



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Departamento de Estadística, I.O. y Computación
Teoría de Automatas y Lenguajes Formales

PRACTICA 4: NFA con y sin ϵ -transiciones

4.1. Requisito de codificación

Indentación:

Una correcta indentación es uno de los mecanismos más adecuados para que el código fuente resulte claro y legible.

Recomendamos utilizar un *tamaño de indentación* adecuado para que el código no aparezca muy desplazado hacia la derecha: 2 ó 3 espacios en cada sangrado es lo adecuado. La mayoría de los editores poseen mecanismos para fijar la equivalencia en espacios que corresponde a un tabulador. En concreto en vi, esto se puede fijar con el comando `: set ts=N` (*set tabspace to N*). Se puede configurar el editor para que siempre utilice esta definición cada vez que se ejecuta el editor. En el caso del editor vim basta incluir el comando en el fichero `/.vimrc`.

Un aspecto esencial es no mezclar tabulaciones y espacios: se debe decidir cuál de las dos opciones utilizar y usar siempre la misma. En las prácticas de la asignatura recomendamos utilizar dos espacios en cada indentación.

4.2. Introducción

Un NFA se define mediante una tupla $(Q, \Sigma, \delta, q_0, F)$. La diferencia entre un NFA con ϵ -transiciones y uno que sí las tiene es que en el primer caso, la función de transición se define $\delta : Q \times \Sigma \rightarrow \mathfrak{P}(Q)$, mientras que en un NFA con ϵ -transiciones, la función de transición es $\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \mathfrak{P}(Q)$, es decir, el autómata puede producir transiciones sin consumir ningún símbolo de la cadena de entrada (ϵ -transiciones).

Es evidente que un NFA sin ϵ -transiciones es un caso particular de NFA con ϵ -transiciones (en el que la posibilidad de tener este tipo de transiciones, no se usa), pero también es posible pasar de un NFA con ϵ -transiciones a uno equivalente (en el sentido de que reconozca el mismo lenguaje) sin ϵ -transiciones.

Para pasar de un NFA con ϵ -transiciones $M \equiv (Q, \Sigma, \delta, q_0, F)$ a uno sin ϵ -transiciones $M' \equiv (Q, \Sigma, \delta', q_0, F')$ equivalente, es decir $L(M) = L(M')$ basta con definir:

- $F' = F \cup \{q \in Q \mid \epsilon\text{-clausura}(q) \cap F \neq \emptyset\}$
- $\forall a \in \Sigma, q \in Q$
 $\delta'(q, a) = \widehat{\delta}(q, a)$

```

forall (q ∈ T) do
  begin
  push(q)
  ε-clausura(T) := T;
  while (not pila_vacia) do
    begin
    p := pop();
    for (cada estado u con una arista desde p a u etiquetada con ε) do
      if u ∉ ε-clausura(T) then
        begin
        ε-clausura(T) := ε-clausura(T) ∪ {u}
        push(u);
        end;
    end;
  end;

```

Figura 4.1: Un algoritmo para el cálculo de la ε-clausura de un conjunto de estados, T

Donde la función $\hat{\delta}$ está definida como:

$$\hat{\delta}(q, a) = \epsilon\text{-clausura}(\delta(\epsilon\text{-clausura}(q), a))$$

La figura 4.1 presenta el pseudocódigo de un algoritmo para el cálculo de la ε-clausura de un conjunto de estados.

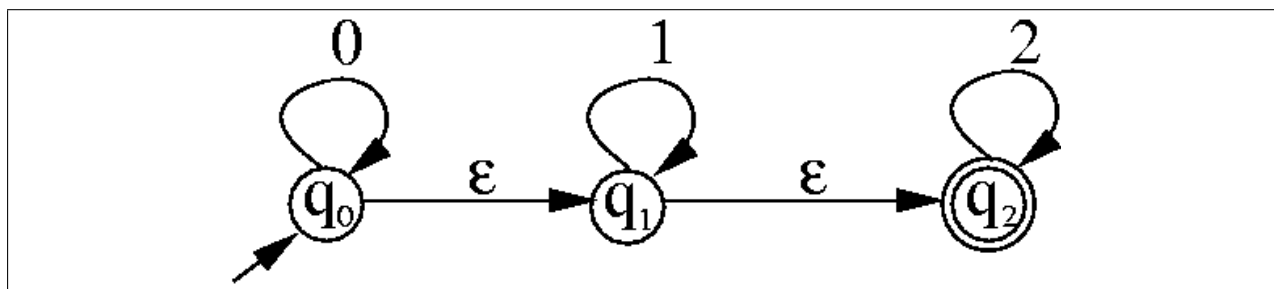


Figura 4.2: NFA con ε-transiciones

Veamos un ejemplo práctico de aplicación de esta transformación. Consideremos el NFA que se muestra en la figura 4.2. Este autómata acepta cadenas del lenguaje representado por la expresión regular $0^*1^*2^*$. Calculemos la nueva función de transición del NFA sin ε-transiciones correspondiente:

$$\hat{\delta}(q_0, 0) = \epsilon\text{-clausura}(\delta(\epsilon\text{-clausura}(q_0), 0)) = \epsilon\text{-clausura}(\delta(\{q_0, q_1, q_2\}, 0)) = \epsilon\text{-clausura}(q_0) = \{q_0, q_1, q_2\}$$

$$\hat{\delta}(q_0, 1) = \epsilon\text{-clausura}(\delta(\epsilon\text{-clausura}(q_0), 1)) = \epsilon\text{-clausura}(\delta(\{q_0, q_1, q_2\}, 1)) = \epsilon\text{-clausura}(q_1) = \{q_1, q_2\}$$

$$\hat{\delta}(q_0, 2) = \epsilon\text{-clausura}(\delta(\epsilon\text{-clausura}(q_0), 2)) = \epsilon\text{-clausura}(\delta(\{q_0, q_1, q_2\}, 2)) = \epsilon\text{-clausura}(q_2) = \{q_2\}$$

$$\hat{\delta}(q_1, 0) = \epsilon\text{-clausura}(\delta(\{q_1, q_2\}, 0)) = \epsilon\text{-clausura}(\emptyset) = \emptyset$$

$$\widehat{\delta}(q_1, 1) = \epsilon - \text{clausura}(\delta(\{q_1, q_2\}, 1) = \epsilon - \text{clausura}(q_1) = \{q_1, q_2\}$$

$$\widehat{\delta}(q_1, 2) = \epsilon - \text{clausura}(\delta(\{q_1, q_2\}, 2) = \epsilon - \text{clausura}(q_2) = \{q_2\}$$

$$\widehat{\delta}(q_2, 0) = \epsilon - \text{clausura}(\delta(\{q_2\}, 0) = \epsilon - \text{clausura}(\emptyset) = \emptyset$$

$$\widehat{\delta}(q_2, 1) = \epsilon - \text{clausura}(\delta(\{q_2\}, 1) = \epsilon - \text{clausura}(\emptyset) = \emptyset$$

$$\widehat{\delta}(q_2, 2) = \epsilon - \text{clausura}(\delta(\{q_2\}, 2) = \epsilon - \text{clausura}(q_2) = \{q_2\}$$

Si agrupamos estos resultados para formar la tabla de transiciones del NFA sin ϵ -transiciones, obtenemos la tabla 4.1.

δ'	0	1	2
q_0	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
q_1	\emptyset	$\{q_1, q_2\}$	$\{q_2\}$
q_2	\emptyset	\emptyset	$\{q_2\}$

Cuadro 4.1: Tabla de transiciones del NFA sin ϵ -transiciones

Y si los escribimos en forma de diagrama de transiciones, obtenemos el que se muestra en la figura 4.3

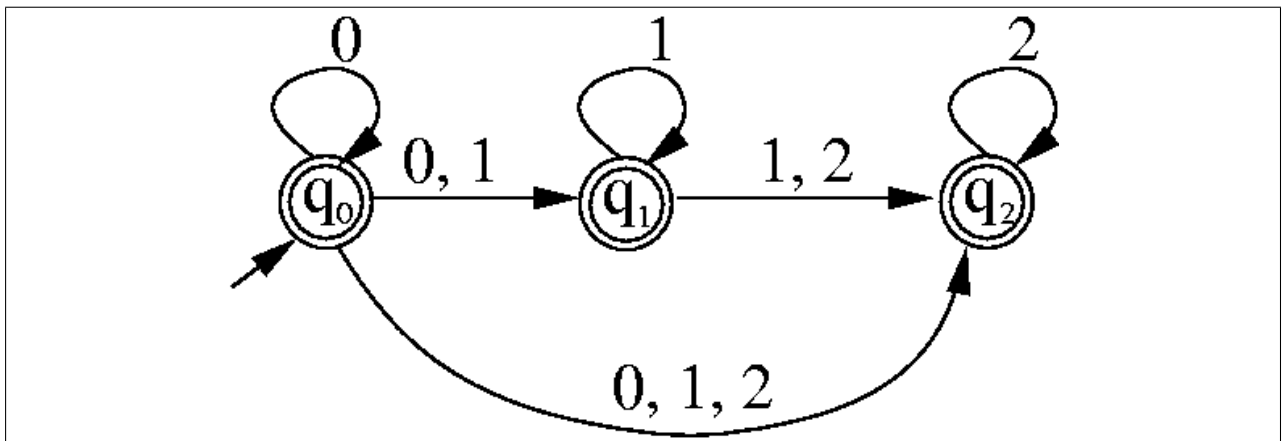


Figura 4.3: NFA sin ϵ -transiciones

4.3. Práctica: Eliminación de las ϵ -transiciones en un NFA

La práctica consistirá en la realización de un programa que lea desde un fichero la especificación de un NFA genere un fichero correspondiente a la especificación de un NFA equivalente sin ϵ -transiciones.

4.3.1. Estructura de los ficheros de definición de NFAs

Tanto el fichero de entrada como el de salida especifican un NFA. Estos ficheros tendrán la extensión `.nfa` y contendrán la misma información que se ha utilizado en

prácticas anteriores. La única diferencia que introduciremos en esta práctica es que en este caso el NFA puede tener varios estados de aceptación, por lo cual en la línea 3 del fichero aparecerá la lista de estados de aceptación del autómatá separados por espacios en blanco, es decir:

- Línea 1: Número total de estados del NFA.
- Línea 2: Estado de arranque del NFA.
- Línea 3: Lista de los estados de aceptación del NFA, separados por espacios.

A continuación viene una línea para cada estado, conteniendo los siguientes números, separados entre sí por espacios en blanco.

- Número identificador del estado.
- Número de transiciones que posee el estado.

A continuación para cada una de las transiciones y separado por espacios:

- Carácter de entrada necesario para que se produzca la transición.
- Estado de destino de la transición.

También, al igual que en prácticas anteriores, una ϵ -transición se representará mediante el carácter ~ (código ASCII 126).

La línea de comando en la que se ejecutará el programa tendrá un formato como:

```
nfae2nfa in.nfa out.nfa
```

Donde `in.nfa` y `out.nfa` son respectivamente el fichero de especificación del NFA de entrada y de salida.