

# Conversión entre Expresiones Regulares y Autómatas Finitos

Matemáticas Computacionales  
(TC2020)

M.C. Xavier Sánchez Díaz  
sax@itesm.mx



# Tabla de contenidos

1 Preliminares

2 De AFs a ERs

# ¿Por qué convertir entre ERs y AFs?

## Conversión entre ERs y AFs

- Los AFs son más **fáciles de entender** (y por tanto, de **diseñar**).
- Los AFs son muy **aparatosos**: describen qué pasa en cada estado y con cada acción.
- Las ERs son más **compactas**: sólo describen cómo va a ser el *output* del autómata—el lenguaje de palabras que acepta.
- Pasar de una condición de aceptación de un problema a un AF puede ser **más complicado**.
- Explicar una condición de aceptación de un problema, a otra persona mediante una ER puede ser **muy complicado**.

# ¿Por qué convertir entre ERs y AFs?

## Conversión entre ERs y AFs

- Los AFs son más **fáciles de entender** (y por tanto, de **diseñar**).
- Los AFs son muy **aparatosos**: describen qué pasa en cada estado y con cada acción.
- Las ERs son más **compactas**: sólo describen cómo va a ser el *output* del autómatas—el lenguaje de palabras que acepta.
- Pasar de una condición de aceptación de un problema a un AF puede ser **más complicado**.
- Explicar una condición de aceptación de un problema, a otra persona mediante una ER puede ser **muy complicado**.

# ¿Por qué convertir entre ERs y AFs?

## Conversión entre ERs y AFs

- Los AFs son más **fáciles de entender** (y por tanto, de **diseñar**).
- Los AFs son muy **aparatosos**: describen qué pasa en cada estado y con cada acción.
- Las ERs son más **compactas**: sólo describen cómo va a ser el *output* del autómatas—el lenguaje de palabras que acepta.
- Pasar de una condición de aceptación de un problema a un AF puede ser **más complicado**.
- Explicar una condición de aceptación de un problema, a otra persona mediante una ER puede ser **muy complicado**.

# ¿Por qué convertir entre ERs y AFs?

## Conversión entre ERs y AFs

- Los AFs son más **fáciles de entender** (y por tanto, de **diseñar**).
- Los AFs son muy **aparatosos**: describen qué pasa en cada estado y con cada acción.
- Las ERs son más **compactas**: sólo describen cómo va a ser el *output* del autómatas—el lenguaje de palabras que acepta.
- Pasar de una condición de aceptación de un problema a un AF puede ser **más complicado**.
- Explicar una condición de aceptación de un problema, a otra persona mediante una ER puede ser **muy complicado**.

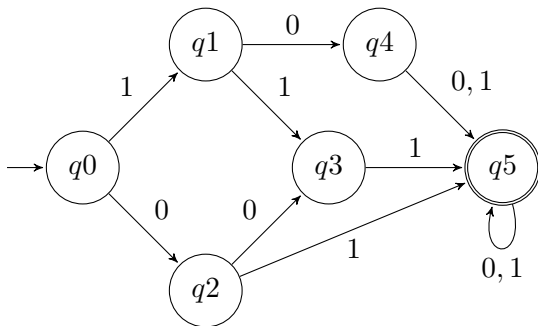
# ¿Por qué convertir entre ERs y AFs?

## Conversión entre ERs y AFs

- Los AFs son más **fáciles de entender** (y por tanto, de **diseñar**).
- Los AFs son muy **aparatosos**: describen qué pasa en cada estado y con cada acción.
- Las ERs son más **compactas**: sólo describen cómo va a ser el *output* del autómata—el lenguaje de palabras que acepta.
- Pasar de una condición de aceptación de un problema a un AF puede ser **más complicado**.
- Explicar una condición de aceptación de un problema, a otra persona mediante una ER puede ser **muy complicado**.

## Ejemplo: BSL (4-AQA)

De AFs a ERs

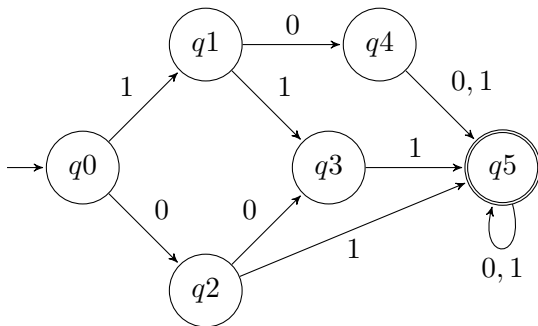


Es un AFN ya que  $q3$  no tiene transición 0.



## Ejemplo: BSL (4-AQA)

De AFs a ERs



Es un AFN ya que  $q_3$  no tiene transición 0.

# Paso a paso

## De AFs a ERs

- ➊ Agregamos un nuevo estado inicial y otro final por medio de transiciones vacías.
- ➋ Empezando desde el **final** y yendo hacia atrás, reemplazamos transiciones por sus equivalencias en ERs:
  - ➊ Primero cambiamos sus transiciones por ERs.
  - ➋ Después cambiamos las transiciones que llegan al estado.
  - ➌ Marcamos el nodo, *uniendo o concatenando* las ERs.
  - ➍ Varias ERs yendo de un estado a otro pueden reducirse a una sola **uniéndolas**.
- ➌ Continuamos reemplazando hasta tener una sola ER.

# Paso a paso

## De AFs a ERs

- ➊ Agregamos un nuevo estado inicial y otro final por medio de transiciones vacías.
- ➋ Empezando desde el **final** y yendo hacia atrás, reemplazamos transiciones por sus equivalencias en ERs:
  - ➊ Primero cambiamos sus transiciones por ERs.
  - ➋ Después cambiamos las transiciones que llegan al estado.
  - ➌ Marcamos el nodo, *uniendo o concatenando* las ERs.
  - ➍ Varias ERs yendo de un estado a otro pueden reducirse a una sola **uniéndolas**.
- ➌ Continuamos reemplazando hasta tener una sola ER.

# Paso a paso

## De AFs a ERs

- ➊ Agregamos un nuevo estado inicial y otro final por medio de transiciones vacías.
- ➋ Empezando desde el **final** y yendo hacia atrás, reemplazamos transiciones por sus equivalencias en ERs:
  - ➊ Primero cambiamos sus transiciones por ERs.
  - ➋ Después cambiamos las transiciones que llegan al estado.
  - ➌ Marcamos el nodo, *uniendo o concatenando* las ERs.
  - ➍ Varias ERs yendo de un estado a otro pueden reducirse a una sola **uniéndolas**.
- ➌ Continuamos reemplazando hasta tener una sola ER.

# Paso a paso

## De AFs a ERs

- ➊ Agregamos un nuevo estado inicial y otro final por medio de transiciones vacías.
- ➋ Empezando desde el **final** y yendo hacia atrás, reemplazamos transiciones por sus equivalencias en ERs:
  - ➊ Primero cambiamos sus transiciones por ERs.
  - ➋ Después cambiamos las transiciones que llegan al estado.
  - ➌ Marcamos el nodo, *uniendo o concatenando* las ERs.
  - ➍ Varias ERs yendo de un estado a otro pueden reducirse a una sola **uniéndolas**.
- ➌ Continuamos reemplazando hasta tener una sola ER.

# Paso a paso

## De AFs a ERs

- ➊ Agregamos un nuevo estado inicial y otro final por medio de transiciones vacías.
- ➋ Empezando desde el **final** y yendo hacia atrás, reemplazamos transiciones por sus equivalencias en ERs:
  - ➊ Primero cambiamos sus transiciones por ERs.
  - ➋ Después cambiamos las transiciones que llegan al estado.
  - ➌ Marcamos el nodo, *uniendo o concatenando* las ERs.
  - ➍ Varias ERs yendo de un estado a otro pueden reducirse a una sola **uniéndolas**.
- ➌ Continuamos reemplazando hasta tener una sola ER.

# Paso a paso

## De AFs a ERs

- ➊ Agregamos un nuevo estado inicial y otro final por medio de transiciones vacías.
- ➋ Empezando desde el **final** y yendo hacia atrás, reemplazamos transiciones por sus equivalencias en ERs:
  - ➊ Primero cambiamos sus transiciones por ERs.
  - ➋ Después cambiamos las transiciones que llegan al estado.
  - ➌ Marcamos el nodo, *uniendo o concatenando* las ERs.
  - ➍ Varias ERs yendo de un estado a otro pueden reducirse a una sola **uniéndolas**.
- ➌ Continuamos reemplazando hasta tener una sola ER.

# Paso a paso

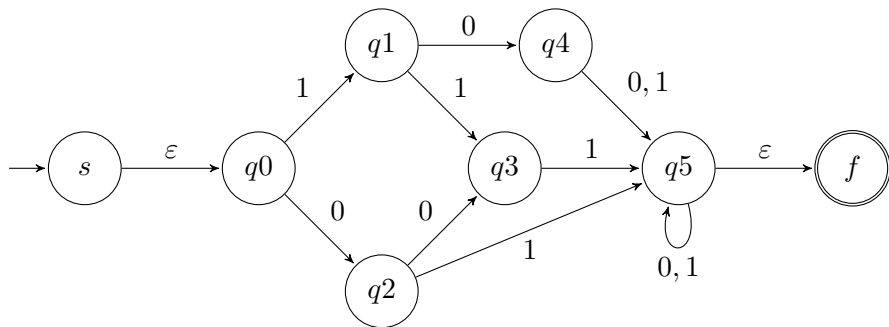
## De AFs a ERs

- ➊ Agregamos un nuevo estado inicial y otro final por medio de transiciones vacías.
- ➋ Empezando desde el **final** y yendo hacia atrás, reemplazamos transiciones por sus equivalencias en ERs:
  - ➊ Primero cambiamos sus transiciones por ERs.
  - ➋ Después cambiamos las transiciones que llegan al estado.
  - ➌ Marcamos el nodo, *uniendo o concatenando* las ERs.
  - ➍ Varias ERs yendo de un estado a otro pueden reducirse a una sola **uniéndolas**.
- ➌ Continuamos reemplazando hasta tener una sola ER.



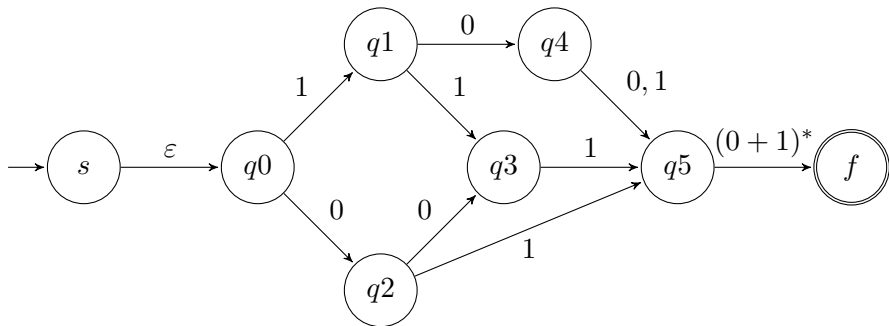
# Ejemplo: BSL (4-AQA)

De AFs a ERs



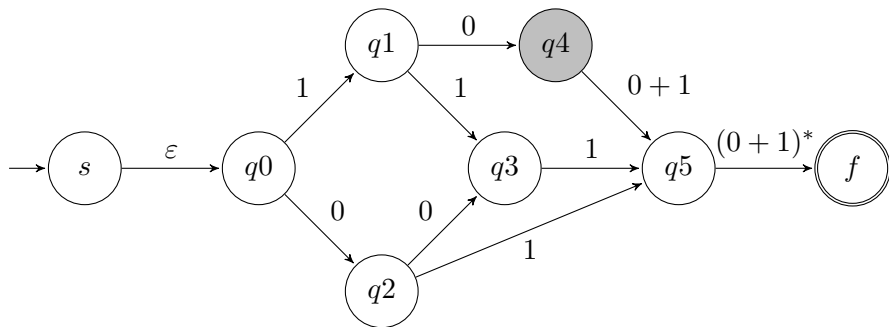
## Ejemplo: BSL (4-AQA)

De AFs a ERs



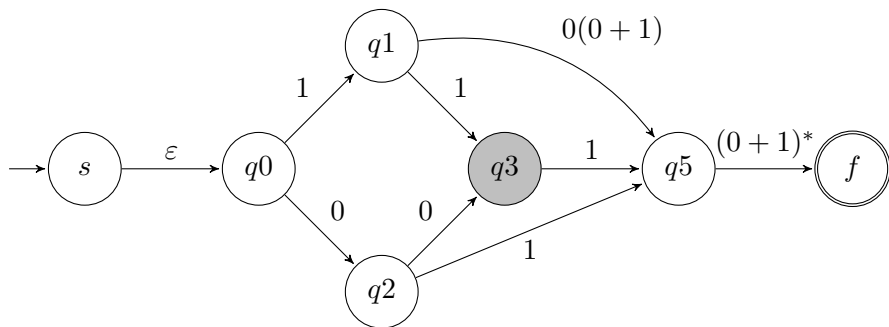
# Ejemplo: BSL (4-AQA)

De AFs a ERs



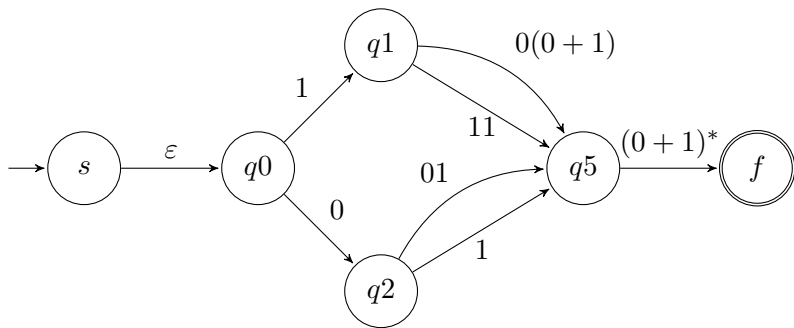
# Ejemplo: BSL (4-AQA)

De AFs a ERs



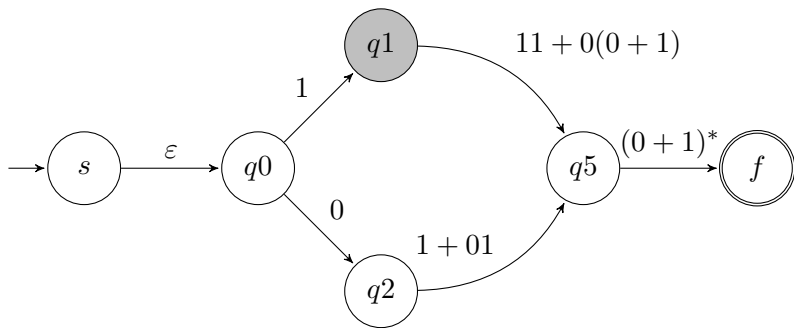
# Ejemplo: BSL (4-AQA)

De AFs a ERs



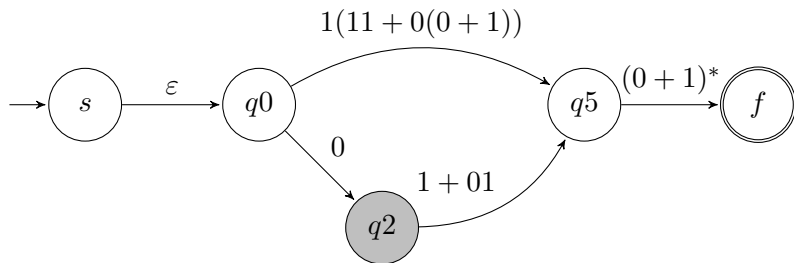
# Ejemplo: BSL (4-AQA)

De AFs a ERs



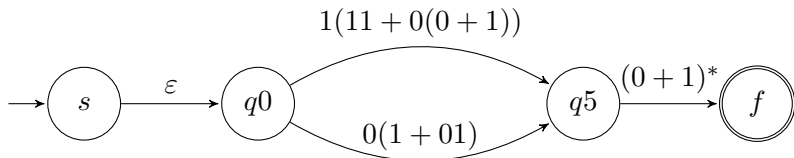
# Ejemplo: BSL (4-AQA)

De AFs a ERs



# Ejemplo: BSL (4-AQA)

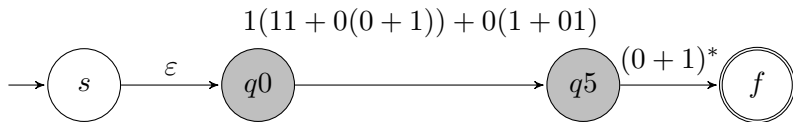
De AFs a ERs





# Ejemplo: BSL (4-AQA)

De AFs a ERs



## Ejemplo: BSL (4-AQA)

De AFs a ERs

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

# Paso a paso

## Conversión de ERs a AFs

Para convertir de una ER a un AF hay que seguir el proceso inverso, el cual parte del mismo principio:

- 1 Empezamos con un autómata *genérico* con un estado inicial y uno final, unidos por la ER. A este AF le llamaremos **Gráfica de Transición**.
- 2 *Partimos* la ER en ERs más pequeñas (vía **unión**, por ejemplo).
- 3 Reemplazamos cada ER con un estado-acción.

# Paso a paso

## Conversión de ERs a AFs

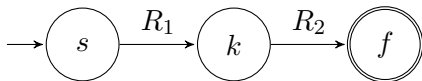
Para convertir de una ER a un AF hay que seguir el proceso inverso, el cual parte del mismo principio:

- 1 Empezamos con un autómata *genérico* con un estado inicial y uno final, unidos por la ER. A este AF le llamaremos **Gráfica de Transición**.
- 2 *Partimos* la ER en ERs más pequeñas (vía **unión**, por ejemplo).
- 3 Reemplazamos cada ER con un estado-acción.

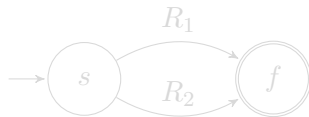
# Generalizando

Conversión entre ERs y AFs

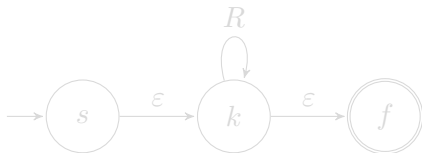
$$R_1 R_2$$



$$R_1 + R_2$$



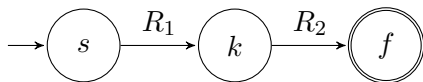
$$R^*$$



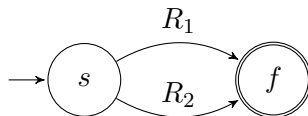
# Generalizando

Conversión entre ERs y AFs

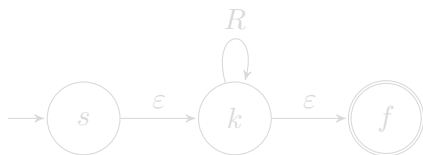
$$R_1 R_2$$



$$R_1 + R_2$$



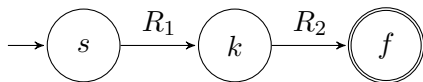
$$R^*$$



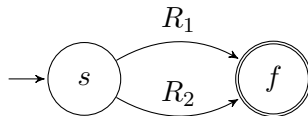
# Generalizando

Conversión entre ERs y AFs

$$R_1 R_2$$



$$R_1 + R_2$$



$$R^*$$

