Lección 2:

Comportamientos Reactivos

Laboratorio de Bio-Robótica

Dr. Jesús Savage Carmona

Facultad de Ingeniería, UNAM

biorobotics.fi-p.unam.mx

Trabajo realizado con el apoyo del Programa

UNAM-DGAPA-PAPIME PE100821

CONTENIDO

- Introducción
- Lógica de Orden Cero
- Máquinas de Estados Finitas (FSM)

Comportamientos Reactivos

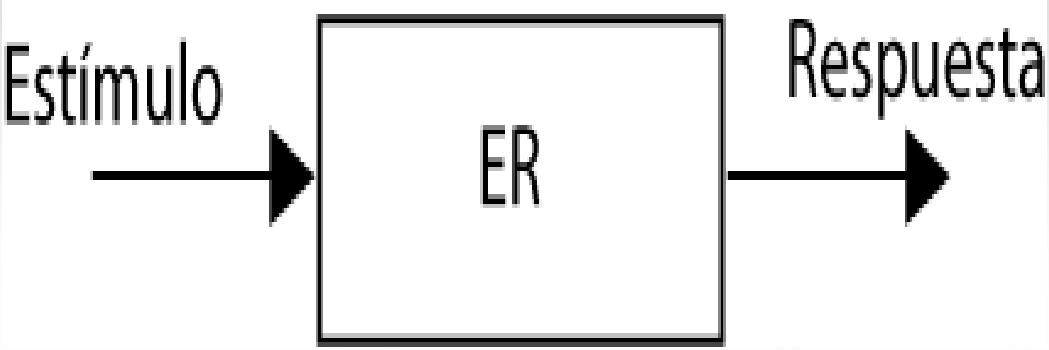
Características:

- Basado en el comportamiento de los insectos.
- No es necesaria una representación del medio ambiente.
- No utiliza planeación de acciones ni de movimientos.
- Es adecuado para entornos dinámicos y con errores en el sensado.
- Esta basado en comportamientos funcionando en paralelo.

Comportamientos Reactivos

Características:

Los comportamientos se representan usando diagramas estimulo- respuesta o ER.

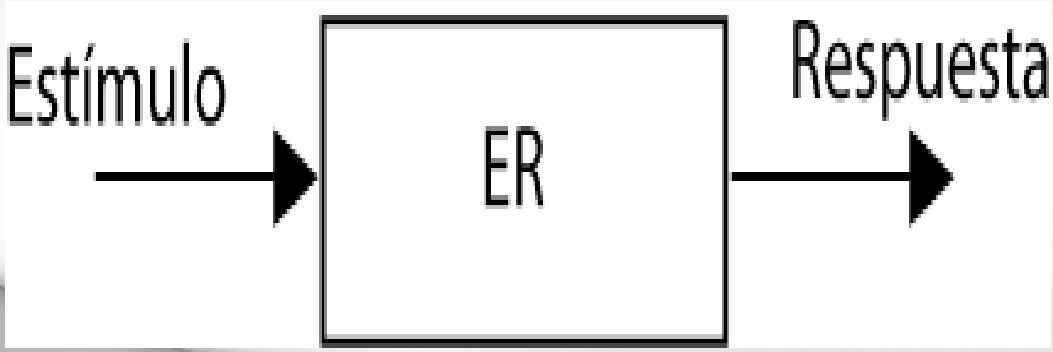


La salida de cada comportamiento debe ser instantánea a partir del momento que hay una entrada.
Los comportamientos son independientes entre si.

Comportamientos Reactivos

Características:

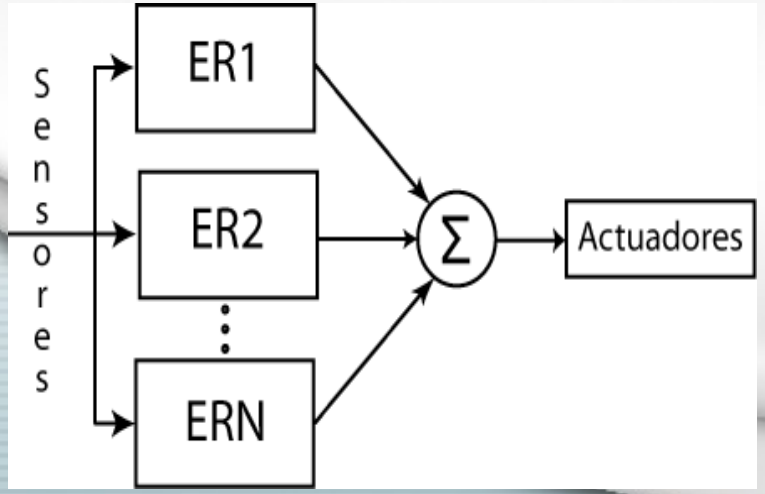
Los comportamientos se pueden diseñar usando, lógica de orden cero, máquinas de estados, campos potenciales, redes neuronales, etc.

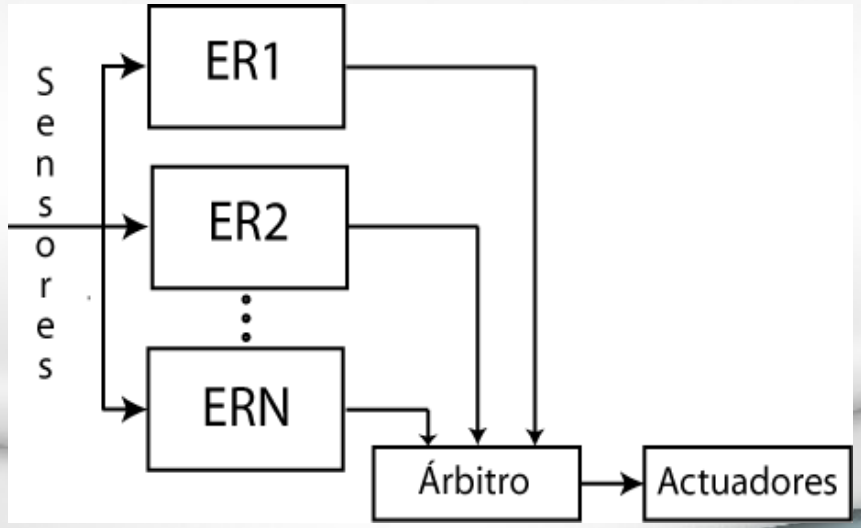


Comportamientos Reactivos

Características:

Los ER se pueden combinar en diferentes estructuras conectandolos en paralelo sumando la salida de cada uno de ellos o seleccionando una de las salidas utilizando un arbitro.

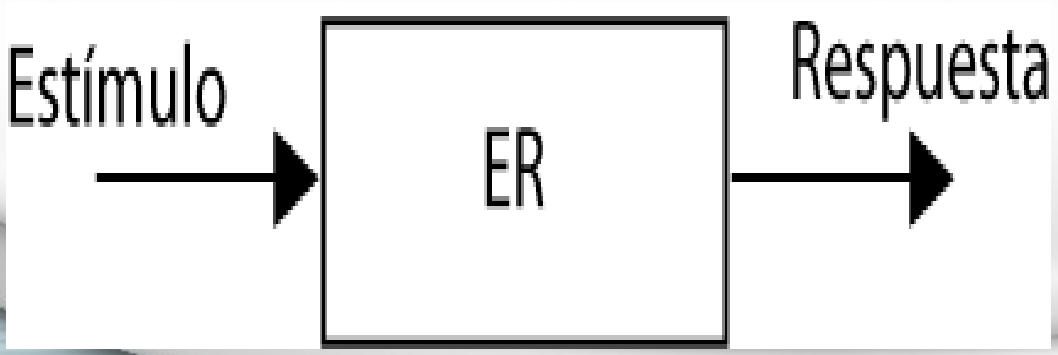




Lógica de Orden Cero

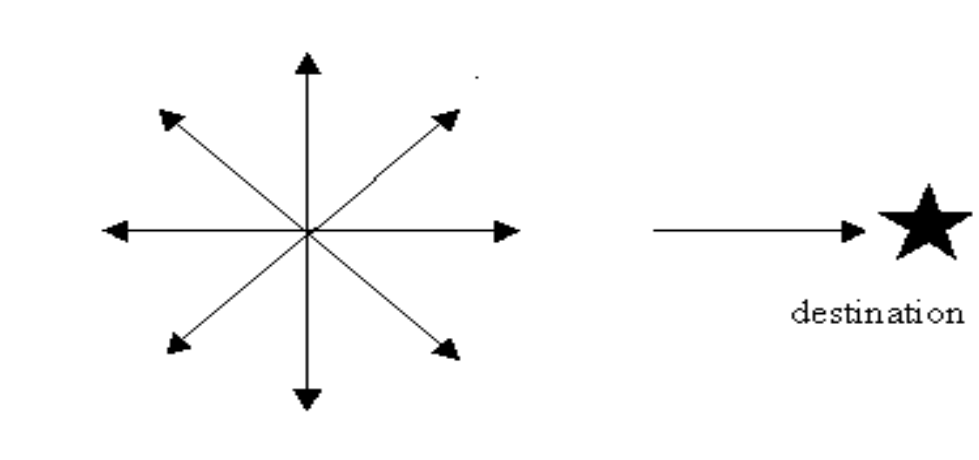
Características:

Se revisan valores de los sensores de entrada y si éstos cumplen con cierta propiedad se genera una salida, la cual dura un tiempo determinado. Estos comportamientos no tienen memoria.



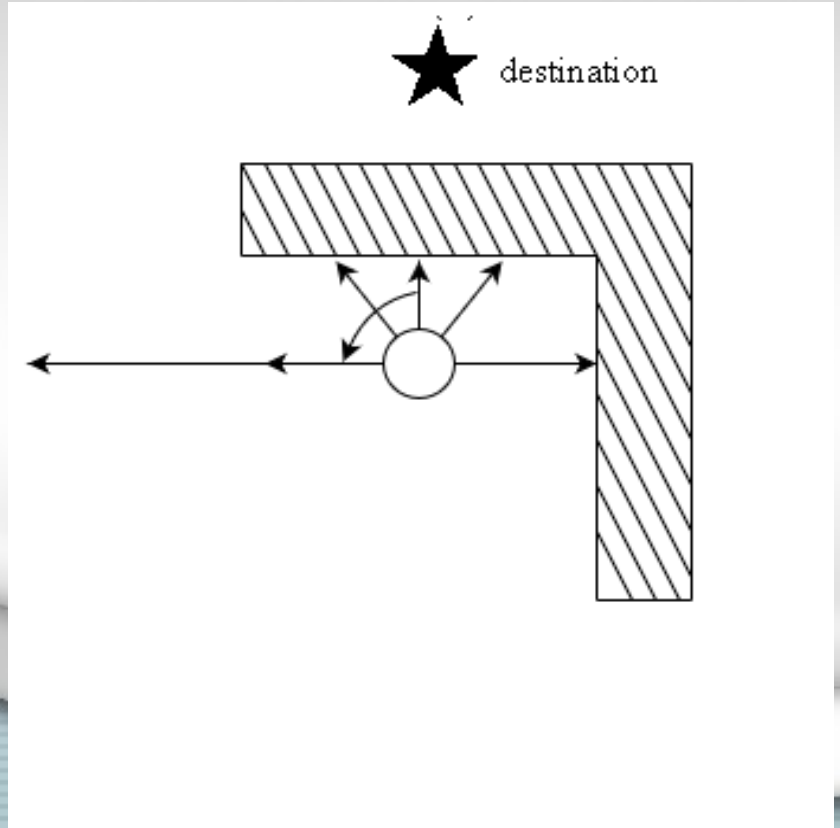
Lógica de Orden Cero

Ejemplo: Si el destino, una fuente luminosa, esta a enfrente del robot y no hay obstáculos entonces éste debera avanzar hacia adelante.



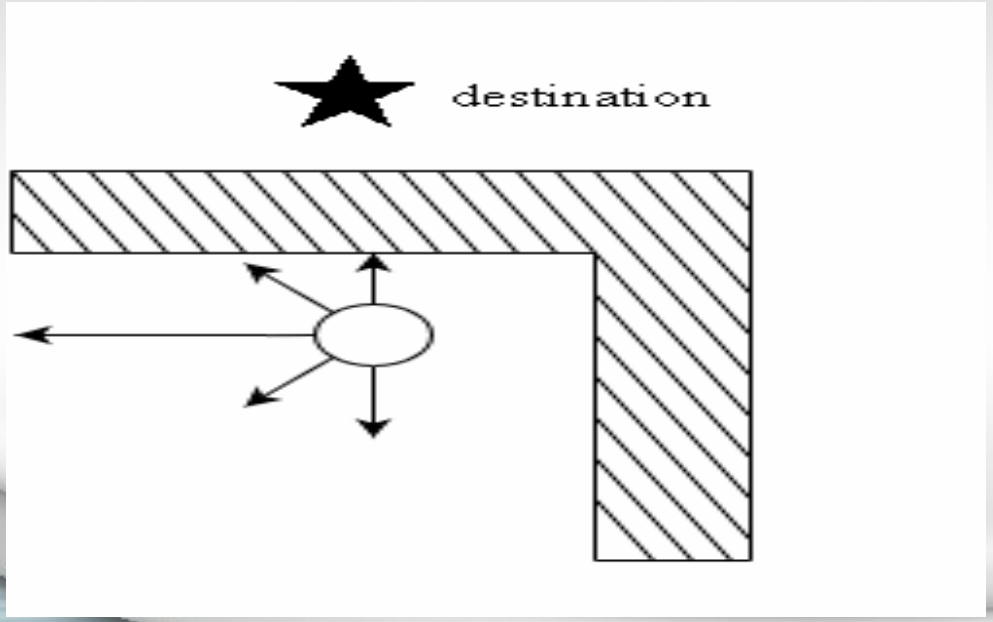
Lógica de Orden Cero

Si la fuente luminosa esta arriba del robot y hay un obstáculo enfrente y al lado derecho, entonces el robot gira a la izquierda.



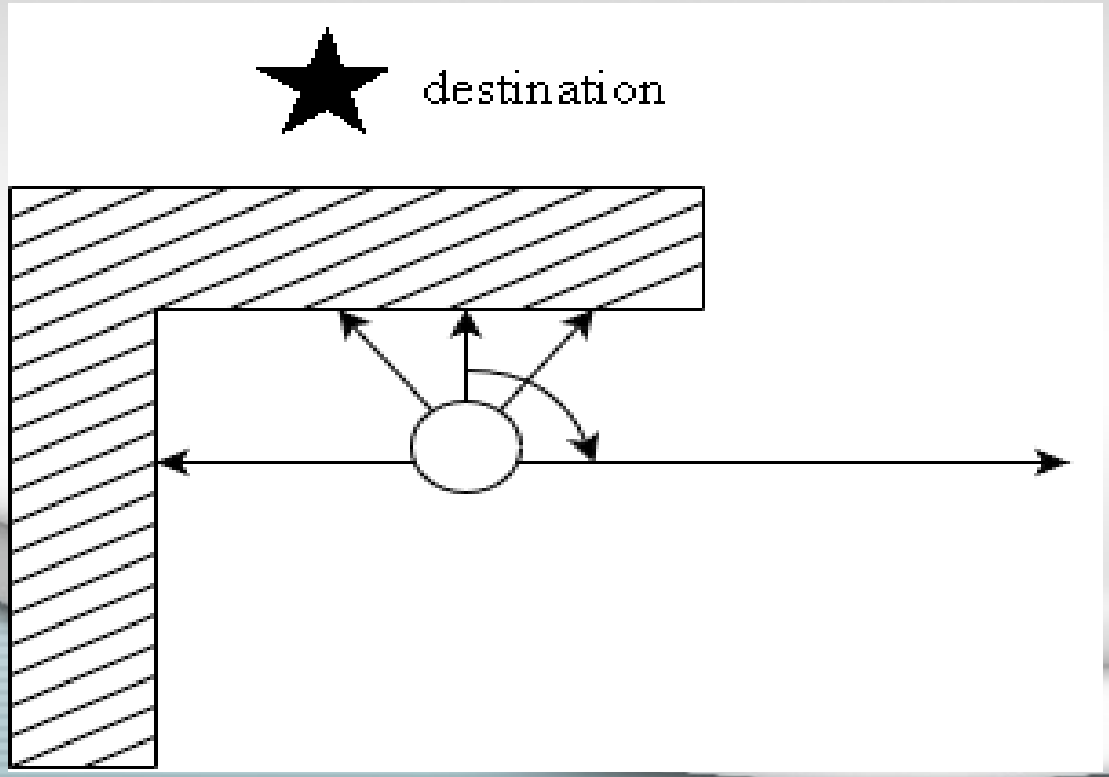
Lógica de Orden Cero

Si la fuente luminosa esta a la derecha del robot y hay un obstáculo a la derecha y atras de éste, entonces el robot avanzara hacia adelante.



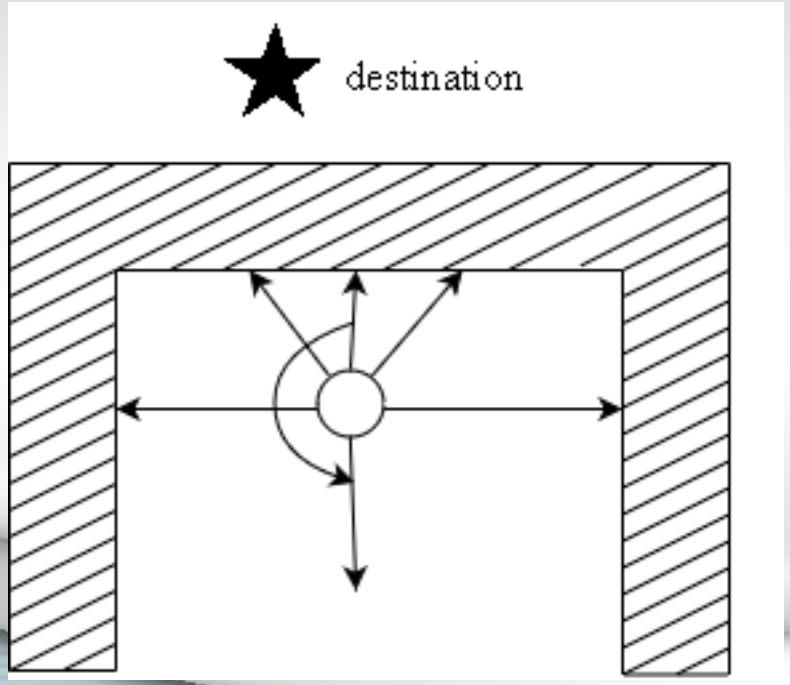
Lógica de Orden Cero

Si la fuente luminosa esta arriba del robot y hay un obstáculo enfrente y al lado izquierdo, entonces el robot gira a la derecha.



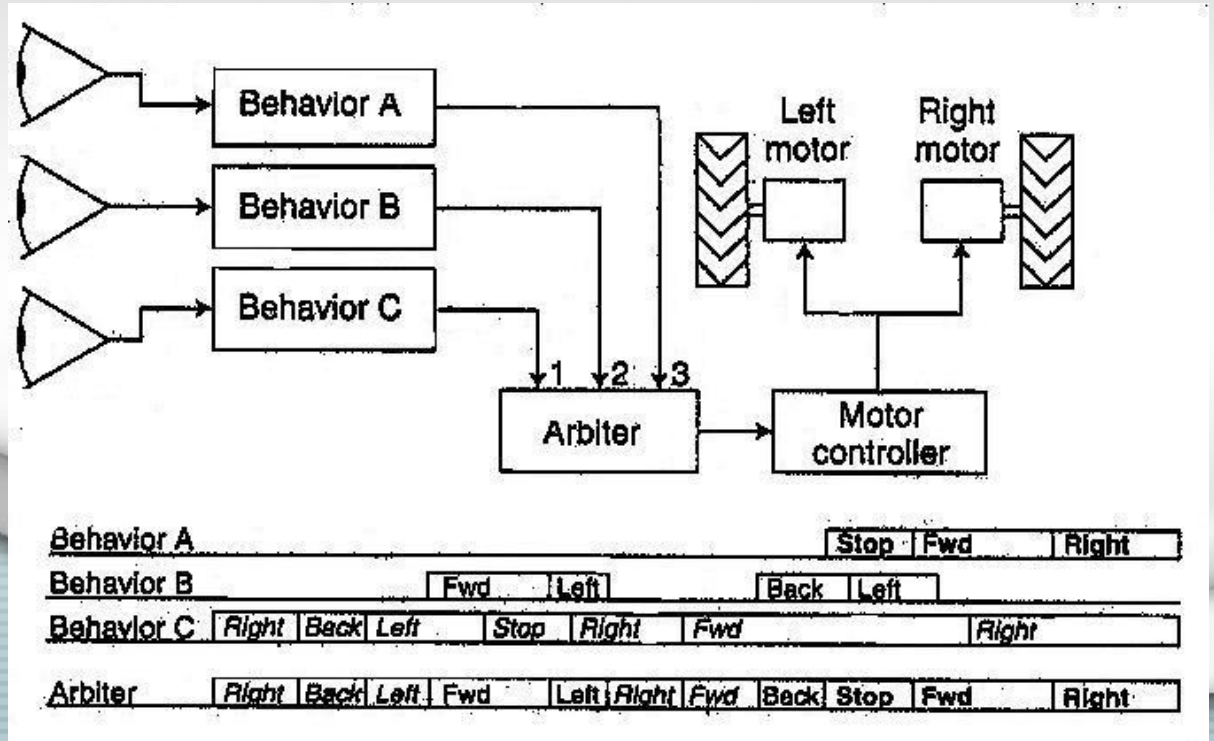
Lógica de Orden Cero

Si la fuente luminosa esta arriba del robot y hay un obstáculo enfrente y al lado izquierdo y derecho entonces el robot gira 180° y avanza hacia adelante.



Lógica de Orden Cero

Se pueden entonces tener varios comportamientos en paralelo en forma jerarquica y un arbitro decide, dependiendo de la jerarquía, cual es la salida del comportamiento que se enviara a los actuadores del robot.



Máquinas de Estado Finitas

Una máquina estados ejecuta algoritmos, éstos se representan por medio de estados, cada estado tiene una duración de **T** segundos. Para un tiempo determinado **t**, el estado siguiente y su salidas, en el tiempo **t+1**, dependen del estado presente y de sus entradas. A diferencia de los sistemas de lógica de primer orden las máquinas de estado si tienen memoria.

Entradas

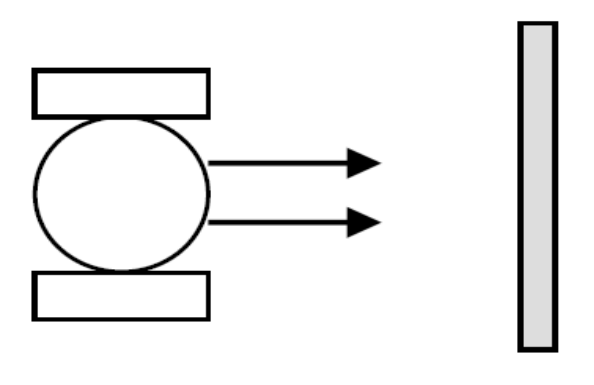
Salidas

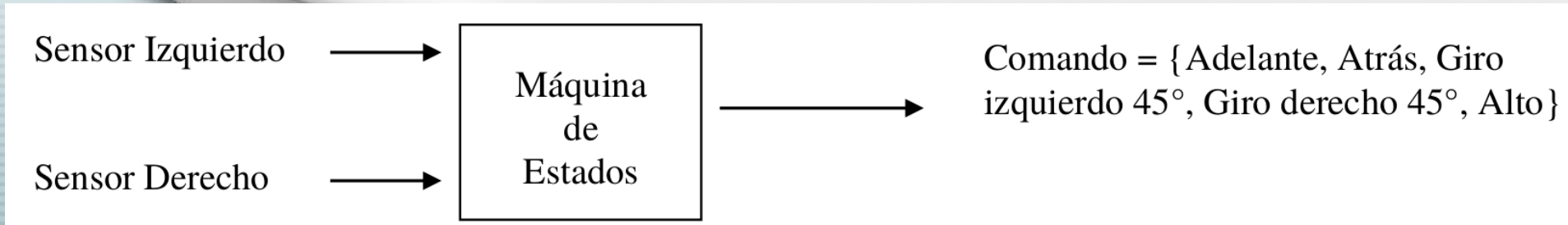
Máquinas de Estado Finitas

Comportamiento de un Robot para Evadir Obstáculos

Se cuenta con un robot móvil con dos motores, uno en el lado izquierdo y uno en el derecho.

Dos sensores (por ejemplo, infrarrojos, ultrasonido, laser, etc.) colocados en la parte delantera, para detectar obstáculos.

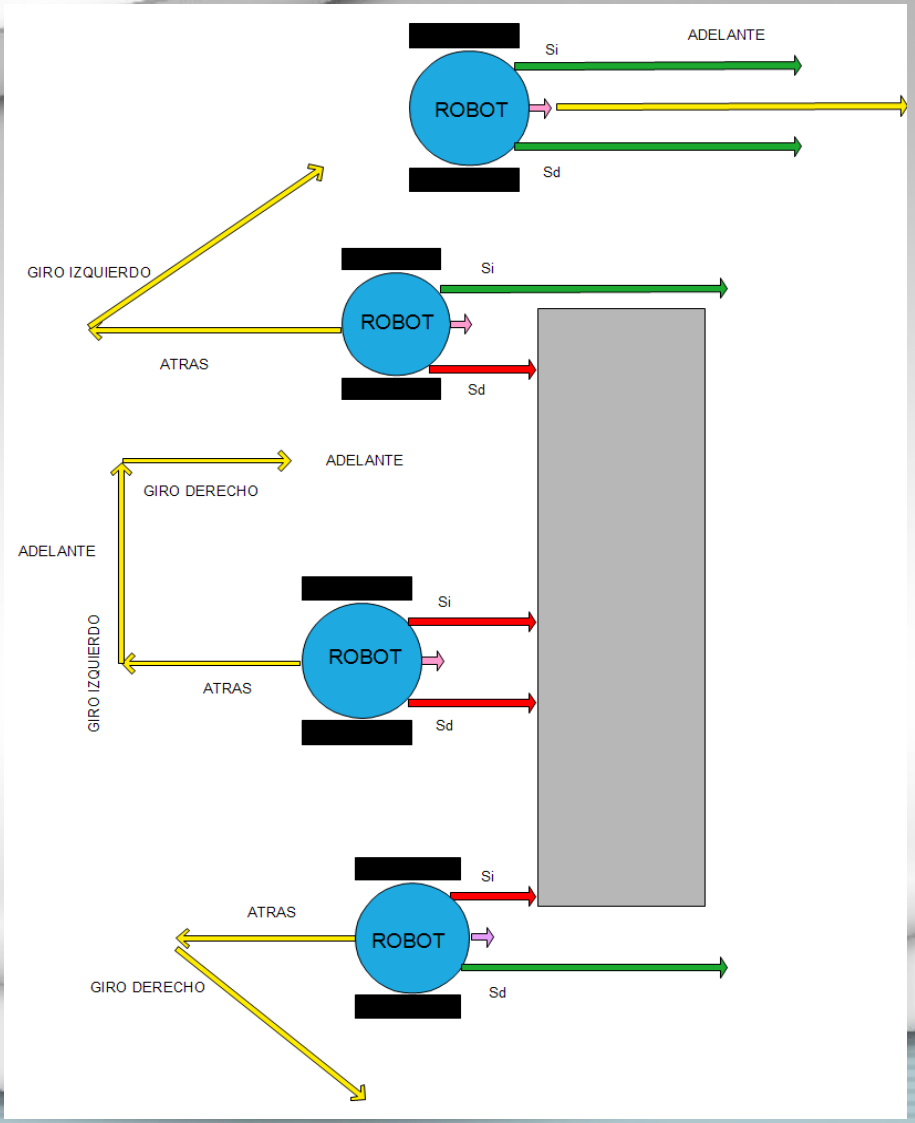




Algoritmo para el Comportamiento de Evadir Obstáculos

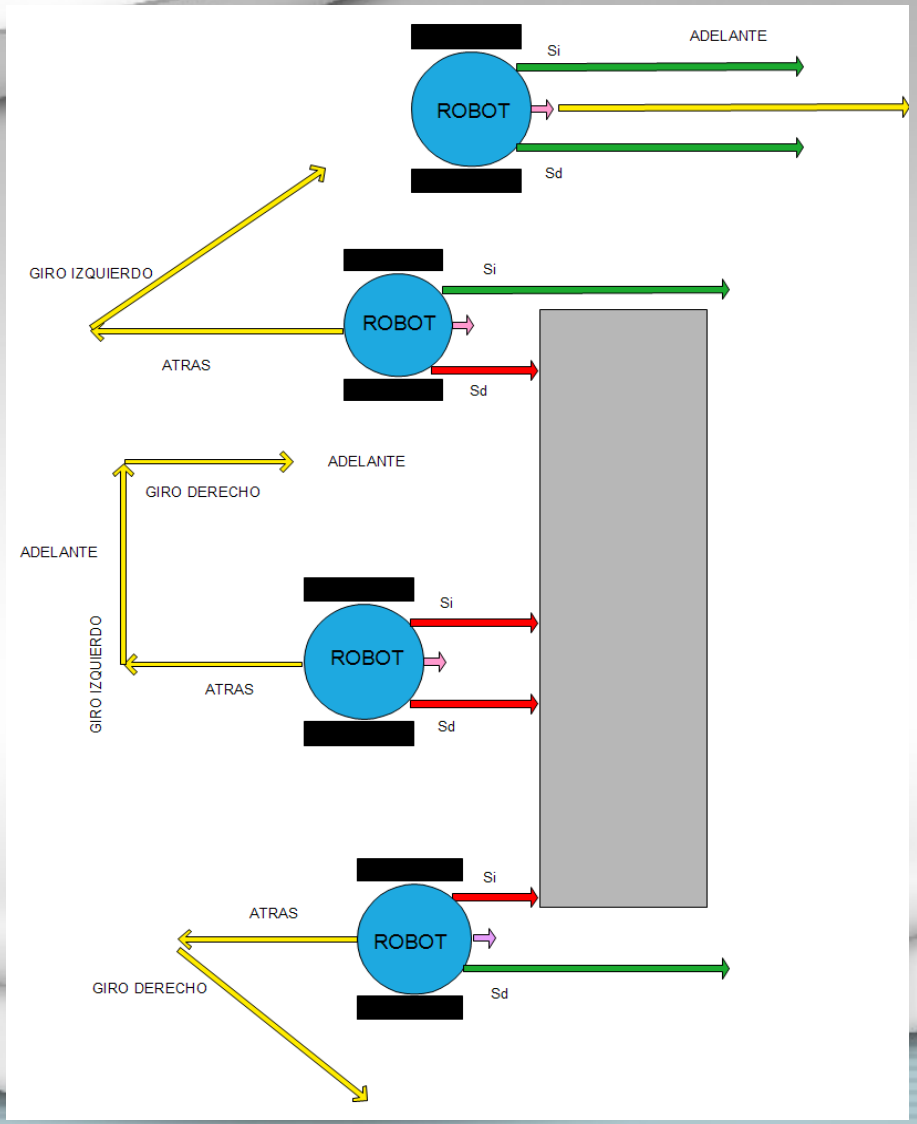
- Si los sensores no detectan un obstáculo, el robot sigue avanzando.

- Si el sensor derecho lo detecta y el izquierdo no, el robot se hace para atrás y después gira hacia la izquierda 45° y sigue avanzando.

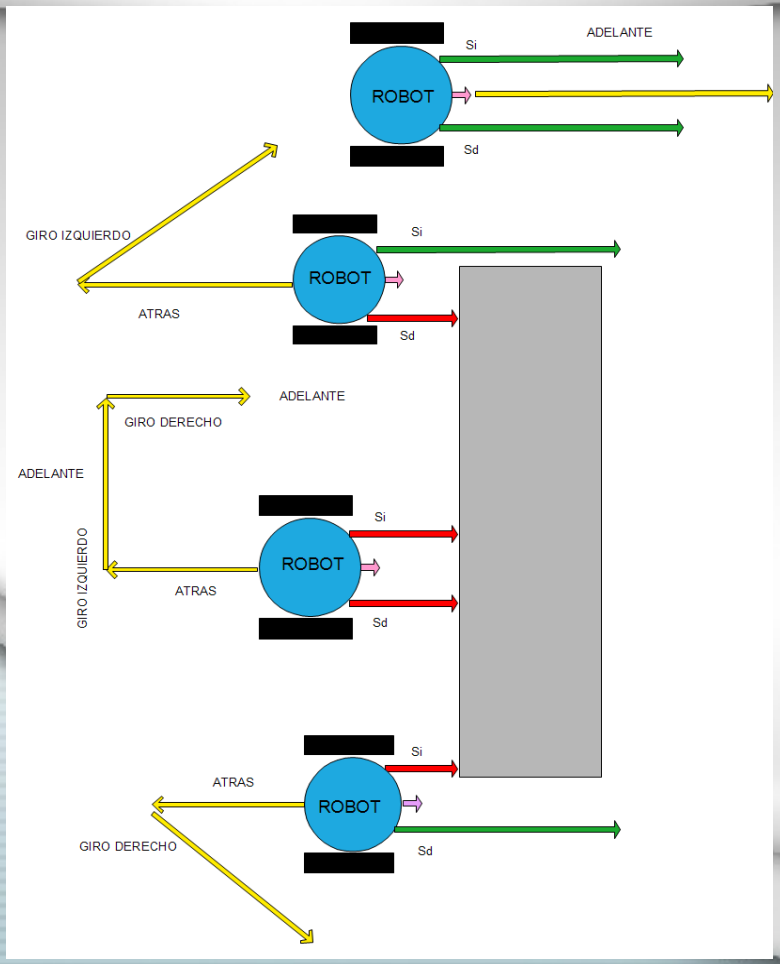


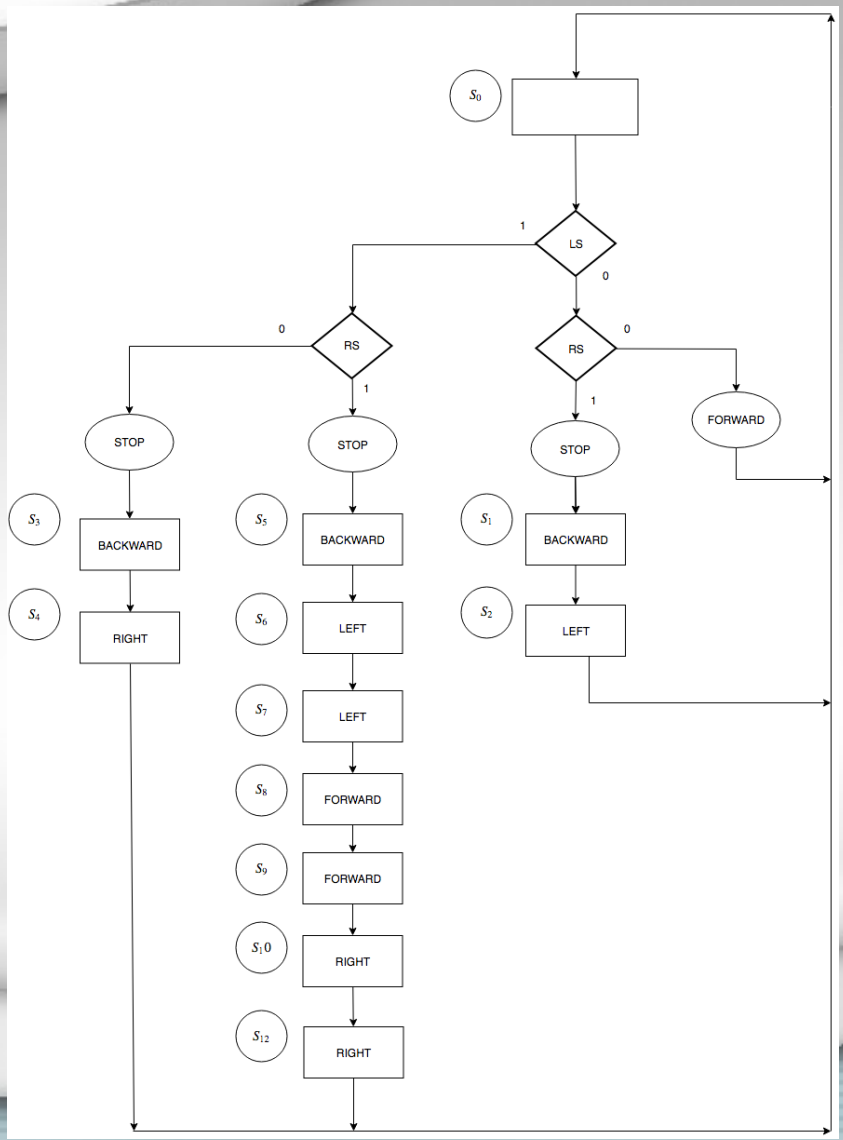
Algoritmo para el Comportamiento de Evadir Obstáculos

- Si el sensor izquierdo lo detecta y el derecho no, el robot se hace para atrás y gira hacia la derecha 45° para seguir avanzando.
- Si los dos sensores detectan el obstaculo, el robot se hace para atrás y gira hacia la izquierda 90°. Después avanza una cierta distancia y gira hacia la derecha 90° y sigue avanzando.

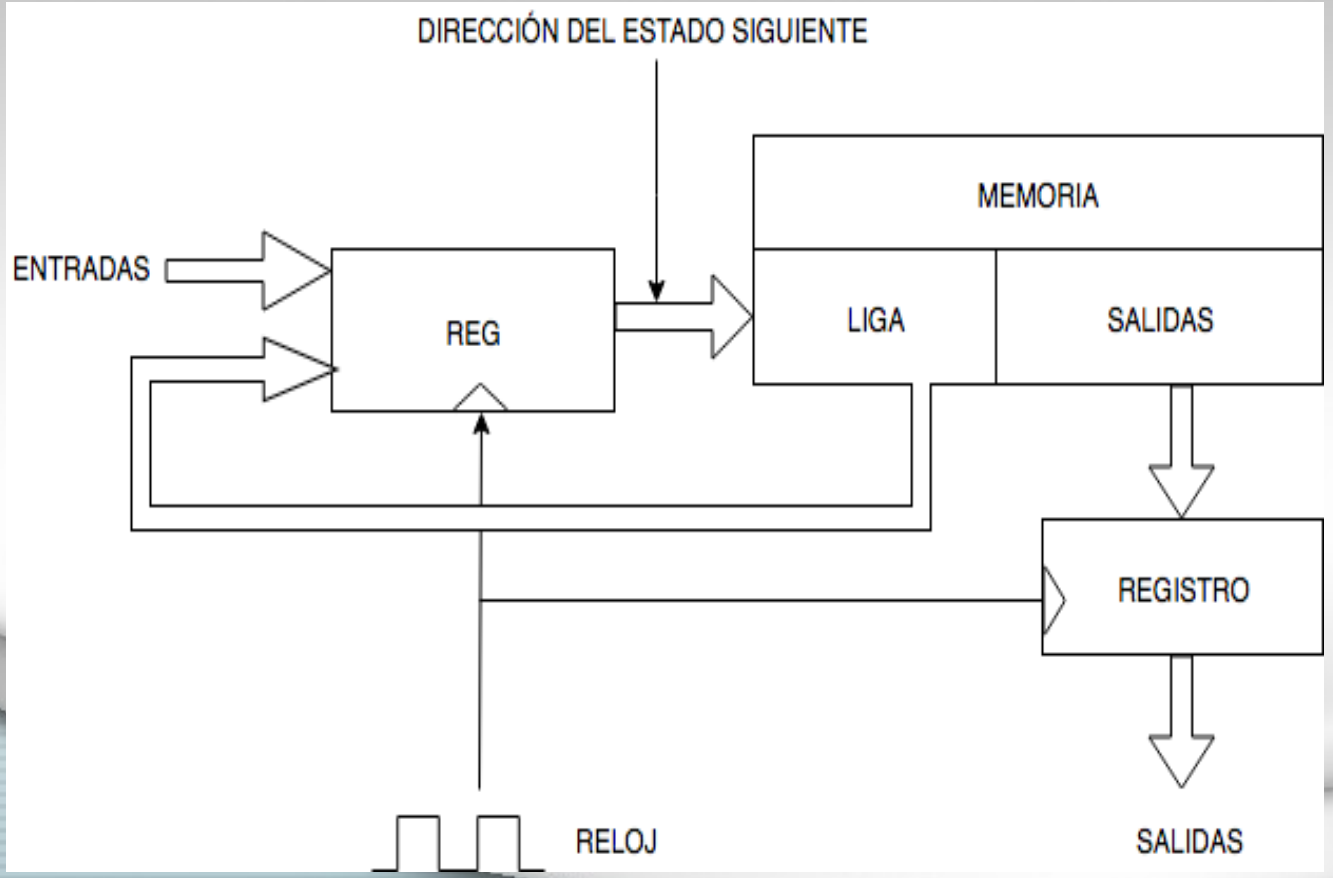


Comportamientos usando algoritmos de máquinas de estados.





Construcción de máquinas de estados usando memorias.

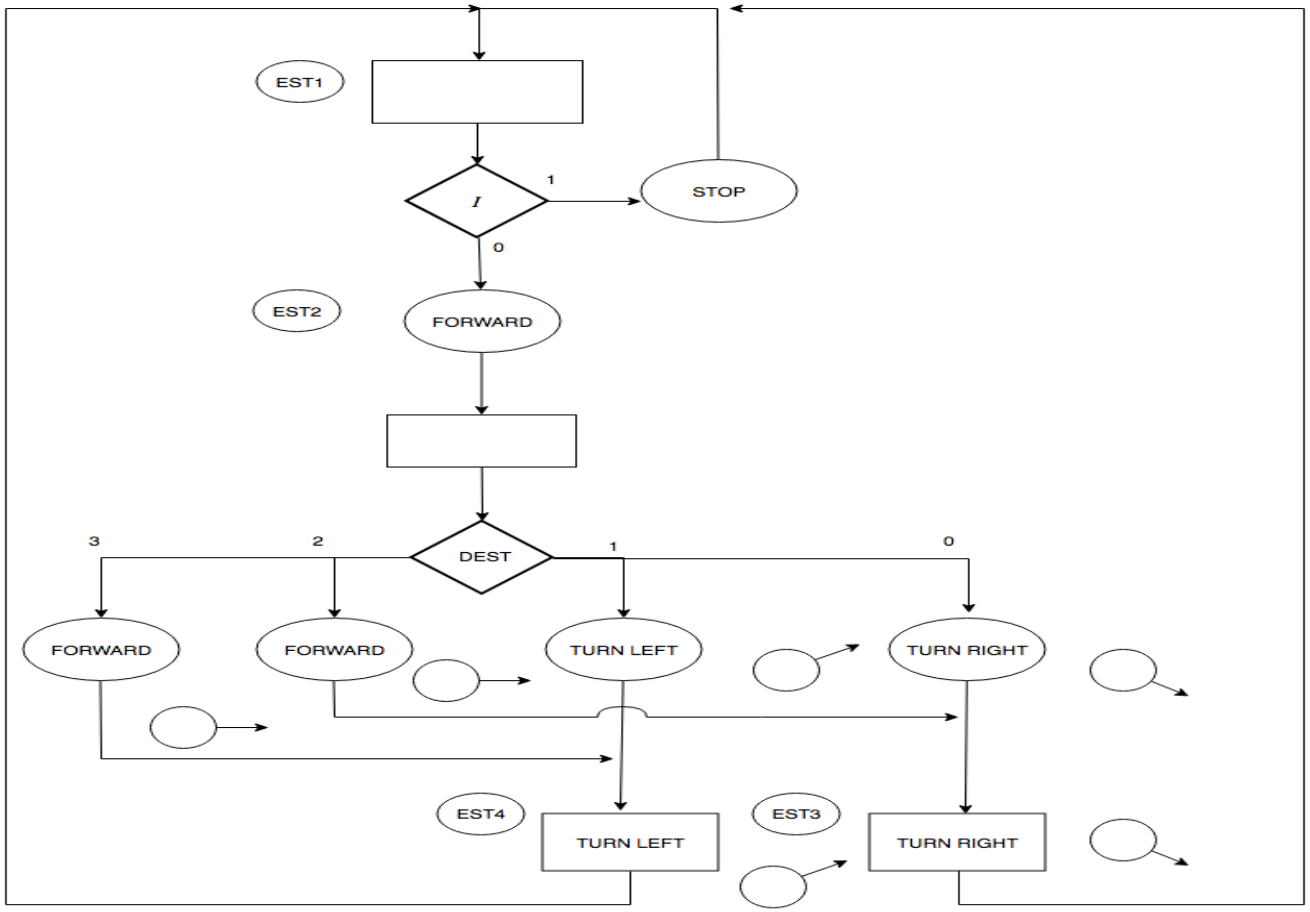


Construcción de máquinas de estados usando memorias.

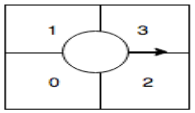
	Dirección de Memoria		Contenido de la Memoria				
	Edo. Presente	Entradas	Liga	Salidas			
	A B C D	$S_i S_d$	A B C D	Stop	Adelante	Atrás	Giro Der	Giro Izq
ST0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 1 1 0 1 1	0 0 0 0 0 0 0 1 0 0 1 1 0 1 0 1	0 1 1 1	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
ST1	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1	0 0 0 1 1 0 1 1	0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0	0 0 0 0	0 0 0 0	1 1 1 1	0 0 0 0	0 0 0 0
ST2	0 0 1 0 . . .	0 0 - - -	0 0 0 0 . . .	0	0	0 . . .	0	1
ST3	0 0 1 1 . . .	0 0 - - -	0 1 0 0 . . .	0	0	1 . . .	0	0
ST4	0 1 0 0 . . .	0 0 - - -	0 0 0 0 . . .	0	0	0 . . .	1	0
ST5	0 1 0 1 . . .	0 0 - - -	0 1 1 0 . . .	0	0	1 . . .	0	0
ST6	0 1 1 0 . . .	0 0 - - -	0 1 1 1 . . .	0	0	0 . . .	0	1
ST7	0 1 1 1 . . .	0 0 - - -	1 0 0 0 . . .	0	0	0 . . .	0	1
ST8	1 0 0 0 . . .	0 0 - - -	1 0 0 1 . . .	0	1	0 . . .	0	0
ST9	1 0 0 1 . . .	0 0 - - -	1 0 1 0 . . .	0	1	0 . . .	0	0
ST10	1 0 1 0 . . .	0 0 - - -	1 0 1 1 . . .	0	0	0 . . .	1	0
ST11	1 0 1 1 . . .	0 0 - - -	0 0 0 0 . . .	0	0	0 . . .	1	0

Algoritmo de un robot que se dirige a una fuente luminosa.

Comportamiento del robot que se dirige al destino



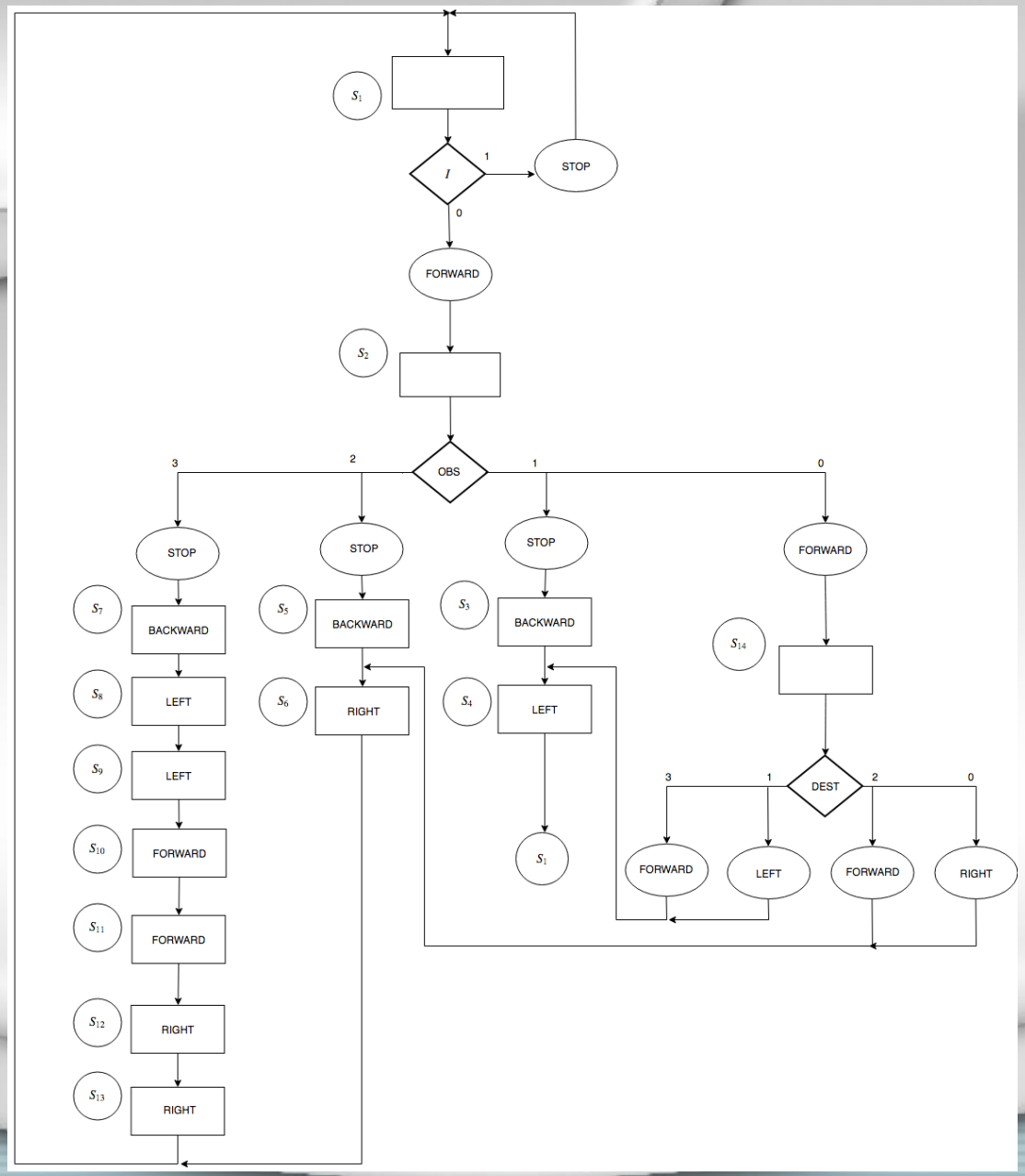
Variable Dest



I = Sensor de intensidad

$$DEST = \begin{cases} 0 & \text{fuente atrás a la derecha} \\ 1 & \text{" " " " izquierda} \\ 2 & \text{" adelante a la derecha} \\ 3 & \text{" " " " izquierda} \end{cases}$$

Algoritmo de un robot que se dirige a una fuente luminosa evadiendo obstáculos.



Máquinas de Estado Finita Aumentada

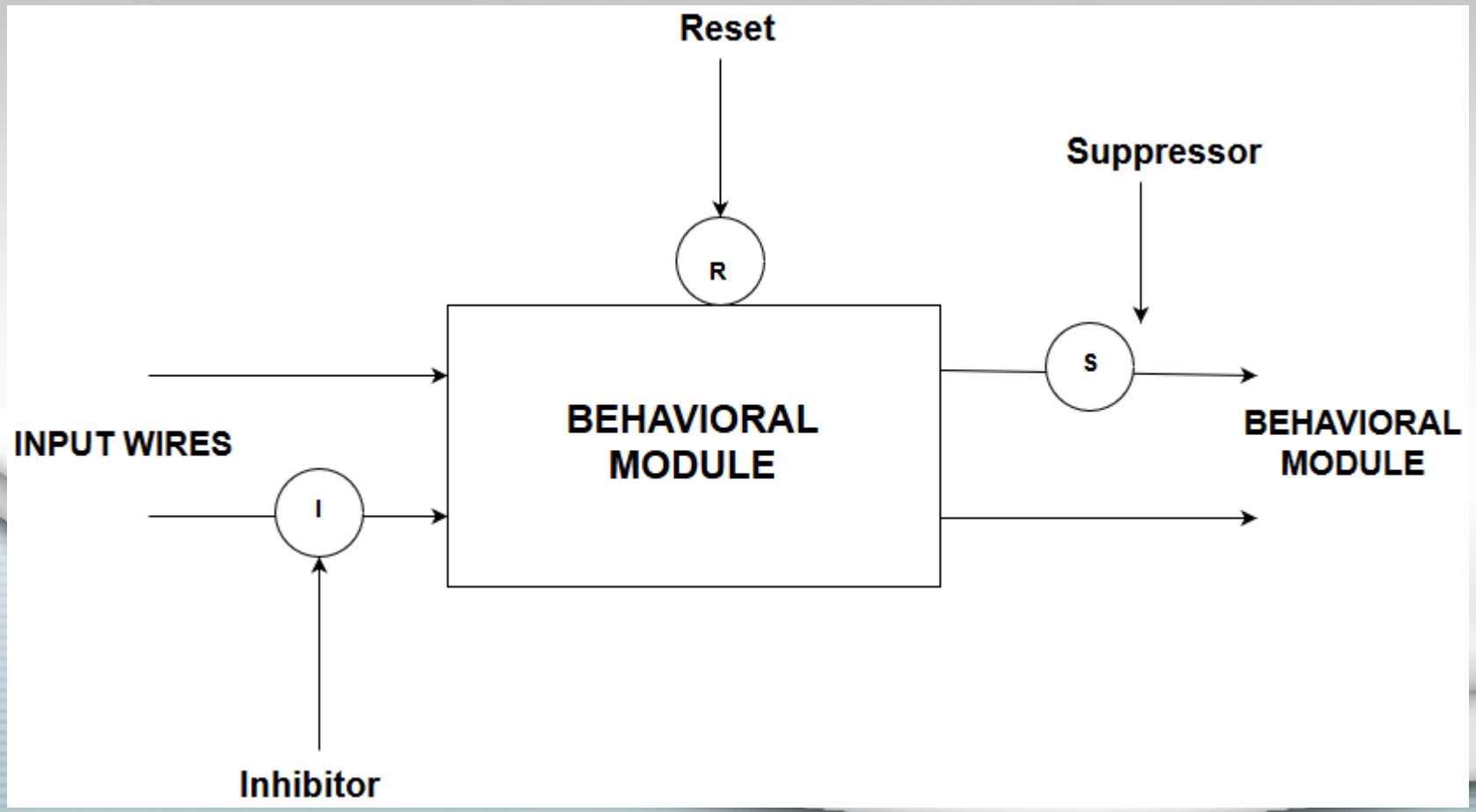
Rodney Brooks (MIT) inventó, en los 80's, la máquina de estados finita aumentada (AFSM, por sus siglas en ingles).

En estas máquinas de estados algunas de sus entradas son inhibidas por otras máquinas de estados. Es decir las salidas de unas máquinas de estados se convierten en entradas de otras.

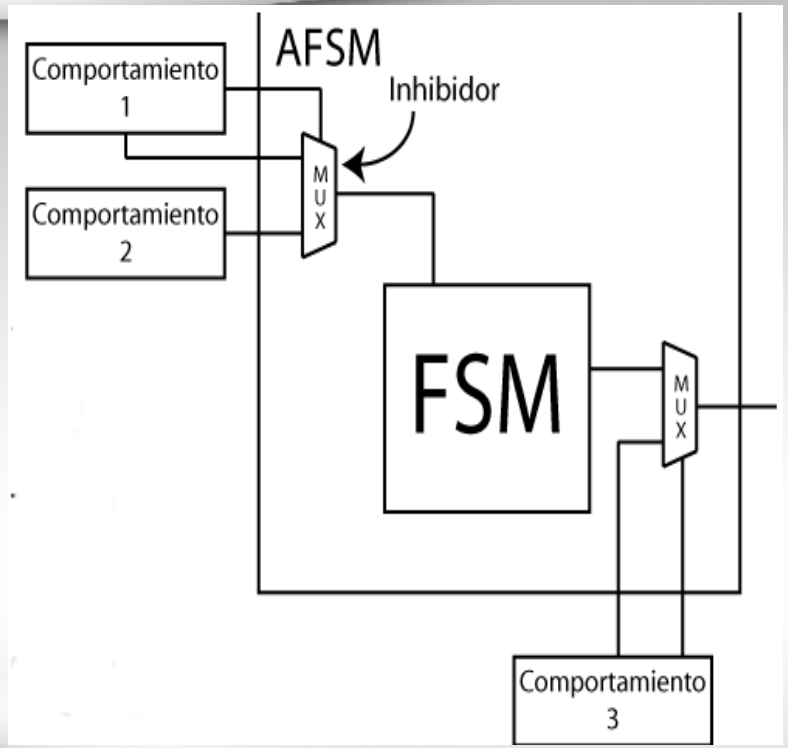
También algunas de las salidas están conectados unos supresores, los cuales bloquean las salidas de la máquina y colocan valores diferentes generadas por otras máquinas de estados.

Máquinas de Estado Finitas Aumentadas

Por último hay una entrada de "reset" que hace que la máquina de estados se coloque en un estado específico.

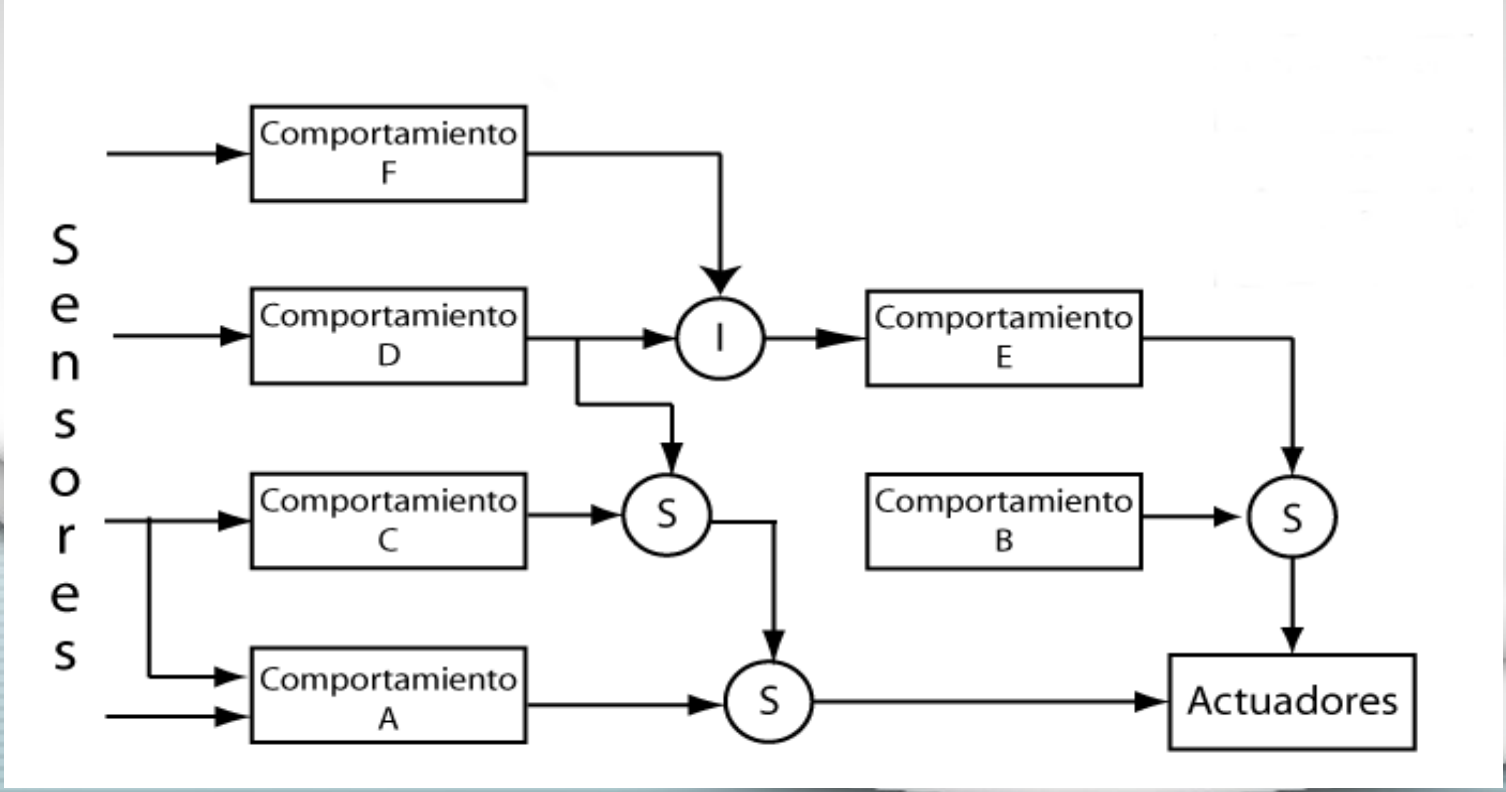


Máquinas de Estado Finitas Aumentadas

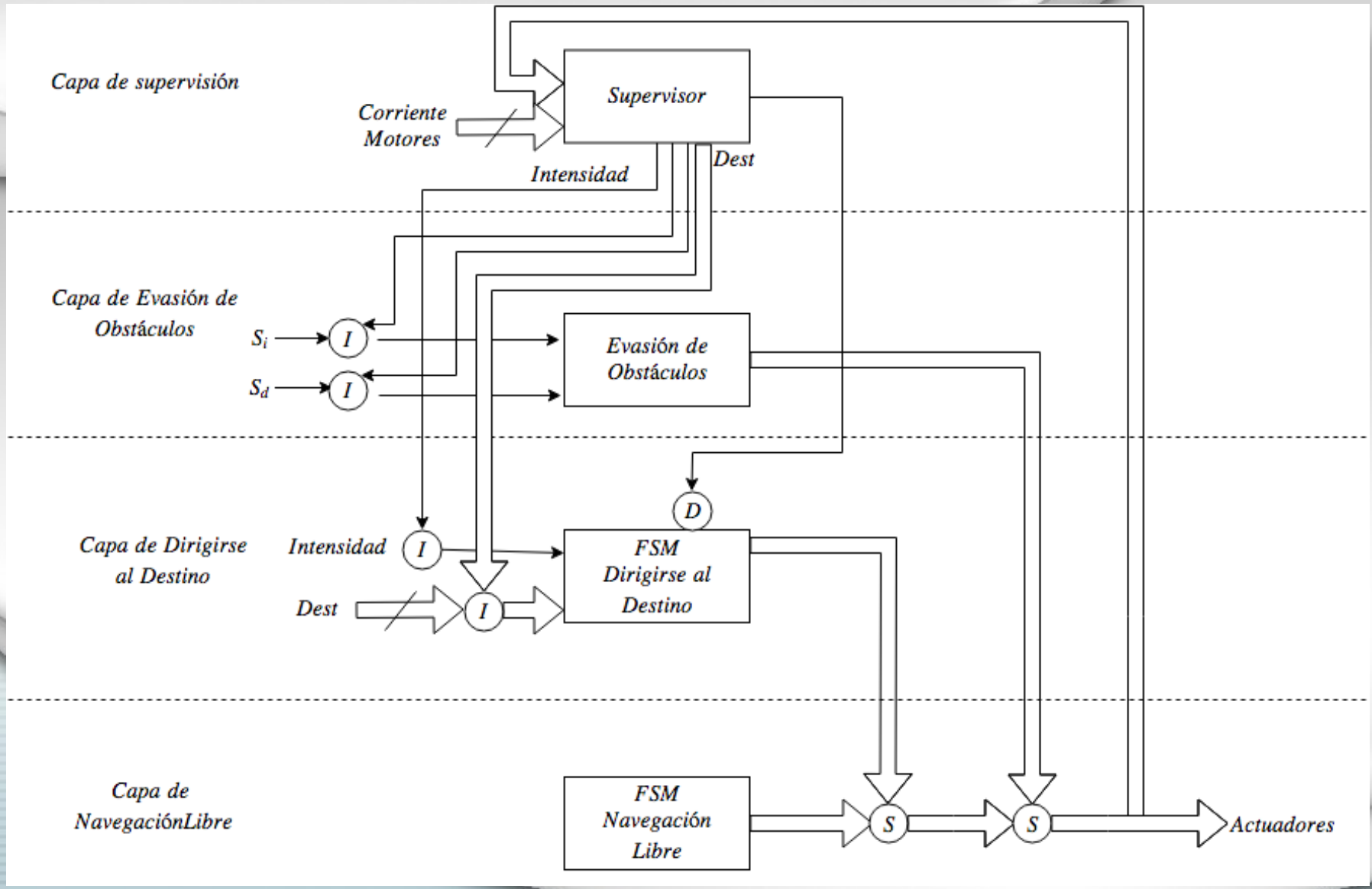


Máquinas de Estado Finitas Aumentadas

Con las AFSMs se puede tener una estructura jerárquica de comportamientos, en donde unos de ellos bloquean y modifican las entradas y salidas de otros comportamientos.



Estuctura jerarquica de AFSMs que buscan una fuente lumínosa y evade obstáculos



Algoritmo Bug I

****

- **Entrada: Un robot con un sensor para detectar obstáculos**
- **Salida: Un camino al destino q_{goal} o la conclusión de que tal camino no existe.**