Construcción de Autómatas de Estados Finitos

DOCENTES:

DR. ING. ALFREDO IGLESIAS

MG. ING. NORA COSTA

Temas

- Autómatas finitos.
- Generación de autómatas a partir de expresiones regulares.
- Autómatas finitos deterministas.
- Autómatas finitos no deterministas.
- Construcción de Thompson.
- Paso de un autómata finito no determinista a uno determinista.
- Cerraduras ε de un conjunto de estados.
- Construcción del subconjunto.

Autómatas finitos

- Autómatas finitos o máquinas de estados finitos
- Se pueden utilizar para describir el proceso de reconocimiento de patrones en cadenas de entrada.
- Se pueden utilizar para construir analizadores léxicos.

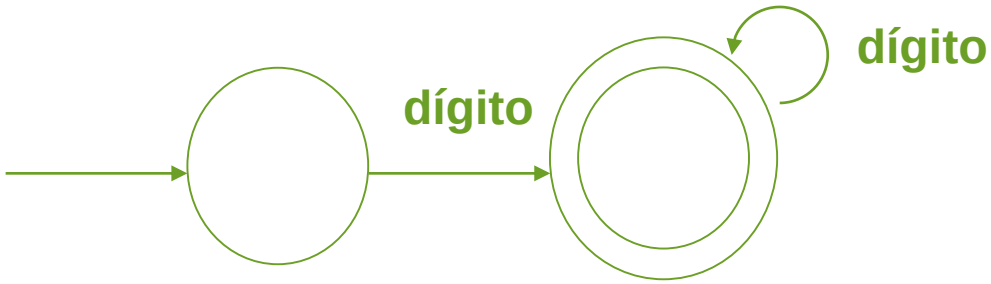
Autómatas finitos deterministícos (AFD)

- Se limitan a aceptar o no una determinada cadena recibida en la entrada.
- La salida solo tendrá dos valores posibles aceptar o no aceptar la entrada.

Autómatas finitos deterministicos

Ejemplo 1, número natural sin signo:

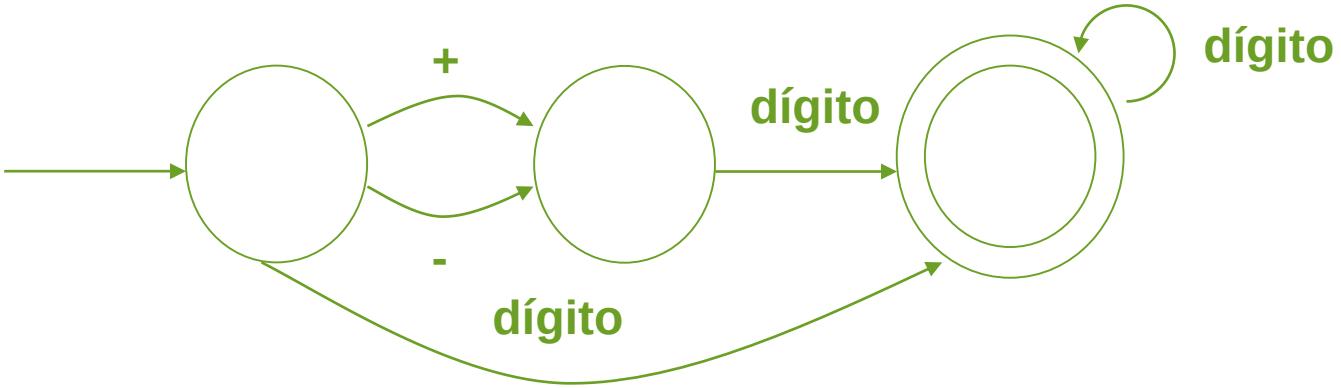
nat = [0-9]+



Autómatas finitos deterministicos

Ejemplo 2, número natural con signo:

natconSigno = (+|-)? nat

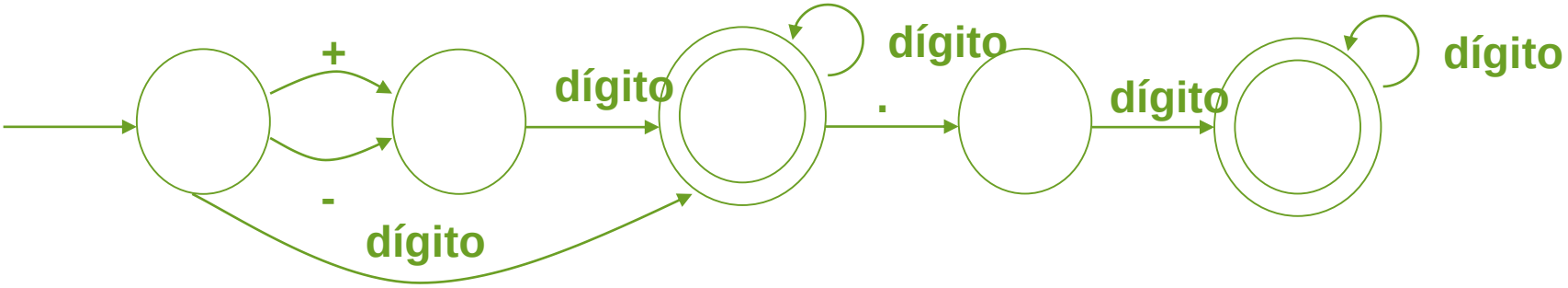


Autómatas finitos deterministicos

Ejemplo 3, número con parte fraccionaria:

número = natconSigno("." nat)?(E natconSigno)?

- Se tienen dos estados de aceptación, ya que la parte decimal es opcional.



Autómatas finitos deterministicos

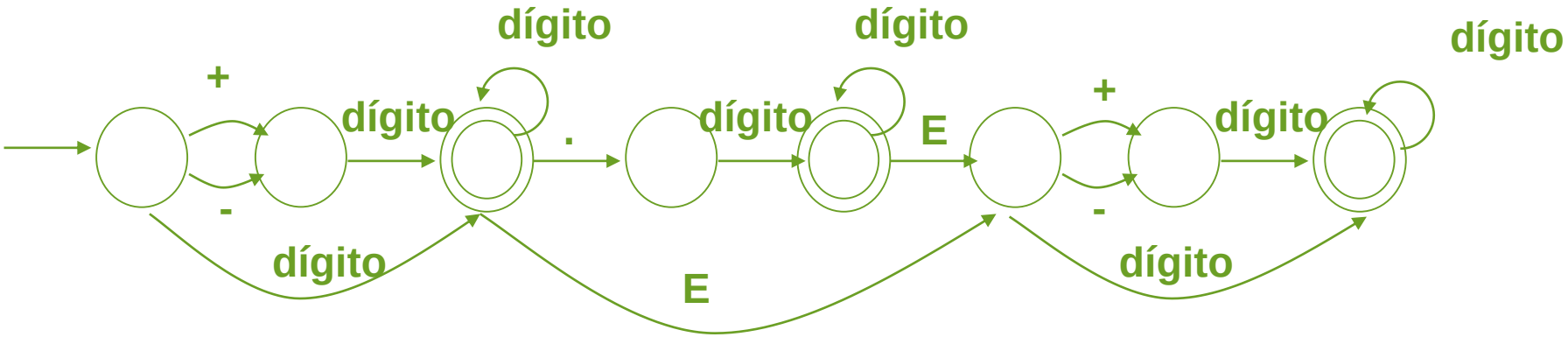
Ejemplo 4, número de punto flotante:

nat = [0-9]+

natconSigno = (+|-)? nat

número = natconSigno("." nat)?(E natconSigno)?

- Se le agrega la parte exponencial opcional.



Autómatas deterministas (AFD) y no deterministas (AFN)

AFD

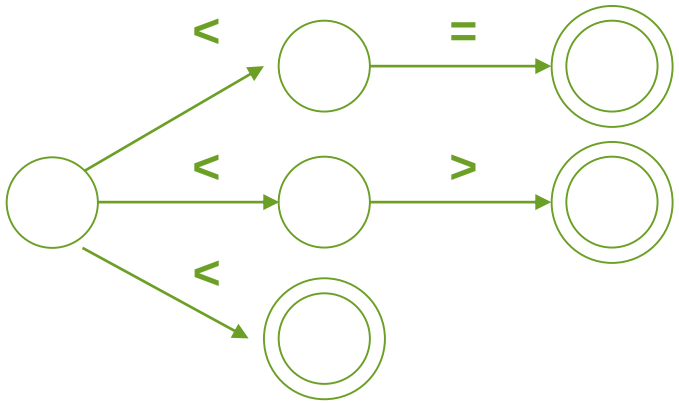
- Debe tener una transición para cada estado y carácter .
- Aquellas transiciones que dan errores como resultado, se dejan fuera del diagrama para el AFD.
- La definición de un AFD no especifica lo que ocurre cuando se presenta un error.
- No especifica la acción que tomará un programa al alcanzar un estado de aceptación.

AFN

- Puede tener ninguna, una o más transiciones de un estado sobre el mismo símbolo de entrada.
- Puede tener transiciones vacías o transiciones ε.

Autómatas no deterministas

- Ejemplo
  - Varios tokens que comienzan con el mismo carácter <, <= y <>.
  - No corresponde a un AFD, de un estado siempre debe haber una transición única etiquetada con cada carácter.
  - Se debería poder combinar todos los tokens en un AFD gigante, pero es una tarea muy compleja.



Generación de Autómatas a partir de Expresiones Regulares

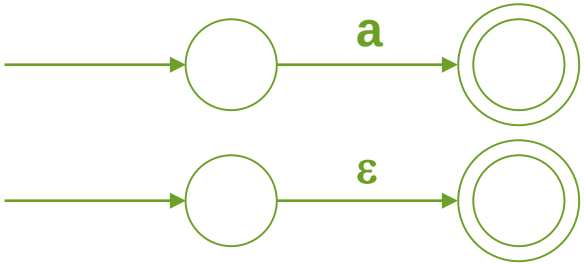
- El proceso de la construcción de un analizador léxico se puede automatizar en tres pasos partiendo de la expresión regular:
  1. Expresión regular.
  2. Traducir una expresión regular en un AFN.
  3. Traducir un AFN en un AFD.

Construcción de Thompson

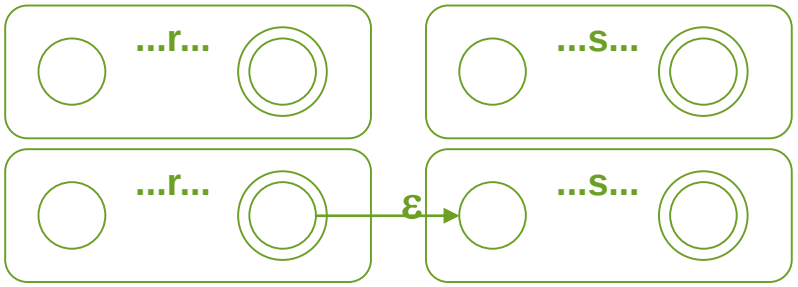
- AFN para cada expresión regular básica.
- Obtención de cada operación de expresión regular al conectar entre sí los AFN de las subexpresiones.

Construcción de Thompson

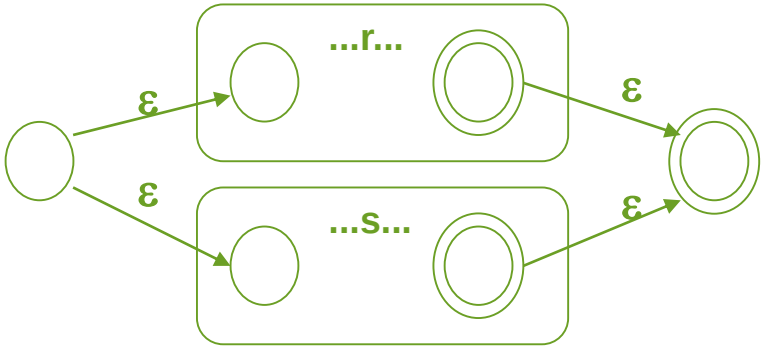
EXPRESIONES REGULARES BÁSICAS



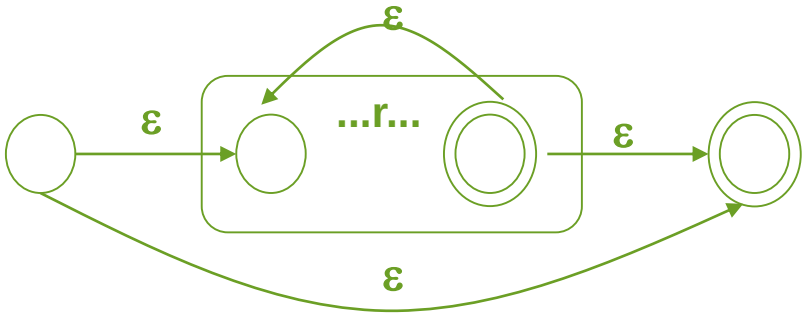
CONCATENACIÓN



SELECCIÓN DE ALTERNATIVAS

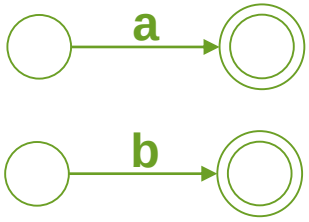


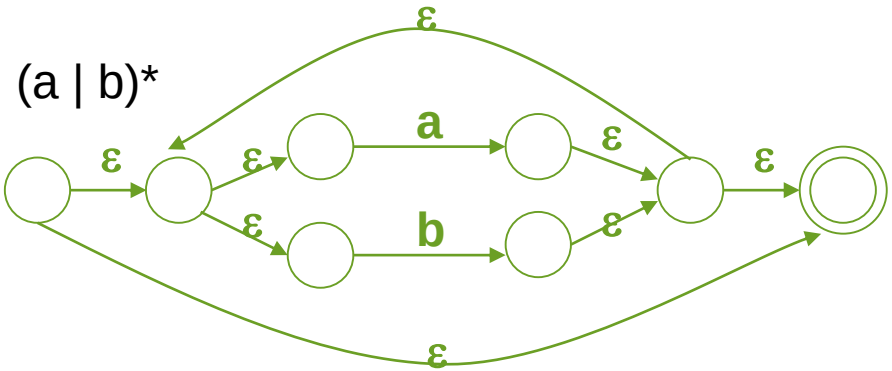
REPETICIÓN O CERRADURA DE KLEENE

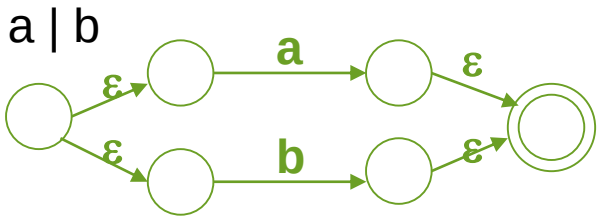


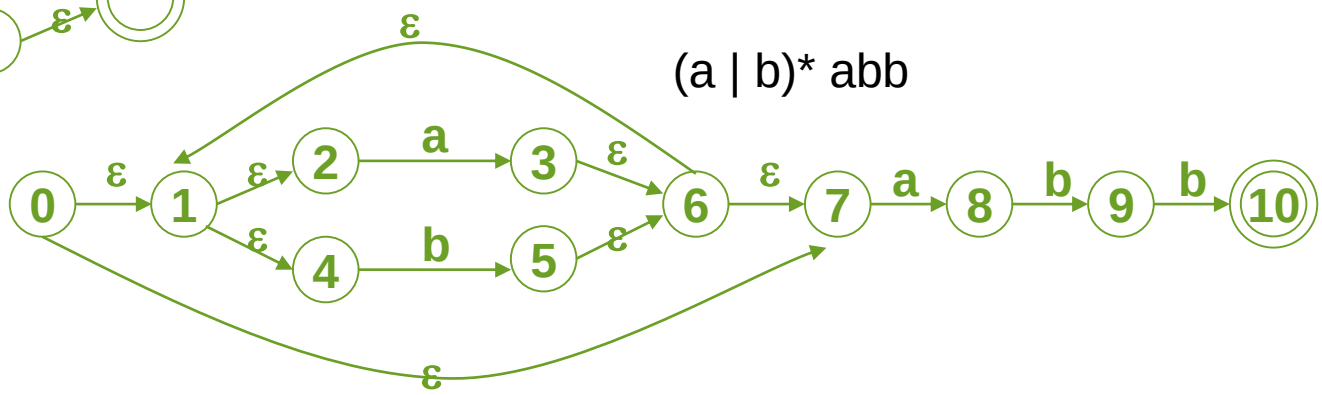
Construcción de Thompson

Ejemplo 1: (a | b)* abb



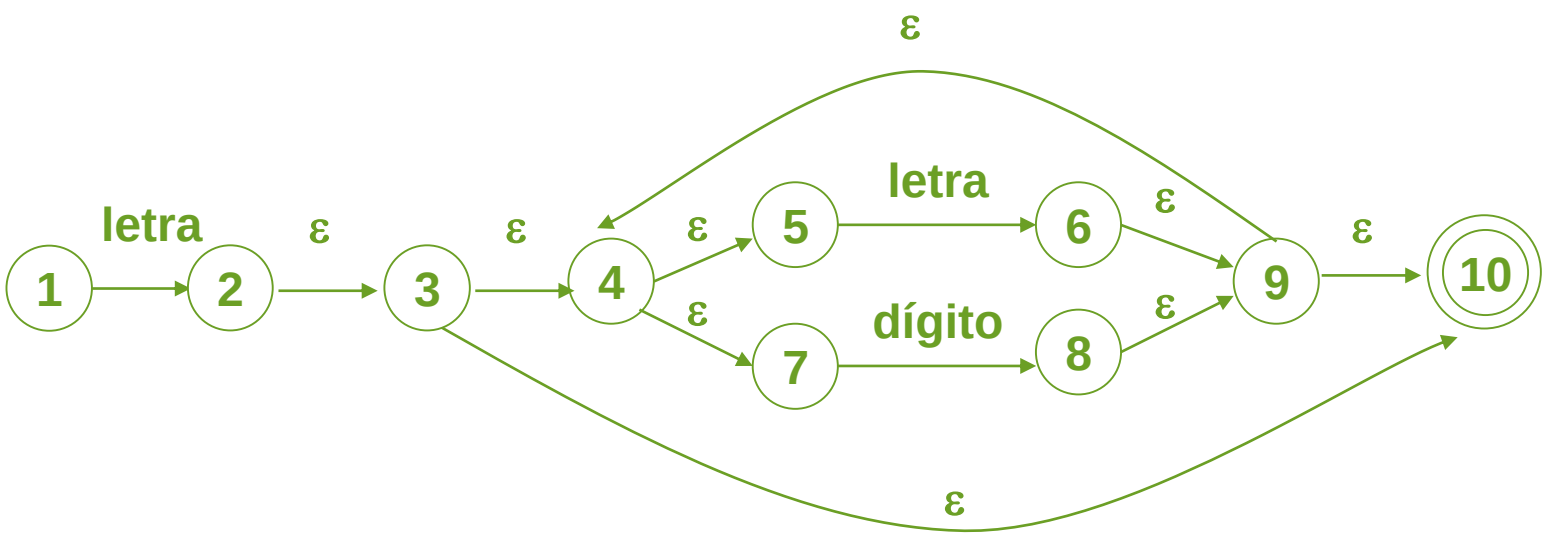






Construcción de Thompson

Ejemplo 2: letra (letra | dígito)*



Paso de un autómata finito no determinista a uno determinista

Dado un AFN arbitrario, se construirá un AFD equivalente, que acepte las mismas cadenas.

- Método con el que se eliminen las transiciones ε y las transiciones múltiples de un estado en un carácter de entrada simple.

1. Cerraduras ε
   - Son el conjunto de todos los estados que pueden alcanzar las transiciones ε desde un estado o estados.
   - La eliminación de las transiciones ε implica el construir cerraduras ε.
2. Construcción de subconjuntos
   - Considera conjuntos de estados en lugar de estados simples.
   - La eliminación de transiciones múltiples en un carácter de entrada simple implica mantenerse al tanto del conjunto de estados que son alcanzables al igualar un carácter simple.

El AFD que construimos va a tener como sus estados los conjuntos de estados del AFN original.

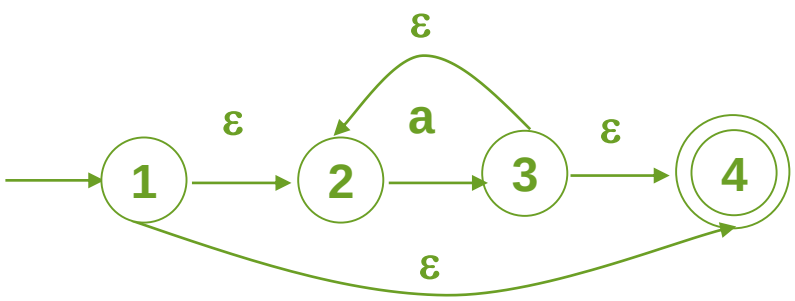
Cerraduras ε de un conjunto de estados

- La cerradura ε de un estado simple s es el conjunto de estados alcanzables por una serie de cero o más transiciones ε.
- La cerradura ε de un estado siempre contiene al estado mismo.

Cerraduras ε de un conjunto de estados

CONSTRUCCIÓN DE THOMPSON

- Ejemplo: a*



CERRADURAS

- 1 = {1, 2, 4}
- 2 = {2}
- 3 = {2, 3, 4}
- 4 = {4}

Construcción del subconjunto

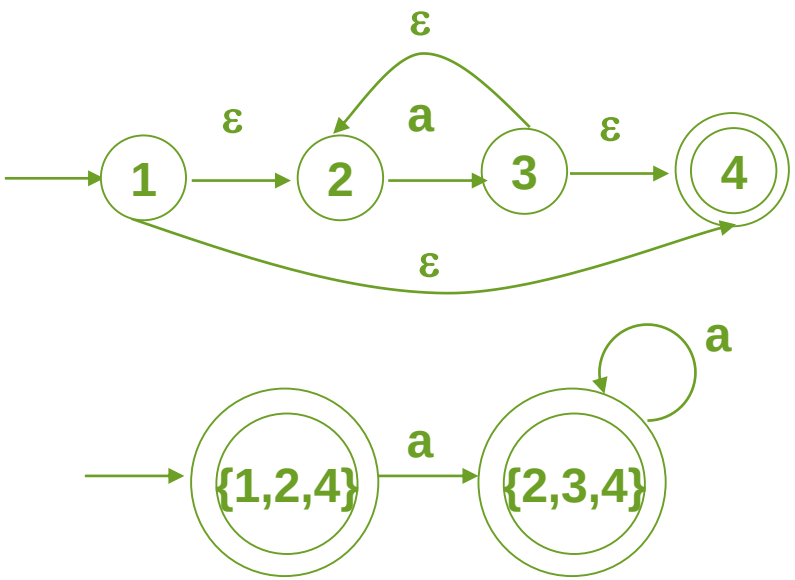
Para la construcción de un AFD a partir de un AFN M dado, M

- Calcularnos la cerradura ε del estado de inicio de M, esto se convierte en el estado de inicio de M.
- Para este conjunto, y para cada conjunto subsiguiente, calculamos las transiciones en los caracteres.

Construcción del subconjunto

CONSTRUCCIÓN DE THOMPSON

- Ejemplo: a*



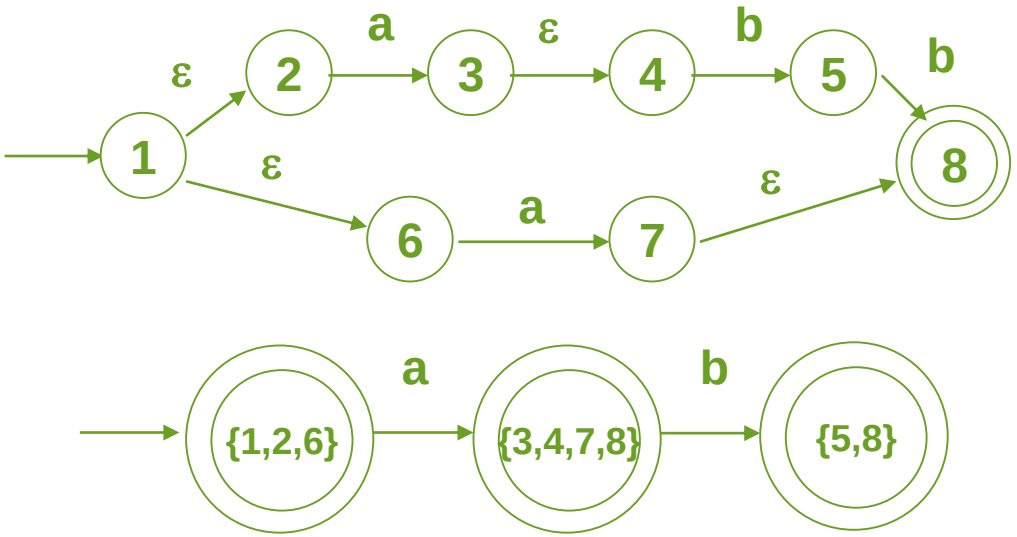
SUBCONJUNTOS

- Estado inicial del AFD: {1, 2, 4} (cerradura ε del estado 1 del AFN)
- $\{1,2,4\}_a = \{3\}$
- cerradura ε de {3}:
- 3 = {2, 3, 4} (estado alcanzado desde el estado inicial de AFD con a)
- $\{2,3,4\}_a = \{3\}$ (existe una transición a desde {2, 3, 4} hacia sí mismo)

Paso de un autómata finito no determinista a uno determinista

CONSTRUCCIÓN DE THOMPSON

- Ejemplo: ab|a



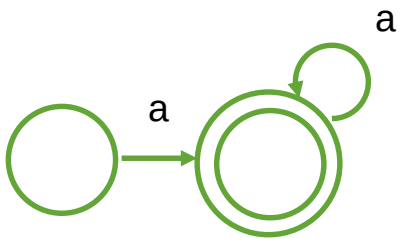
CERRADURAS Y SUBCONJUNTOS

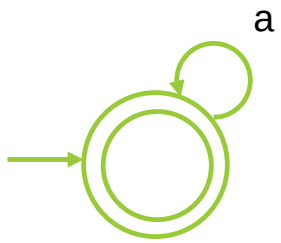
- Cerradura ε
- 1 = {1,2,6} (estado inicial del AFD)
- Subconjunto
- $\{1,2,6\}_a = \{3,7\}$ (estado AFD)
- $\{1,2,6\}_b = \{\}$
- Cerradura ε
- {3,7} = {3,4,7,8}
- Subconjunto
- $\{3,4,7,8\}_b = \{5,8\}$
- $\{3,4,7,8\}_a = \{\}$
- Cerradura ε
- {5,8} = {5,8}

Minimización del número de estados en un AFD

El proceso para derivar un AFD a partir de un expresión regular da como resultado un AFD que puede ser más complejo de lo necesario.

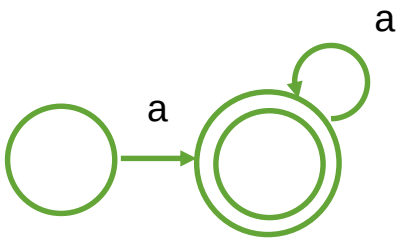
Ejemplo: a*

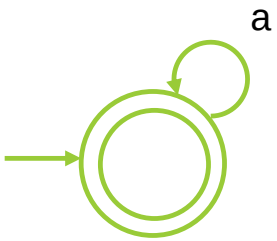




Minimización del número de estados en un DFA

- Se crean dos conjuntos, uno compuesto de todos los estados de aceptación y el otro compuesto de todos los estados de no aceptación.
  - Considere las transiciones en cada carácter a del alfabeto. Si todos los estados de aceptación tienen transiciones en a para estados de aceptación, entonces esto define una transición a desde el nuevo estado de aceptación (el conjunto de todos los estados de aceptación antiguos) hacia sí misma.
  - Si todos los estados de aceptación tienen transiciones en a hacia estados de no aceptación, entonces esto define una transición a desde el nuevo estado de aceptación hacia el nuevo estado de no aceptación (el conjunto de todos los estados de no aceptación antiguos).

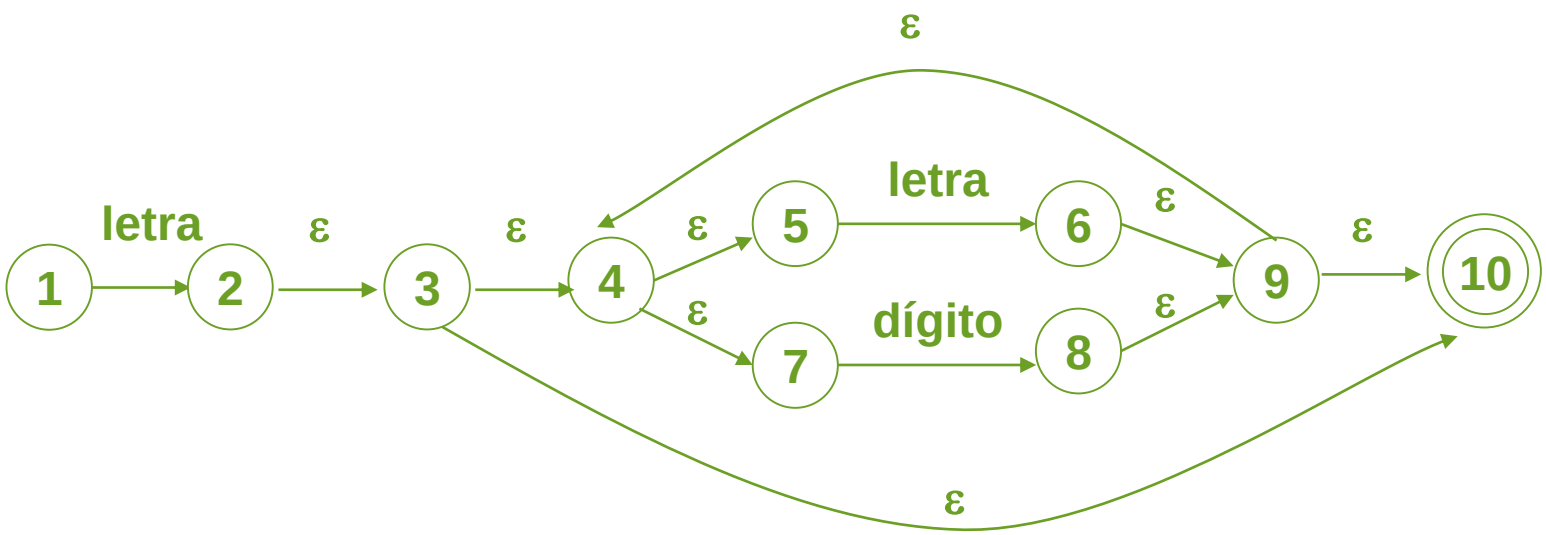




Minimización del número de estados en un AFD

Ejemplo: letra (letra | dígito)*

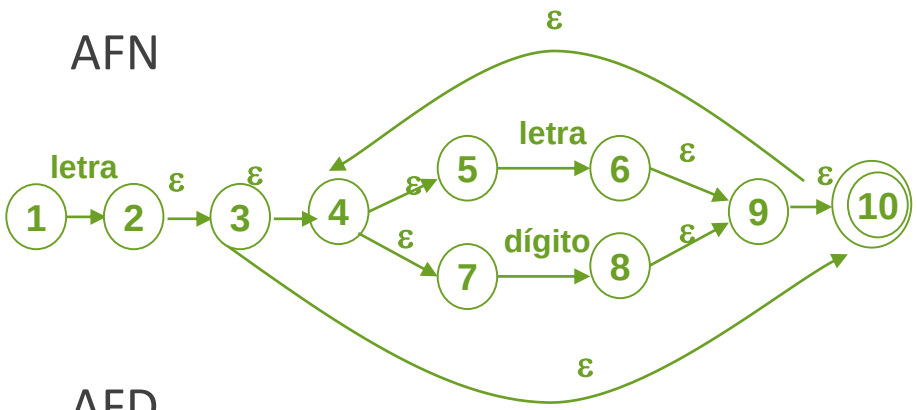
Construcción de Thompson



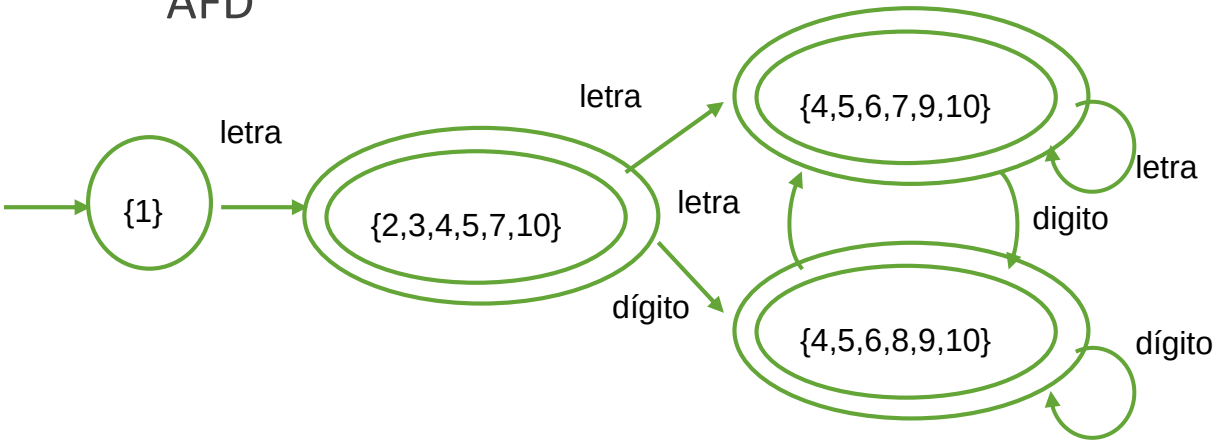
Minimización del número de estados en un AFD

Ejemplo: letra (letra | dígito)*

AFN



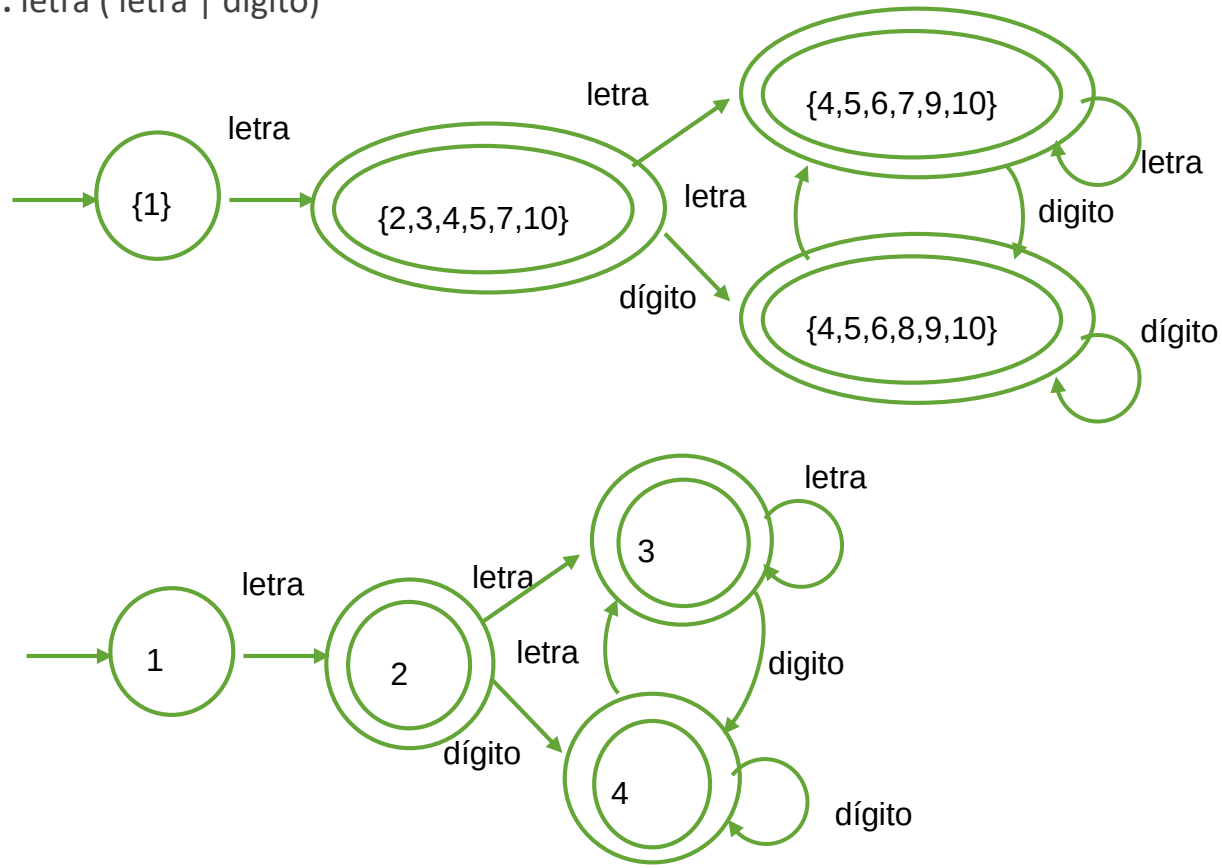
AFD



- Cerradura
- 1 = {1}
- Construcción
- $\{1\}_{letra} = \{2\}$
- Cerradura
- 2 = {2,3,4,5,7,10}
- Construcción
- $\{2,3,4,5,7,10\}_{letra} = \{6\}$
- $\{2,3,4,5,7,10\}_{digito} = \{8\}$
- Cerradura
- 6 = {4,5,6,7,9,10}
- 8 = {4,5,6,8,9,10}
- Construcción
- $\{4,5,6,7,9,10\}_{letra} = \{6\}$
- $\{4,5,6,7,9,10\}_{digito} = \{8\}$
- $\{4,5,6,8,9,10\}_{letra} = \{6\}$
- $\{4,5,6,8,9,10\}_{digito} = \{8\}$

Minimización del número de estados en un AFD

Ejemplo: letra (letra | dígito) *



Minimización del número de estados en un AFD

- Ejemplo: letra (letra 1 dígito) *
  - Tenía cuatro estados compuestos del estado inicial y tres estados de aceptación.
  - Los tres estados de aceptación tienen transiciones a otros estados de aceptación tanto en letra como en dígito y ninguna otra transición
  - Los tres estados aceptación no se pueden distinguir por ningún carácter, y el algoritmo de minimización da como resultado la combinación de los tres estados de aceptación en uno