
→	p	{q}	{q}	
*	q	{p,r}	{q}	{r}
	r		{q}	

AFD equivalente:

	A'	0	1
	{}	{}	{}
→	{p}	{q,r}	{q,r}
*	{q}	{p,r}	{q,r}
	{r}	{}	{q,r}
*	{p,q}	{p,q,r}	{q,r}
	{p,r}	{q,r}	{q,r}
*	{q,r}	{p,r}	{q,r}
*	{p,q,r}	{p,q,r}	{q,r}

Notas:

- El AFD obtenido de la forma anterior no tiene porqué ser mínimo. De hecho, puede tener muchos estados inaccesibles.
- Para obtener el autómata conexo (**que no mínimo!**) se puede construir la tabla de transición para f' partiendo del estado $CLAUS_\lambda(q_0)=q_0'$, y definiendo a continuación todos los estados $f'(q_0',a)$, hasta que todos los nuevos estados que surjan hayan sido ya definidos.

Ejemplo anterior: AFD conexo equivalente

	A'	0	1
→	{p}	{q,r}	{q,r}
*	{q,r}	{p,r}	{q,r}
	{p,r}	{q,r}	{q,r}

$L(A')=\{x10^n|x\in\{0,1\}^* \text{ y n es par o } 0\}\cup\{0^n|\ n \text{ es impar}\}$

Teorema 2:
Dado un AFD $A=(Q,\Sigma,f,q_0,F)$, existe un AFND A' tal que $L(A)=L(A')$.

Demostración:
Trivial. Cualquier AFD puede entenderse, como un AFND. Consideramos el AFND $A'=(Q,\Sigma,f',q_0,F)$, donde:

a) $f'(q,\lambda)=\varnothing$, para todo $q\in Q$
b) $f'(q,a)=\{f(q,a)\}$, para todo $q\in Q$ y $a\in\Sigma$

Es obvio que $L(A)=L(A')$, por lo que $A'\equiv A$.

Teorema 3:
Las clases de los lenguajes aceptados por AFDs y AFND's son idénticas.

Ejemplo:

1. Encontrar un AFD conexo equivalente al autómata $AFND_3$.

	AFND$_3$	a	b	λ
→*	p	{q}		
	q	{q,r,s}	{p,r}	{s}
	r		{p,s}	{r,s}
*	s			{r}

2. Encontrar un AFD equivalente al siguiente autómata A:

	A	0	1
→	q_0	$\{q_1\}$	
	q_1	$\{q_1, q_2\}$	
*	q_2		$\{q_2\}$

6.2.2 Equivalencia AFs y Gramáticas Regulares

Teorema:

Dado una gramática lineal por la izquierda $G=(\Sigma_N,\Sigma_T,S,P)$, existe un AFND A tal que L(A)=L(G).

Demostración:

Considérese el autómata $A=(\Sigma_N\cup\{p\},\Sigma_T,f,p,\{S\})$, donde la función f viene definida de la forma siguiente:

- $B \in f(C,a)$ sii $B::=Ca \in P$, para todo $a\in\Sigma_T$ y $B,C\in\Sigma_N$
- $B \in f(p,a)$ sii $B::=a \in P$, para todo $a\in\Sigma_T$ y $B\in\Sigma_N$
- $S \in f(p,\lambda)$ sii $S::=\lambda \in P$

Se puede observar lo siguiente:

$a_1a_2...a_n\in L(G) \Leftrightarrow S\rightarrow V_{n-1}a_n\rightarrow\ldots\rightarrow V_1\ldots a_n\rightarrow a_1a_2\ldots a_n$

$\Leftrightarrow S \in f^*(p, a_1a_2...a_n)$

$\Leftrightarrow a_1a_2...a_n\in L(A)$

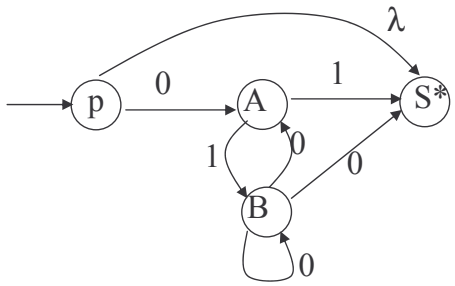
por lo que se demuestra el teorema.

Ejemplo:

Sea la gramática G=({S,A,B},{0,1},S,P) con

P={S::=B0|A1|λ, A::=B0|0, B::=B0|A1}

Un autómata equivalente es A=({S,A,B,p},{0,1},f,p,{S}) con f definido por es siguiente diagrama:

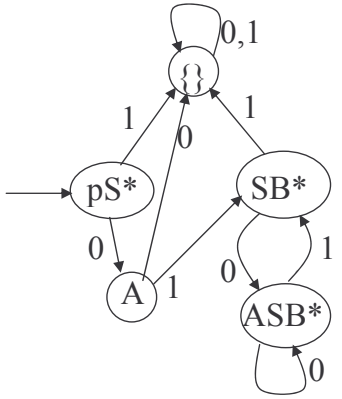


Nota:

- En general, los autómatas así construidos no tiene porqué ser deterministas.

El AFD equivalente al ejemplo anterior es:

A'=({pS,SB,{},A,ASB},{0,1},f',pS,{pS,SB,ASB}):



Este AFD ya es mínimo.

Teorema:
Dado una gramática lineal por la derecha $G=(\Sigma_N,\Sigma_T,S,P)$, existe un AFND A tal que $L(A)=L(G)$.

Demostración:
Considérese el autómata $A=(\Sigma_N\cup\{p\},\Sigma_T,f,S,\{p\})$, donde la función f viene definida de la forma siguiente:

- $C\in f(B,a)$ sii $B::=aC \in P$, para todo $a\in\Sigma_T$ y $B,C\in\Sigma_N$
- $p\in f(B,a)$ sii $B::=a \in P$, para todo $a\in\Sigma_T$ y $B\in\Sigma_N$
- $p\in f(S,\lambda)$ sii $S::=\lambda \in P$

Se puede observar lo siguiente:

$a_1a_2...a_n\in L(G) \Leftrightarrow S\rightarrow a_1V_1\rightarrow...\rightarrow a_1...a_{n-1}V_{n-1}\rightarrow a_1a_2...a_{n-1}a_n$
$\Leftrightarrow p\in f^*(S, a_1a_2...a_n)$
$\Leftrightarrow a_1a_2...a_n\in L(A)$, por lo que $L(A)=L(G)$.

Ejemplo:
$G=(\{V_0,V_1,V_2\},\{a,b\},V_0,\{V_0::=aV_1|aV_2|\lambda,V_1::=aV_2|b,V_2::=bV_0\})$
El autómata equivalente:

	A	a	b	λ
→	V_0	$\{V_1,V_2\}$		$\{V_f\}$
	V_1	$\{V_2\}$	$\{V_f\}$	
	V_2		$\{V_0\}$	
*	V_f			

$L(A)=\{(xb)^n|x\in\{aa, a\}, n\geq 0\}$

¿Bajo que condiciones se obtendría un autómata determinista?
¿Cómo sería el autómata determinista para el ejemplo?

Teorema:
Dado un AFD $A=(Q,\Sigma,f,q_0,F)$ existe una gramática lineal por la derecha G tal que $L(A)=L(G)$.

Demostración:
Considérese la gramática $G=(Q,\Sigma,q_0,P)$, donde P se define por:

- $q_i::=aq_j \in P$ sii $f(q_i,a)=q_j$
- $q_i::=a \in P$ sii $f(q_i,a)\in F$
- $q_0::=\lambda \in P$ sii $q_0\in F$

Se puede observar que:

$a_1a_2...a_n\in L(A) \Leftrightarrow f^*(q_0, a_1a_2...a_n)\in F$
$\Leftrightarrow q_0\rightarrow^*a_1a_2...a_{n-1}a_n$
$\Leftrightarrow a_1a_2...a_n\in L(G)$, lo que demuestra el teorema.

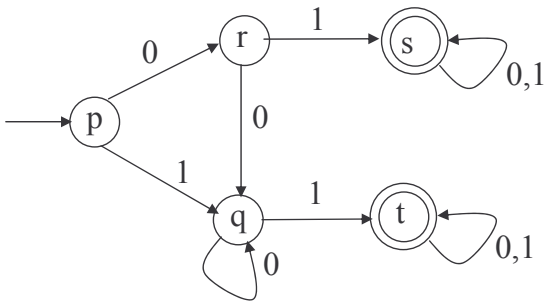
Ejemplos:
1. Sea el siguiente autómata:

	AFD	0	1
→	q_0	q_1	q_0
	q_1	q_1	q_2
*	q_2	q_1	q_0

La gramática equivalente es:
$G=(\{q_0, q_1, q_2\},\{0,1\},q_0,P)$ con
$P=\{q_0::=0q_1|1q_0, q_1::=0q_1|1q_2|1, q_2::=0q_1|1q_0\}$

A partir de esta gramática construye un autómata y comprueba que éste es equivalente al autómata A.

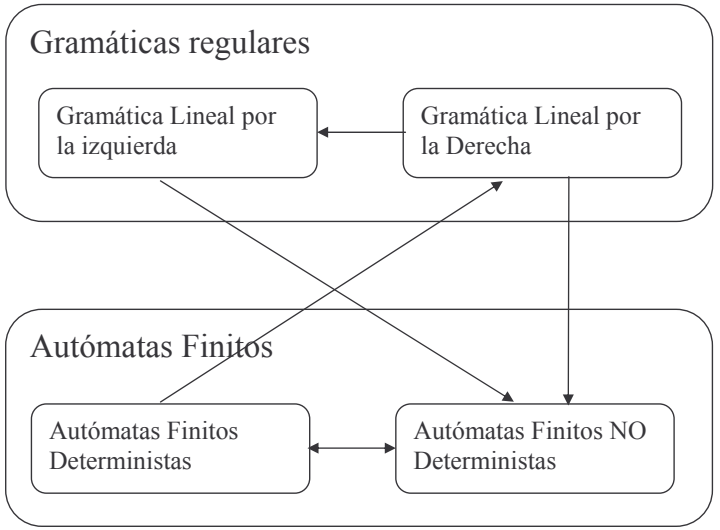
2. Sea el siguiente AFD A=({p,r,q,s,t},{0,1},f,p,{s,t}) con f definido por el siguiente grafo de transición:



Realiza los siguientes pasos:

a. Obtén una gramática lineal por la derecha G tal que L(G)=L(A).
b. A partir de G, construye un AFND A' tal que L(G)=L(A').
c. A partir de A' obtén un AFD A'' equivalente.

Teorema:
Las clases de los lenguajes aceptados por autómatas finitos y generados por gramáticas regulares son idénticas.



Nota:

- Los algoritmos vistos hasta ahora permiten convertir cualquier representación de un lenguaje regular (autómatas finitos deterministas y no deterministas, gramáticas lineales por la izquierda o por la derecha) en otra
- Los algoritmos permiten comprobar la equivalencia de dos representaciones cualesquiera de lenguajes regulares
  - Pasos:
    - Convertir ambas representaciones a AFD
    - Minimizar los AFD obtenidos
    - Comparar los AFD mínimos

Ejemplos:

1. Sea la gramática G=({S,A,B},{0,1},S,P) con
   P={S::=0B|1A|λ|0, A::=0B|0, B::=0B|1A|0}
   Obtén un autómata determinista mínimo equivalente.

2. Obtén una gramática lineal por la izquierda equivalente al siguiente autómata:

	A	0	1
→	p	r	q
	r	q	s
	q	q	t
*	s	s	s
*	t	t	t

3. Compruebe si el siguiente AFND y la siguiente gramática representan el mismo lenguaje:

   Gramática: G=({S,A,B},{0,1},S,P) con
   P={S::=0B|1A|λ|0, A::=0B|0, B::=0B|1A|0}

   AFND: