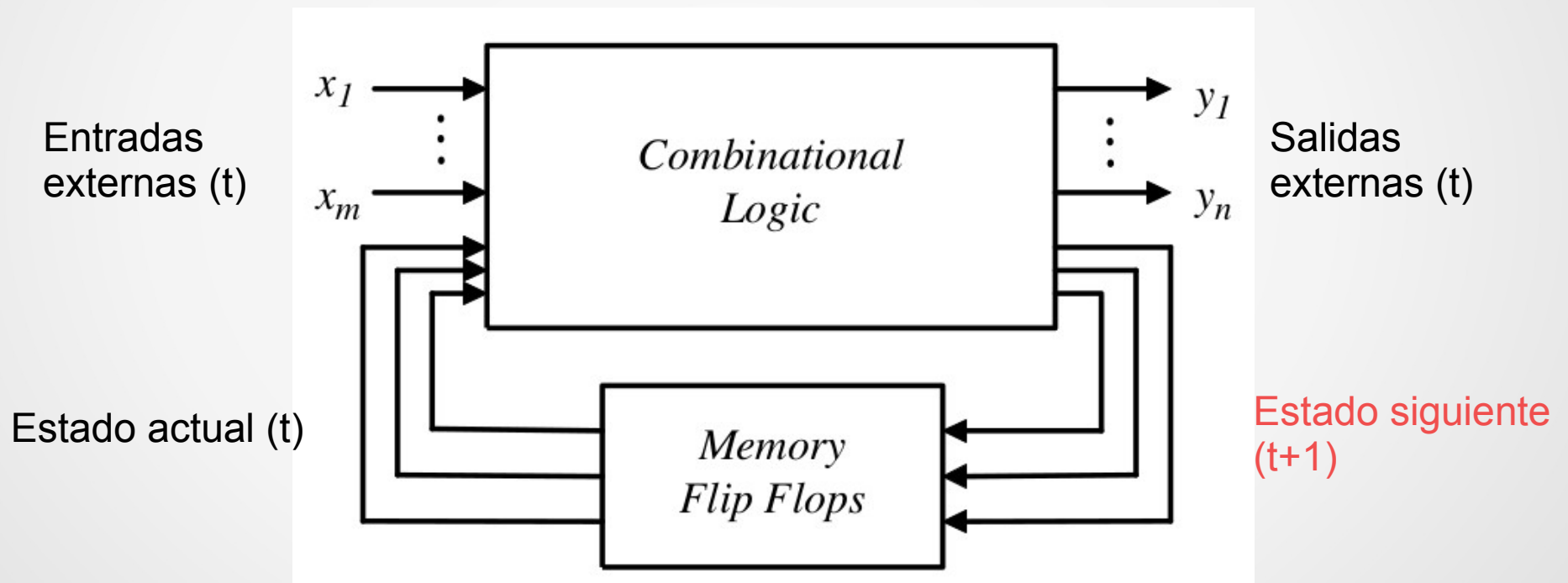




# Autómatas finitos

# Autómatas finitos: revisando sistemas secuenciales



## Autómatas finitos: revisando sistemas secuenciales (2)

- Los circuitos secuenciales vistos en el tema anterior realizaban funciones sencillas tales como registros de almacenamiento y desplazamiento y contadores.
- Estos circuitos eran razonablemente sencillos de diseñar y entender su funcionamiento; sin embargo, los métodos utilizados no son los adecuados para realizarlo cuando se trata de un circuito o sistema digital complejo.
- Los diagramas temporales no son adecuados para diseñar o comprender la evolución de estados de un circuito secuencial.
- Un circuito secuencial que posee entradas que pueden cambiar su estado actual se denomina *máquina de estados finitos* o *autómata finito*, con la condición que la única memoria del mismo sea la que mantiene el estado.

# Autómatas finitos: definiciones

Formalmente, un autómata finito es una 5-tupla  $(Q, \Sigma, q_0, \delta, F)$  donde:<sup>6</sup>

- $Q$  es un conjunto finito de estados;
- $\Sigma$  es un alfabeto finito;
- $q_0 \in Q$  es el estado inicial;
- $\delta: Q \times \Sigma \rightarrow Q$  es una función de transición;
- $F \subseteq Q$  es un conjunto de estados finales o de aceptación.

Para ejemplificar la definición, tomaremos el ejemplo de la máquina de venta de la clase anterior

# Autómatas finitos: máquina de venta

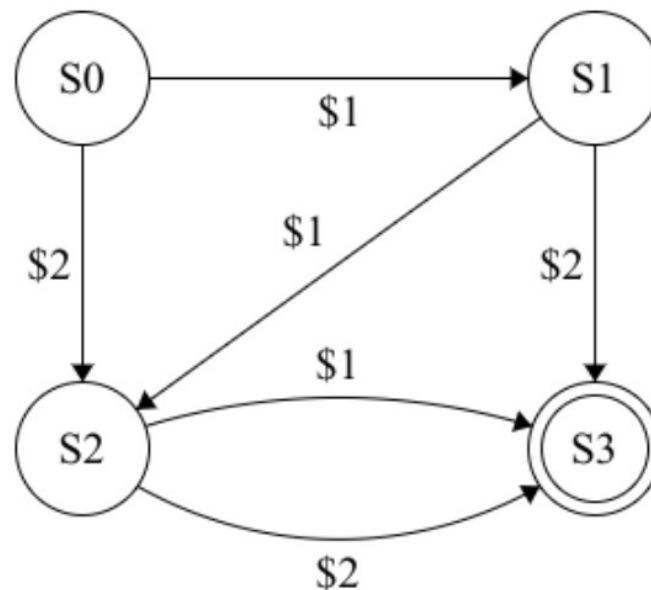
$\{ S0, S1, S2, S3 \}$  conjunto de estados

$\{ \$1, \$2 \}$  alfabeto

S0 estado inicial

Funciones de transición: el conjunto de segmentos dirigidos

$\{ S3 \}$  conjunto de estados finales



# Autómatas finitos: aceptores

- Los autómatas finitos que tienen estados finales se denominan *aceptores* ya que se considera que, si llegan a alguno de esos estados, *aceptan* la cadena de símbolos de entrada que se le han propuesto.
- Si no llegan a esos estados cuando se han consumido toda la cadena de entrada, se consideran que *rechazan* dicha cadena.
- La totalidad de cadenas que aceptan pertenecen a una gramática que, en el caso de autómatas finitos, se llama *gramática regular*.

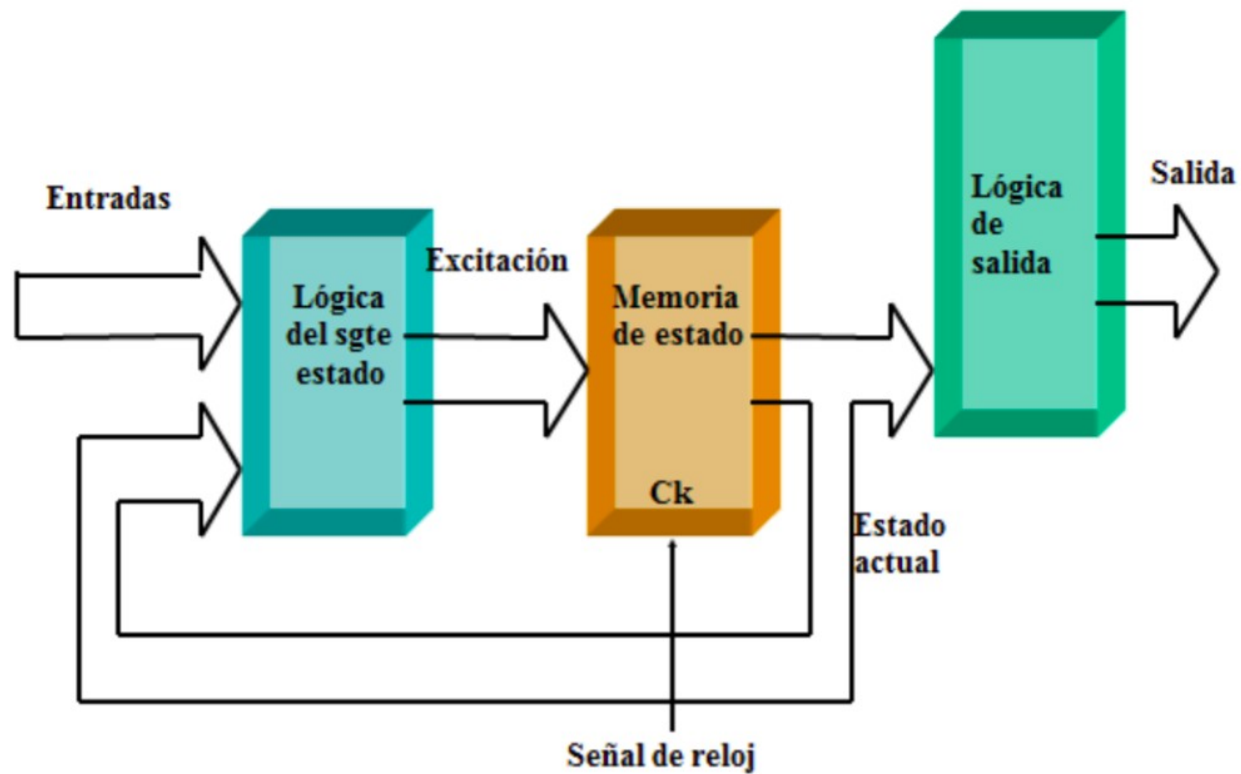
## Autómatas finitos: trasductores de estado finito (FST)

- Los *trasductores* son autómatas finitos que producen distintas salidas así que alcanzan distintos estados
- Estos son los más utilizados en casos prácticos
- Debe aumentarse la definición agregando el alfabeto de salida, es decir el conjunto de símbolos que se presentarán en la salida
- En este caso, existen dos tipos, dependiendo cómo se presenten las salidas correspondientes a cada estado:
  - *Moore*: cuando la salida depende exclusivamente del nuevo estado
  - *Mealy*: cuando la salida depende del estado original y del símbolo de entrada que produce la transición

# Autómatas finitos: Moore

## Máquina de estado de Moore

### Diagrama general





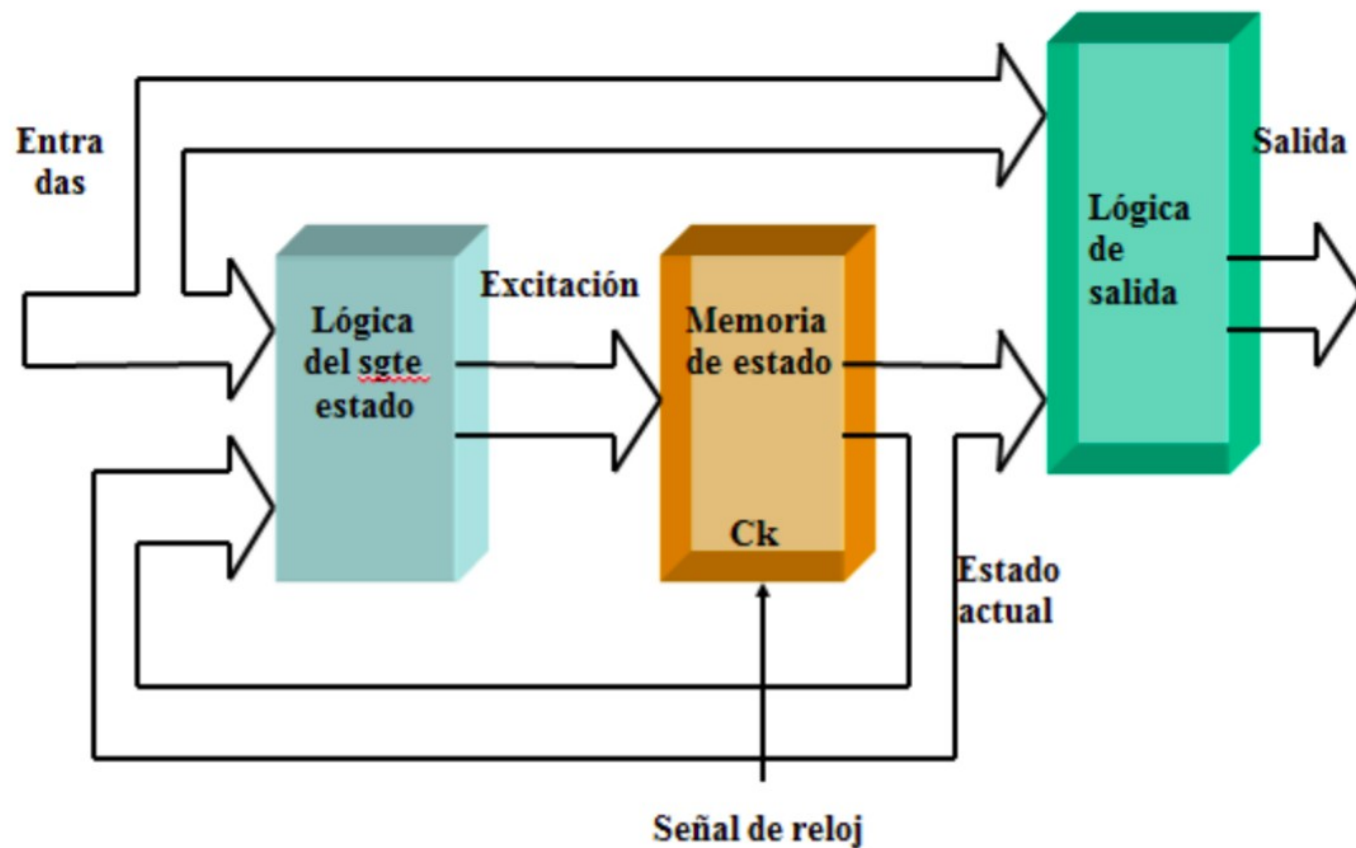
# Autómatas finitos: características de Moore

- La salida depende exclusivamente del estado actual
- Es decir, la salida no depende de las entradas actuales pero, más bien, de las entradas previas.
- La nueva salida debe esperar hasta que el estado es actualizado mediante la validación del *clock*.
- No hay conexión directa entre las entradas y las salidas.
- Si bien desde el punto de vista lógico, *Moore* y *Mealy* son modelos equivalentes (es decir, puede implementarse el transductor necesario mediante cualquiera de ellos), la cantidad de estados de *Moore* respecto de *Mealy* es mayor y como límite puede tener el doble de estados.
- Moore tiene la ventaja que las salidas se obtienen precisamente en los instantes de validación de *clock*.

# Autómatas finitos: Mealy

## Máquina de estado de Mealy

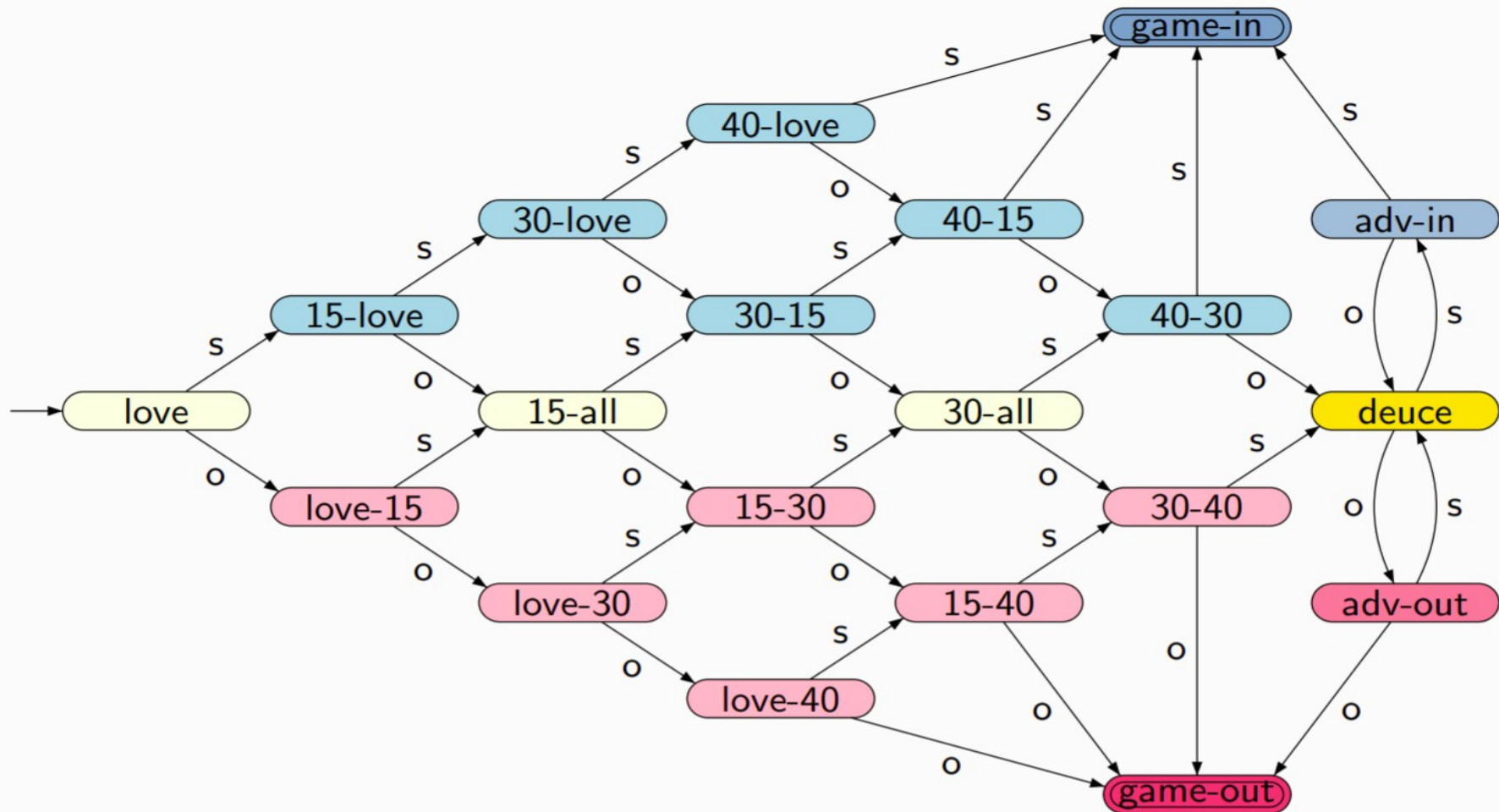
### Diagrama general



# Autómatas finitos: características de Mealy

- La salida depende tanto del estado actual como de las entradas actuales.
- La salida es válida previo a que el *clock* actualice el ciclo corriente.
- Como se dijo en el caso de Moore, Mealy puede tener menos estados comparado con Moore.
- La salida se obtiene antes que en el caso de Moore.

# Automatas finitos: tennis game

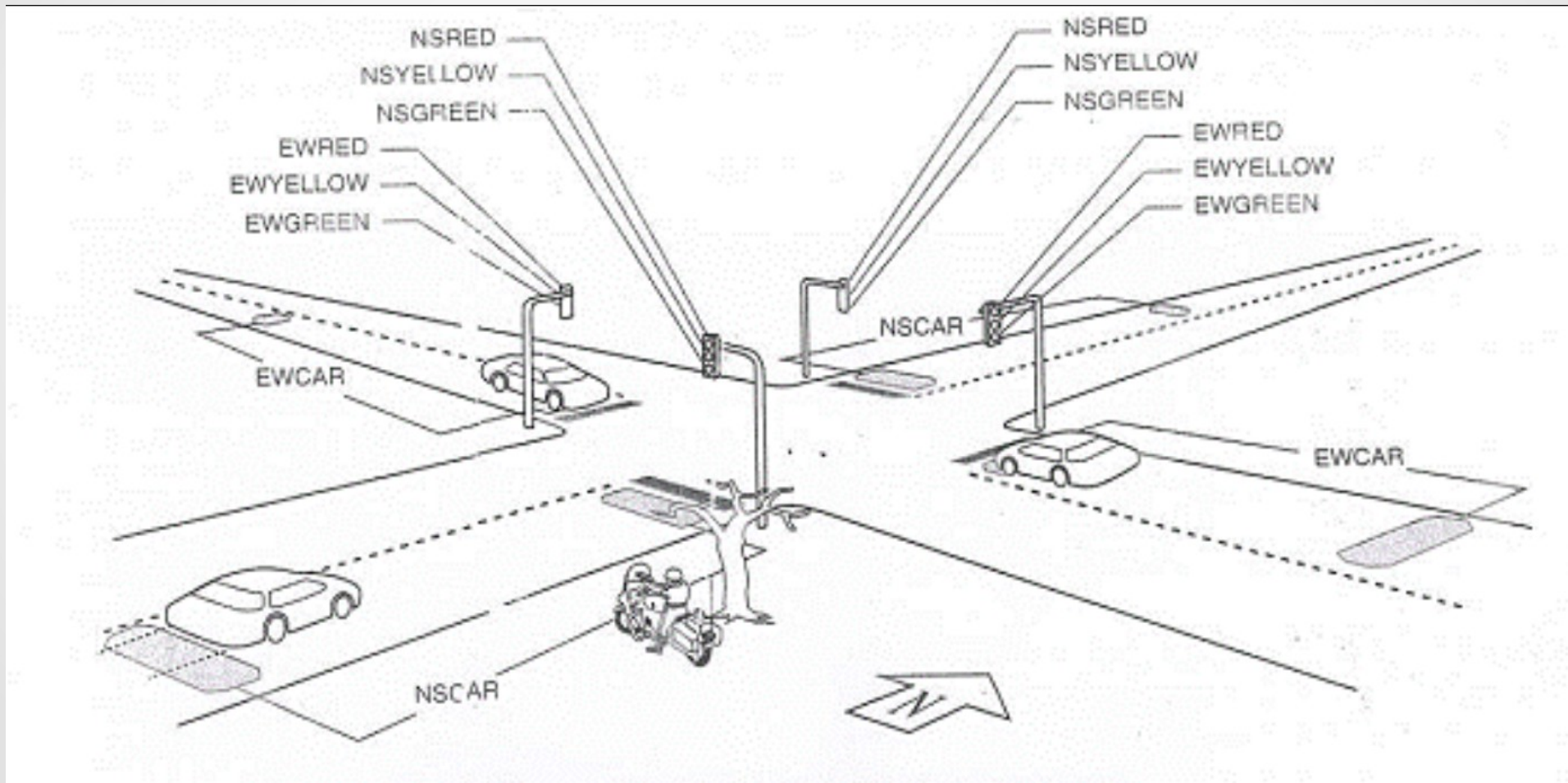


# Autómatas finitos: controladores

- Se denomina *controlador* a un autómata finito que tiene salida (puede ser tipo *Moore* o *Mealy*) pero que a su vez, no posee estados aceptores y que, por lo tanto, funciona en un *loop* continuo
- Es típico su uso para controlar en forma continua un proceso: por ejemplo, el controlador de tráfico que comanda los semáforos de una intersección

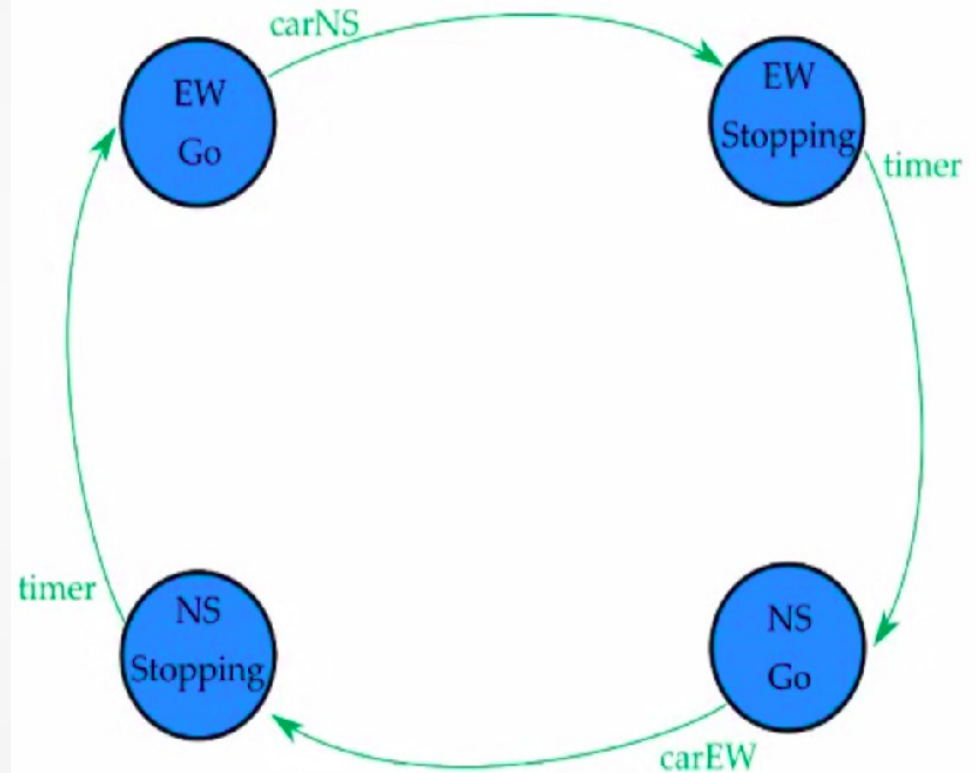


# Autómatas finitos: controlador de tráfico

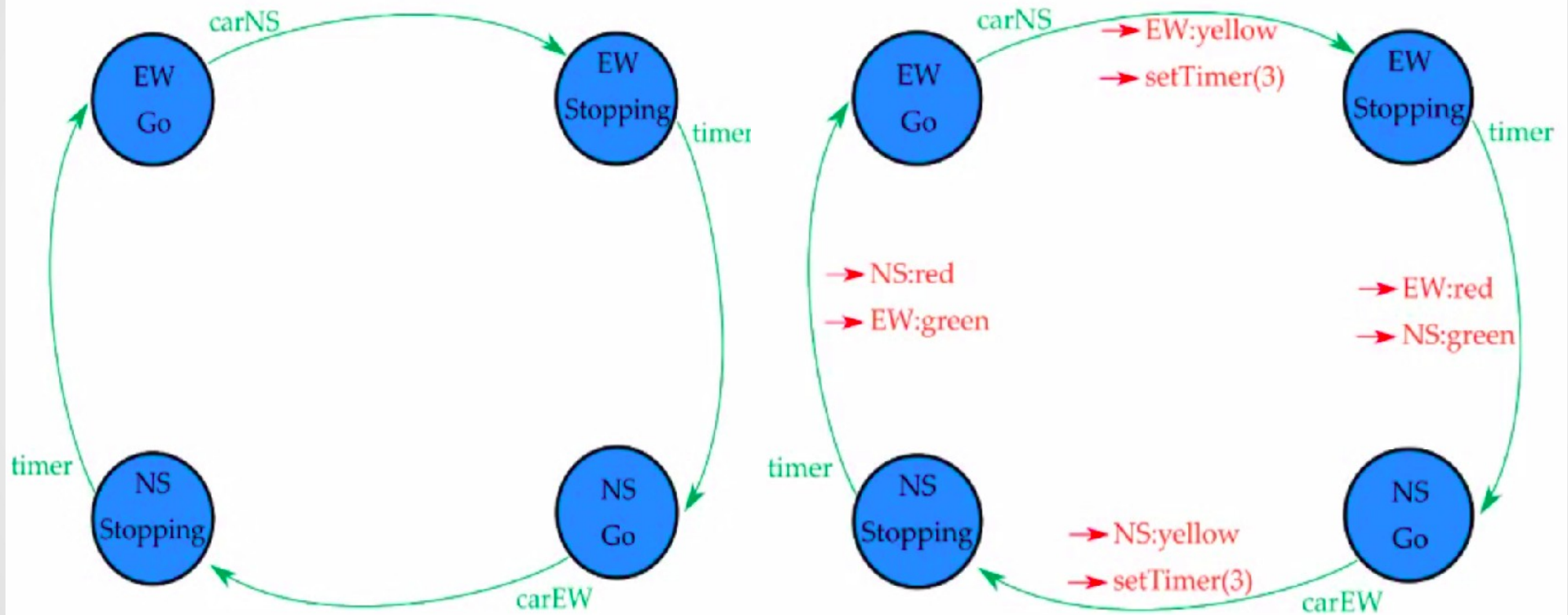




# Autómatas finitos: controlador de tráfico (2)

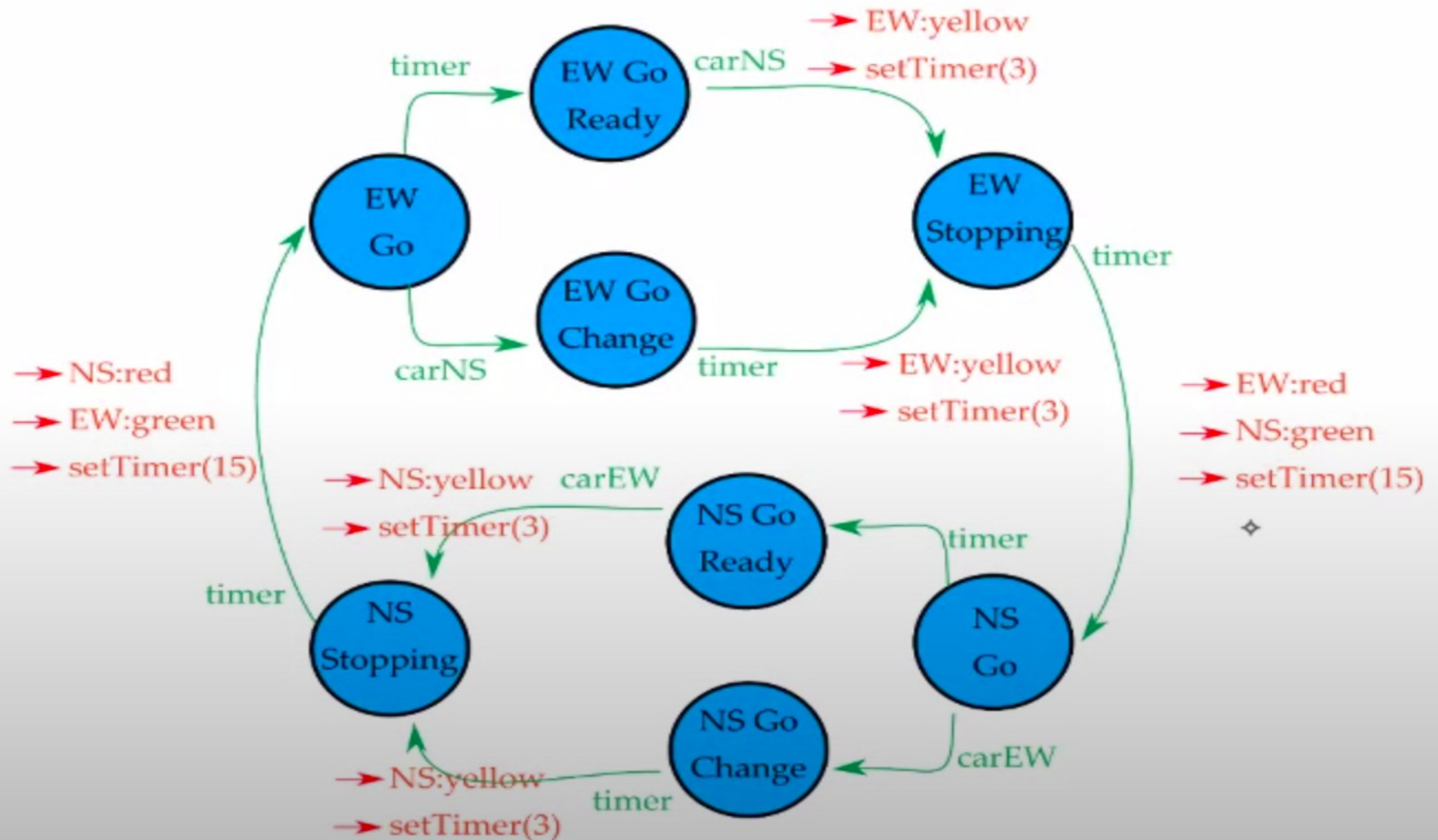


# Autómatas finitos: controlador de tráfico (3)





# Autómatas finitos: controlador de tráfico (4)



# Autómatas finitos: *timer* y controlador

