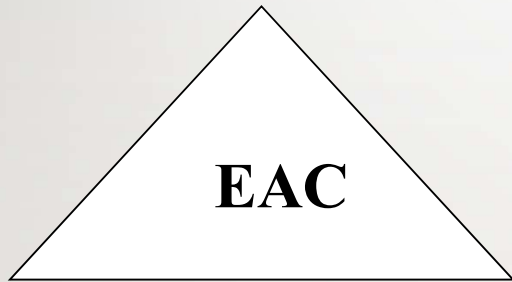


Fragmento sobre Recorrido de Laberintos, tomados de las presentaciones
Evolución y Sistemas Evolutivos Sev www.fgalindosoria.com/eac/2017/doc/sevolutivos17_a.pptx
SISTEMAS CONSCIENTES Scon www.fgalindosoria.com/eac/2017/doc/sconscientes17_a.pptx
(Que les recomiendo que vean)

Consciencia



Evolución **Afectividad**
<http://www.fgalindosoria.com/eac/>

Del
Seminario de Investigación
Sistemas Evolutivos,
Afectivos y Conscientes

www.fgalindosoria.com/seac2017/seac.pdf

Fernando Galindo Soria

www.fgalindosoria.com

Paola Neri Ortiz

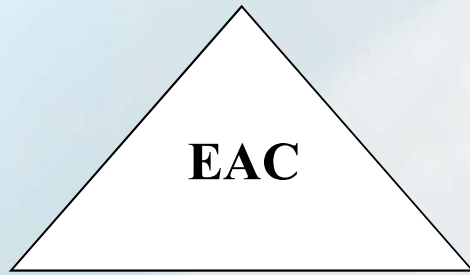
www.eventbrite.com/o/paola-neri-ortiz-12877179662

Itztli García Salas

[http://www.fgalindosoria.com/informaticos/investigadores/Itztli_\(Horacio_Alberto\)_Garcia_Salas/](http://www.fgalindosoria.com/informaticos/investigadores/Itztli_(Horacio_Alberto)_Garcia_Salas/)

Auditorio Sor Juana (edificio C)
Tecnológico de Estudios Superiores de Ecatepec (TESE)
Del Lunes 13 al Jueves 16 de Marzo
de 2017 de 12:00 a 17:00 horas de la Ciudad de México
<http://www.fgalindosoria.com/seac2017/>

Consciencia



Evolución

Afectividad

www.fgalindosoria.com/eac/

Seminario de Investigación Sistemas Evolutivos, Afectivos y Conscientes

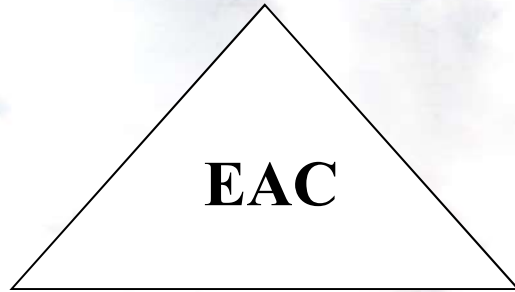
www.fgalindosoria.com/seac/2017/

Evolución y Sistemas Evolutivos Sev

La evolución, el crecimiento, el aprendizaje, el pensamiento, la vida, la transformación de nuestra imagen de la realidad, los procesos de descomposición, el desarrollo y transformación de las empresas, sociedades, organizaciones, países, galaxias, universos, etc., son manifestaciones de un mismo proceso general de transformación o cambio al que por facilidad llamamos evolución.

www.fgalindosoria.com/eac/evolucion/

consciencia



Evolución

Afectividad

<http://www.fgalindosoria.com/eac/>

SISTEMAS CONSCIENTES

Scon

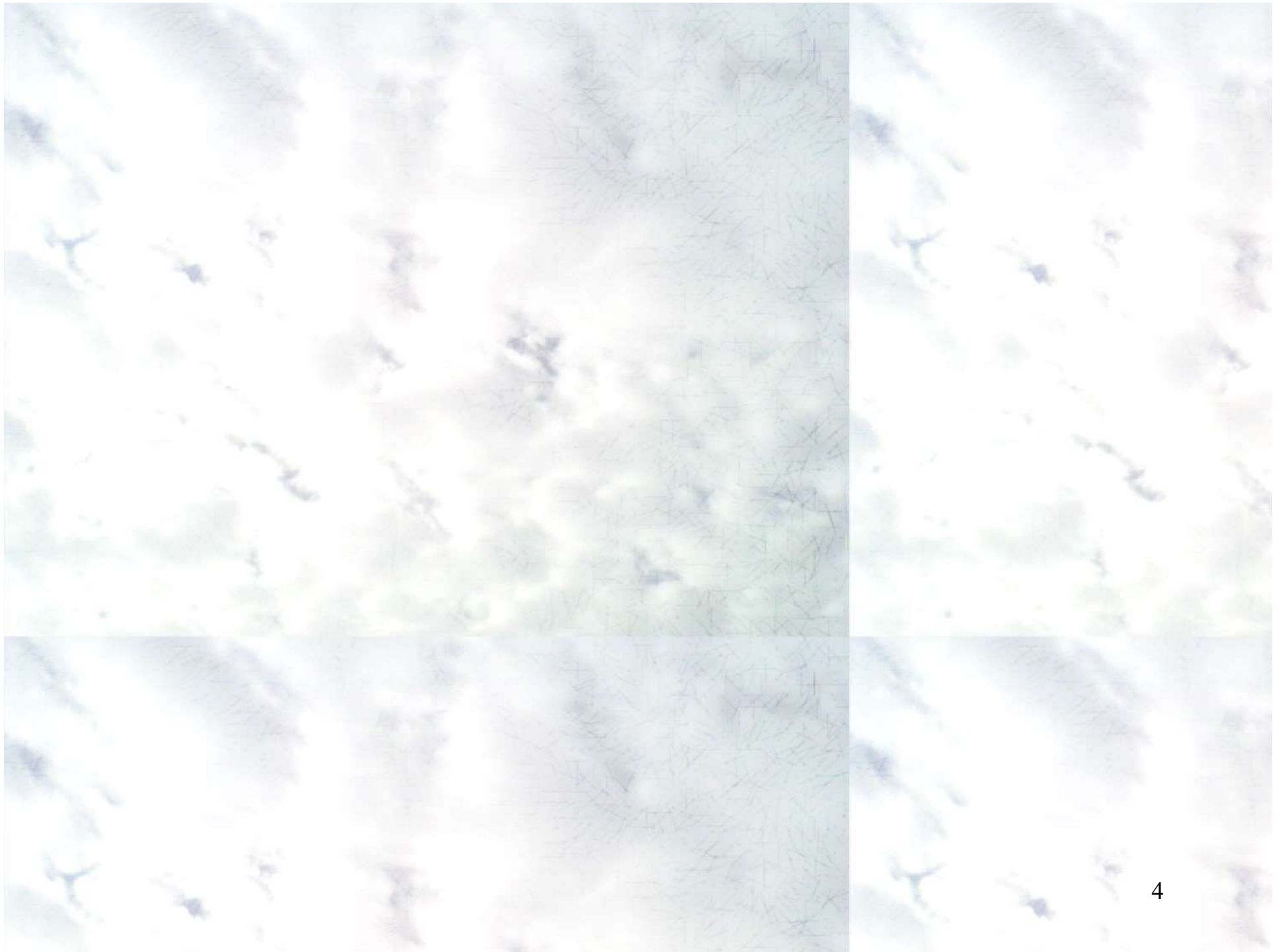
www.fgalindosoria.com/eac/consciencia/

consciencia de si mismo,
de los demás,
del entorno,
de lo trascendente

Cuando Regina paso por el Zócalo de la Ciudad de México, acompañada por los Cuatro Guardianes de la Tradición, uno de ellos le dijo que fueran a Palacio Nacional, para que tomara posesión de su reino, y ella le contesto que todavía no, que ella no quería ser reina de un pueblo sin consciencia.

Versión libre de un fragmento de REGINA de Don Antonio Velasco Piña
Que trata sobre El Despertar de la Consciencia Cósmica y la Percepción de lo Sagrado

Tenayuca, Ciudad de México, Octubre del 2009, Enero del 2017



Ejemplo

Seguidor de Laberintos

1964 1965 Concursos de Laberintos en el pizarrón

1982 Laberintos con número de forma

UPIICSA del IPN, 1984-

Lenguajes de trayectoria

Sistema que sigue una trayectoria y construye una imagen de está

Seguidor de Laberintos

Trazar rutas

Detecta objetos

Tomar decisiones de acuerdo a los objetos



31st All
Japan
Micromouse
Robot
Event:
Ng Beng
Kiat
Robots
Dreams

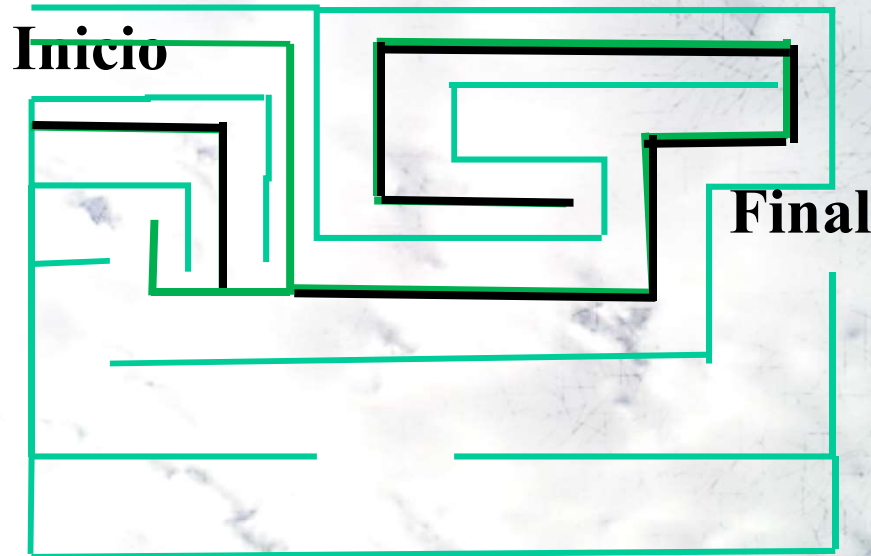
Subido
el 26
nov.
2010

Seguidor de Laberintos

“Ng Beng Kiat, from Singapore, is one of the top micromouse robot builders in the world and has often won the All Japan Micromouse Robot competition. This year, although his mouse turned in excellent times, he had to settle for second place. For more information visit Robots Dreams at <http://www.robots-dreams.com>.”

31st All Japan Micromouse Robot <https://www.youtube.com/watch?v=76bl1lun09Q>

Exploradores y Seguidores de Laberintos



Son programas, robots, representaciones matemáticas que navegan en un laberinto, construyen una imagen de él y lo usan para recorrerlo y resolver problemas

En este ejemplo de exploradores y seguidores **podemos visualizar algunos ejemplos de mecanismos de percepción y representación de la realidad**, todos estos mecanismos y representaciones (u otros) son usados y dependiendo de problema o de las capacidades de los desarrolladores convienen más unos u otros

Por ejemplo, **existe múltiples formas posibles de percepción** como poner una 'cámara' que detecte el movimiento, poner un 'GPS', poner un 'detector de movimiento', etc.

Múltiples formas de representar la imagen de la realidad, como coordenadas (por ejemplo almacena las coordenadas y el ángulo al que se mueve al empezar a caminar y cada que gira) o lo almacena como vector, como programa o código tipo logo o C++ o java o..., lo almacena como número de forma, como una oración, como autómata finito, como matriz, como arreglo o tabla de coordenadas, etc.

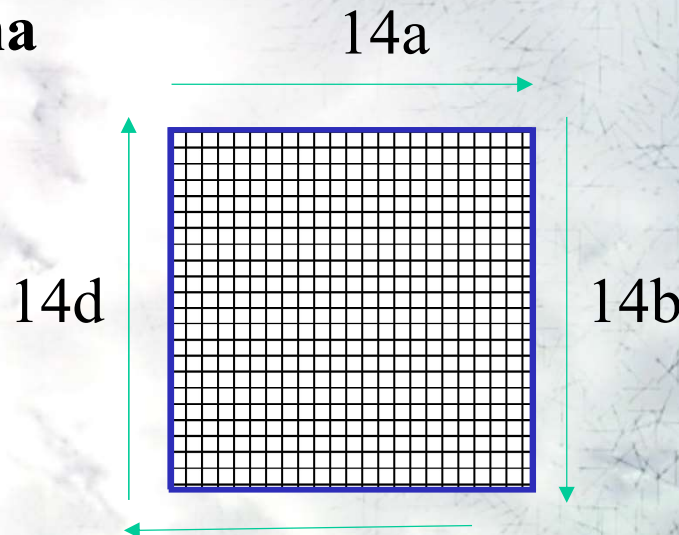
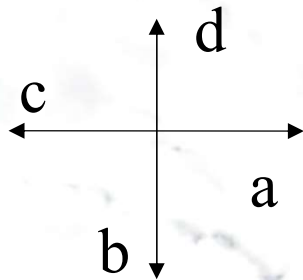
Y dependiendo del tipo de mecanismo de percepción y representación que se maneje es el tipo de análisis y operaciones que se pueden realizar sobre la imagen

Una forma simple de representar un laberinto, como una matriz llena de ceros y poniendo el numero 1 para indicar las paredes.

[illegible]

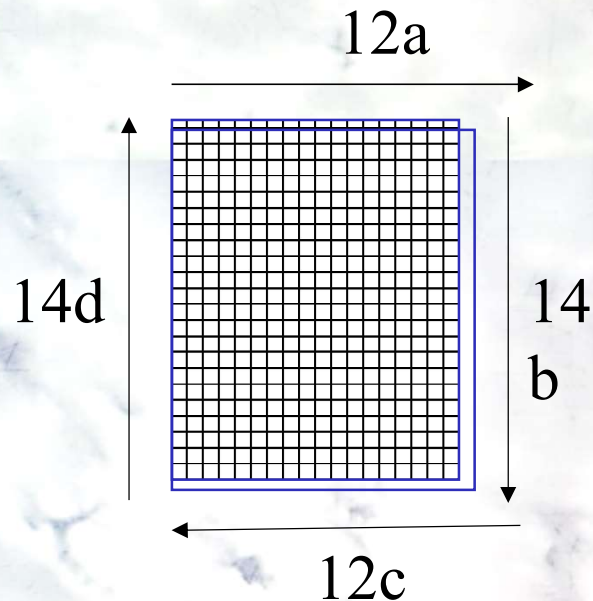
Sistemas que siguen una trayectoria y construyen una imagen de está

Número de Forma
o de Freeman

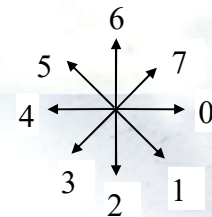


Número de Forma
14a 14b 14c 14d

Otra convención
Número de Forma
De 8 direcciones

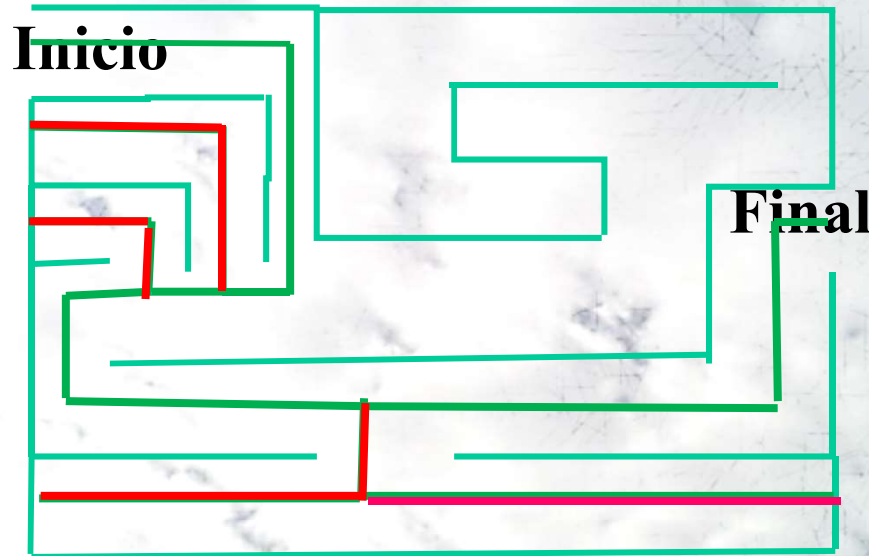


14c
Número de Forma
12a 14b 12c 14d



00066644322

Seguidor de Laberintos



Método

Camina pegado a la derecha

Construyendo el número de forma

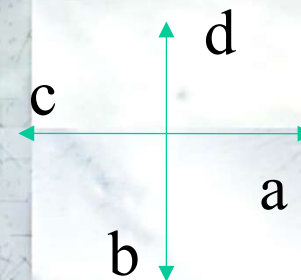
Cuando encuentra tope regresa hasta que encuentra un camino a la derecha que no ha recorrido

Borrando el número de forma hasta el nuevo camino

Hasta que llega al final

I Número de Forma

12a	12b	3c	8d	8c	8a	8b
3c	3d	4c	4a	3b		
3c	4b	12a	<u>4b</u>	14c	14a	16a 16c <u>4d</u>
14a	8d	2a				



F

Si observamos el numero de forma

I

12a 12b 3c 8d 8c 8a 8b
3c 3d 4c 4a 3b
3c 4b 12a 4b 14c 14a 16a 16c 4d
14a 8d 2a

F

Main()

{

Line (12, 0)	dibuja una línea de tamaño 12 a 0 grados	12a
Line (12, 270)	dibuja una línea de tamaño 12 a 270 grados	12b
Line (3, 180)	dibuja una línea de tamaño 3 a 180 grados	3c
Line (8, 90)	dibuja una línea de tamaño 8 a 90 grados	8d

.....

Aquí es importante recordar que las letras, **a, b, c, d**, están indicando dirección o sea que la letra **a** corresponde a moverse a **0 grados**,

la **b** hacia abajo o sea a **270 grados**,

c corresponde a **180 grados** y **d** a **90 grados**

O sea que por ejemplo **12 a** indica **muévete 12 pasos a 0 grados** o **12 0**

Se puede ver que se parece al código de un programa, como por ejemplo en:

Pseudocódigo, C++, Java

Si se observa **este programa es totalmente repetitivo**, por lo que una opción es hacer **un generador de programas elemental**, para lo cual primero creamos un arreglo con dos columnas donde tiene almacenadas todas las magnitudes y direcciones que se detectaron al recorrer el laberinto

Para lo cual primero tomamos el número de forma generado por el seguidor de laberintos

I

12a 12b 3c 8d 8c 8a 8b
 3c 3d 4c 4a 3b
 3c 4b 12a 4b 14c 14a 16a 16c 4d

F

Y lo vemos
 como un arreglo
 de 2 columnas
 Magnitud y
 dirección

12 a
 12 b
 3 c
 8 d
 8 c
 8 a
 8 b
 3 c
 3 d
 ...

Para hacer el programa creamos una tabla a la que llamamos

Numforma

Con 2 columnas

Magnitud y dirección

12	0
12	270
3	180
8	90
8	180
8	0
8	270
3	180
3	90
...	...
...	...

Entonces ya se puede hacer un generador de programas elemental que sigue la siguiente arquitectura

Numforma
Magnitud y dirección

12	0
12	270
3	180
8	90
8	180
8	0
8	270
3	180
3	90
...	...

Generador
de
Programas

Archivo con
Código generado

```
Main()
{
Line (12, 0);
Line (12, 270);
Line ( 3, 180);
Line ( 8, 90);
.....
.....
```

Donde explícitamente
se vuelven
independientes
relativamente, la
estructura y los datos
del sistema (representa
por la tabla
Numforma) de los
procesos

Un programa Generador de Programas seria:

Main()

{

Imprime ("Main()"); Imprime ("{");

Mientras existan elementos en Numforma

Lee (magnitud(i), dirección(i));

Imprime ("Line ("); Imprime (magnitud(i));

Imprime (dirección(i)); Imprime (");"); i++;

Imprime ("}");

}

Imprime Main y {

Mientras Numforma tenga elementos,

Lee los valores de magnitud y dirección
imprime un texto que dice Line(, pone
los valores de magnitud y dirección,
cierra el paréntesis del line

Cierra el programa imprimiendo }

El problema que se presenta en este caso es que **el programa generado se tiene que compilar y ejecutar, para que dibuje el recorrido.**

Existe un método mucho mas directo, que consiste en que **en lugar de que el programa genere otro programa, dibuje el recorrido directamente,** el código seria:

Main()

{

Mientras existan elementos en Numforma

Lee magnitud(i) y dirección(i)

Line (magnitud(i), dirección(i))

i++

}

Mientras la tabla Numforma tenga elementos,

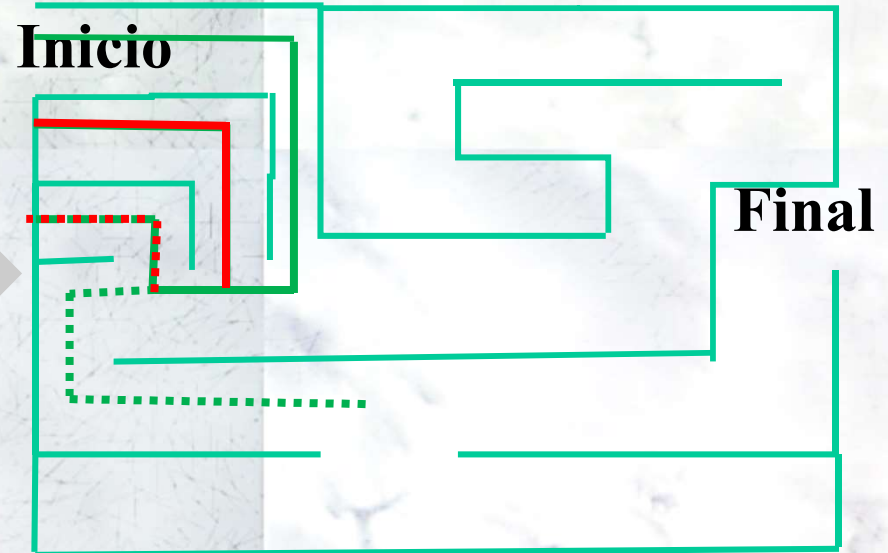
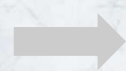
el generador toma los valores de magnitud y dirección

Grafica una línea con magnitud y dirección

Que como se puede ver es mucho mas simple, la arquitectura del sistema es

Numforma
Magnitud y dirección

12	0
12	270
3	180
8	90
8	180
8	0
8	270
3	180
3	90
...	...



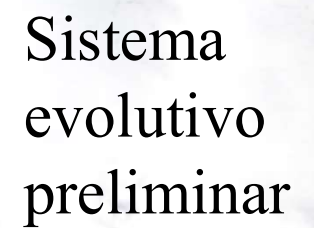
Entonces este seguidor de laberintos tiene la siguiente arquitectura **SensoMotriz**

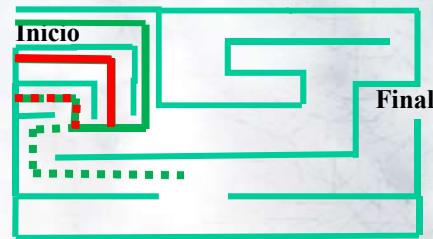
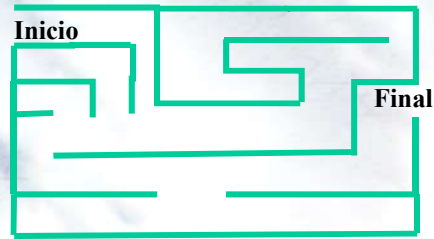
The diagram illustrates the architecture of a maze follower system, labeled **SensoMotriz**. It shows a maze with a starting point (**Inicio**) and an ending point (**Final**). The maze is processed by an **Explorador del laberinto** (Maze Explorer), which outputs a sequence of **Numforma** (Shape Number) and **Magnitud y dirección** (Magnitude and direction) to an **Ejecutor** (Executor). The executor then follows the path through the maze. The entire system is part of a **Sistema evolutivo preliminar** (Preliminary evolutionary system).

The output of the **Explorador del laberinto** is a sequence of **Numforma** and **Magnitud y dirección** values:

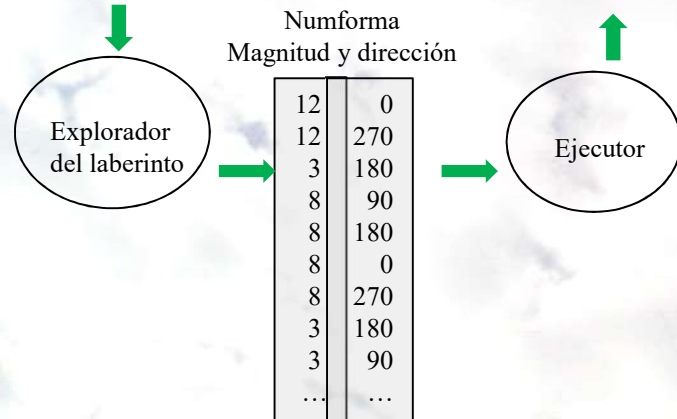
Numforma	Magnitud y dirección
12	0
12	270
3	180
8	90
8	180
8	0
8	270
3	180
3	90
...	...

The **Ejecutor** then follows the path through the maze, starting from **Inicio** and ending at **Final**.





Este es un ejemplo básico de lo que conocemos como un sistema evolutivo.



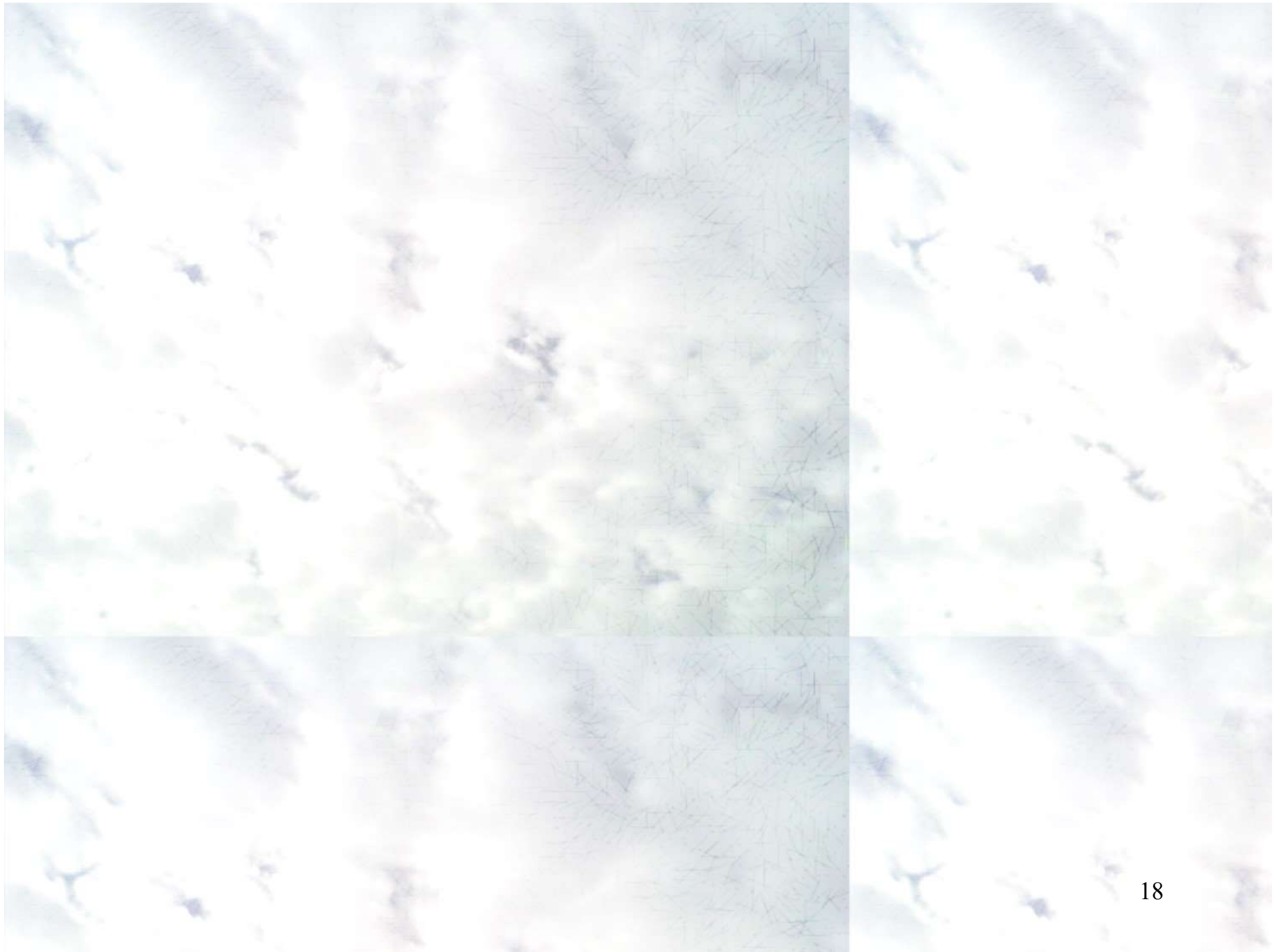
Si se observa no necesitamos programar el recorrido del laberinto, ya que es el propio sistema el que recorre el laberinto y construye una imagen de la realidad que recorrió, para luego poder recorrerlo de nuevo

Para hacer el programa creamos una tabla a la que llamamos **Numforma**
Con 2 columnas
Magnitud y dirección

12	0
12	270
3	180
8	90
8	180
8	0
8	270
3	180
3	90
...	...
...	...

este arreglo representa la estructura de la imagen de la realidad

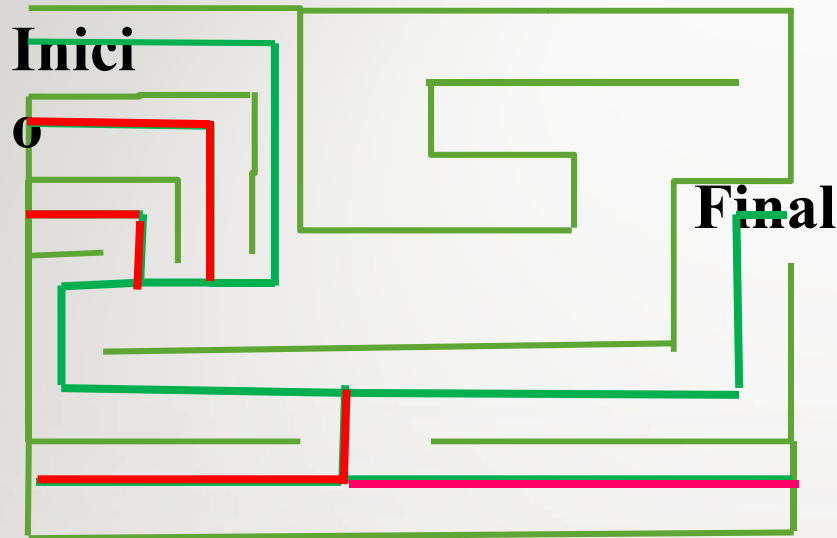
Si se observa esta estructura está **independiente relativamente** de los procesos que se deben ejecutar en el sistema, lo cual le da una gran fuerza al sistema ya que si cambia el laberinto solo se vuelve a generar el arreglo, pero no se requiere hacer otro programa.



Análisis de la Imagen del Entorno

mantienen la imagen
actualizada

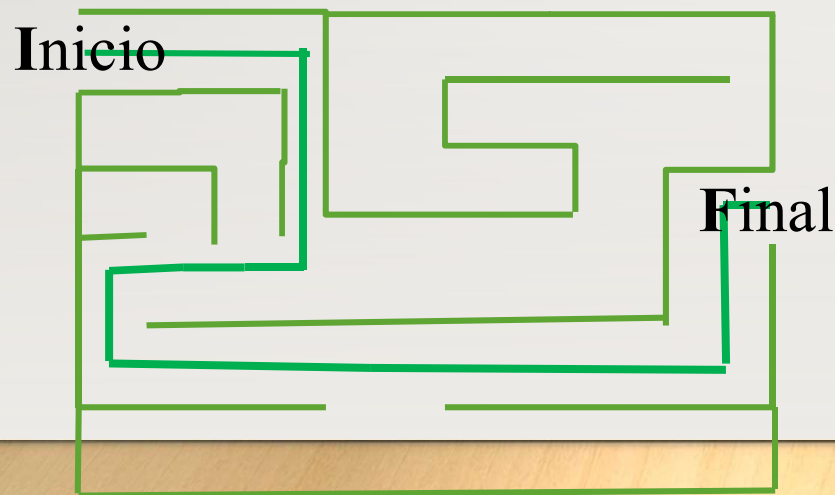
y la utilizan para interactuar con
el entorno



oración en un lenguaje de Trayectoria

I 12a 12b 3c 8d 8c 8a 8b 3c 3d 4c 4a 3b 3c
 4b 12a 4b 14c 14a 16a 16c 4d 14a 8d 2a F

Se analiza la imagen del entorno, por ejemplo en este caso mediante algebra lingüística, para simplificar la expresión



simplificando la oración anterior,
queda

I12a 12b 9c 4b 26a 8d 2a **F**

Que representa el camino mas simple de la trayectoria en el laberinto

Ahora, si observamos el número de forma

I

12a 12b 3c 8d 8c 8a 8b
3c 3d 4c 4a 3b
3c 4b 12a 4b 14c 14a 16a 16c 4d
14a 8d 2a

F

Se puede ver que se parece al código de un programa, como por ejemplo en:
Pseudocódigo, C++, Java

O a una lista de vectores con magnitud y dirección

Y También es equivalente a la siguiente

oración en un lenguaje de Trayectoria

I 12a 12b 3c 8d 8c 8a 8b 3c 3d 4c 4a 3b 3c 4b 12a 4b 14c 14a 16a 16c 4d 14a 8d 2a

F
Donde **I** y **F** se conocen como rutinas semánticas

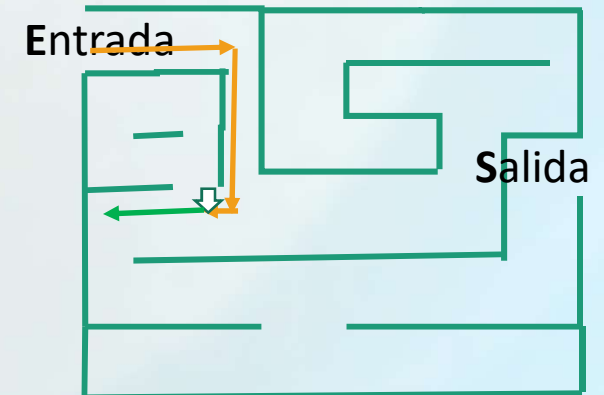
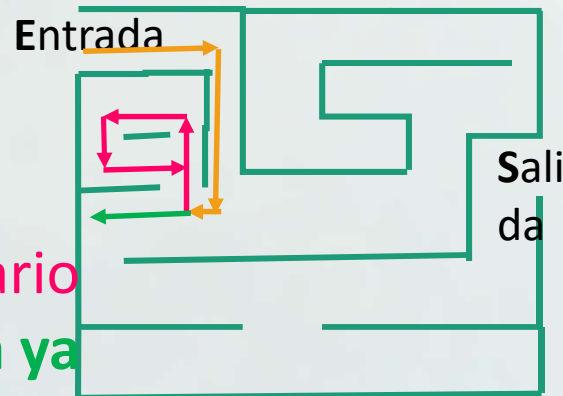
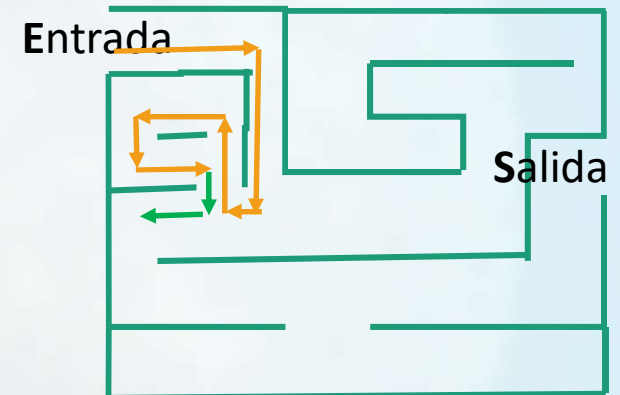
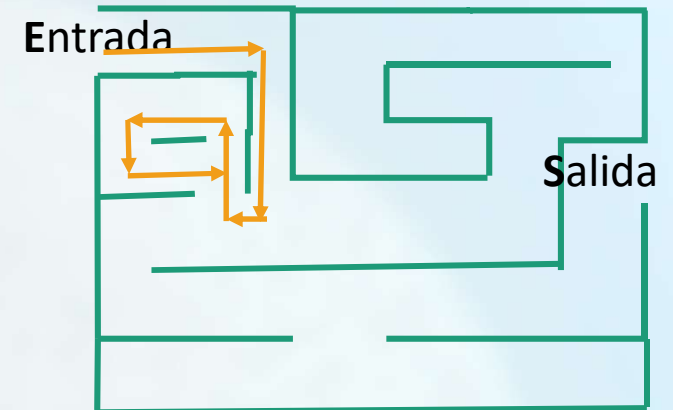
Sistemas Evolutivo de lenguajes de Trayectoria

www.fgalindosoria.com/eac/evolucion/sistemas_evolutivos_lenguajes_trayectoria/sist_evol_leng_trayectoria.pdf

Esto se presenta cuando al recorrer el laberinto se llega a algún punto ya visitado anteriormente como se ve en la imagen lateral

La primer es integrar el recorrido a la oración (esto funciona cuando se esta recorriendo un espacio que contienen además objetos o cosas importantes para el sistema)

La otra consiste en ignorar completamente la trayectoria problema, para lo cual se marca el recorrido como no necesario o se pone una marca para ya no pasar por ese recorrido



Una de las ventajas de ver los números de forma como oración es que se puede hacer algebra para simplificar la expresión

Por ejemplo se pueden agrupar las estructuras que representan caminos que ya se han recorrido o ciclos

La siguiente oración

I 12a 12b 3c 8d 8c 8a 8b 3c 3d 4c 4a 3b 3c 4b 12a 4b 14c 14a 16a 16c 4d 14a 8d 2a F

Detectando los ciclos quedaría

I 12a 12b 3c (8d (8c 8a) 8b) 3c (3d (4c 4a) 3b) 3c 4b 12a [4b (14c 14a) (16a 16c) 4d] 14a 8d 2a F

Los elementos entre paréntesis se conocen como ciclos

En algunos casos y dependiendo del problema conviene recordar donde están los ciclos, por lo que se marcan dentro de la oración y del numero de forma como nuevas rutinas semánticas, en otros (como en este caso) simplemente se anulan, quedando la oración como

I 12a 12b 3c (8d 8b) 3c (3d 3b) 3c 4b 12a [4b 4d] 14a 8d 2a F

Simplificando mas queda la oración:

I 12a 12b 3c 3c 3c 4b 12a 14a 8d 2a F

Que representa el camino mas simple de la trayectoria

*Algunas propiedades matemáticas de los sistemas lingüísticos*²³

www.fgalindosoria.com/linguisticamatematica/propiedades_matematicas_sistemas_linguisticos/prop_mate_sistemas_linguisticos.pdf

[illegible]

Final

La oración

I 12a 12b 3c 3c 3c 4b 12a 14a 8d 2a **F**

Representa la trayectoria directa en el laberinto

Pero si observamos la oración

I12a 12b 3c 3c 3c 4b 12a 14a 8d 2a **F**

Las cadenas adyacentes

3c 3c 3c

Tienen la misma dirección,
por lo que se pueden
simplificar como

9c Que se denota como:

$3c \ 3c \ 3c \rightarrow 9c$

También

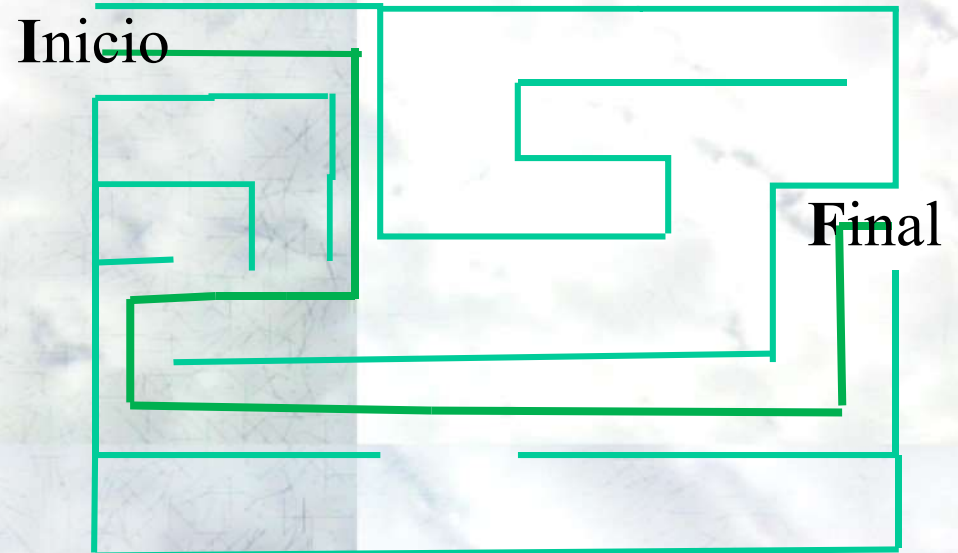
12a 14a → 26a

Donde $a \rightarrow b$ significa
que a se sustituye o
reescribe como b

Por lo que simplificando la oración anterior, queda

I12a 12b 9c 4b 26a 8d 2a **F**

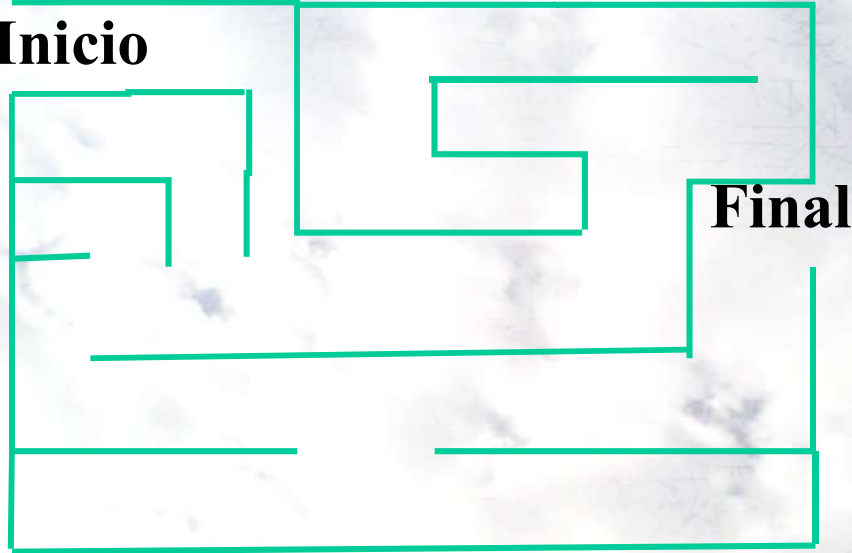
Que representa el camino
mas simple de la
trayectoria en el laberinto



I 12a 12b 9c 4b 26a 8d 2a F

Entonces el seguidor de laberintos tiene la siguiente arquitectura **SensoPsicoMotriz**

Inicio



Inicio

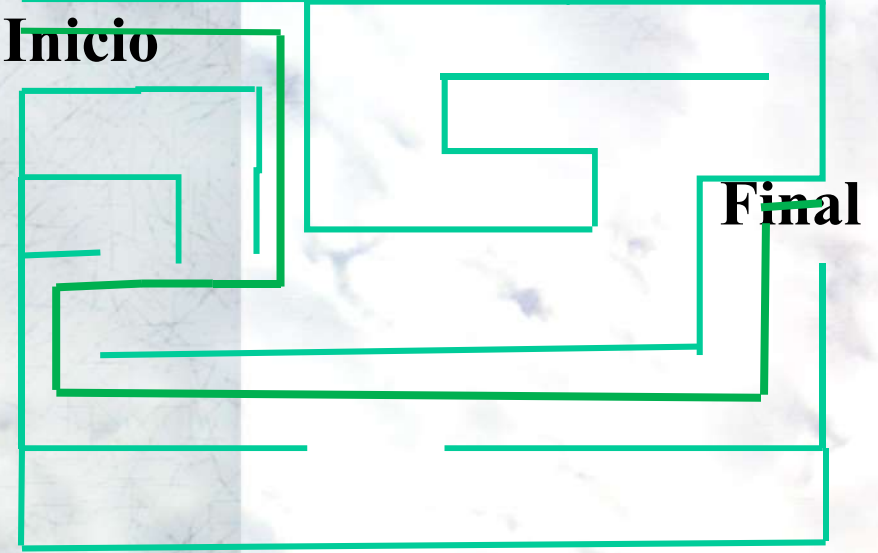


imagen del entorno analizada

I 12a 12b 9c 4b 26a 8d 2a F

Numforma
Magnitud y dirección

12	0
12	270
9	180
4	270
26	0
8	90
2	0

Ejecutor

Explorador
del laberinto

Analizador
de la imagen
del entorno

imagen del entorno

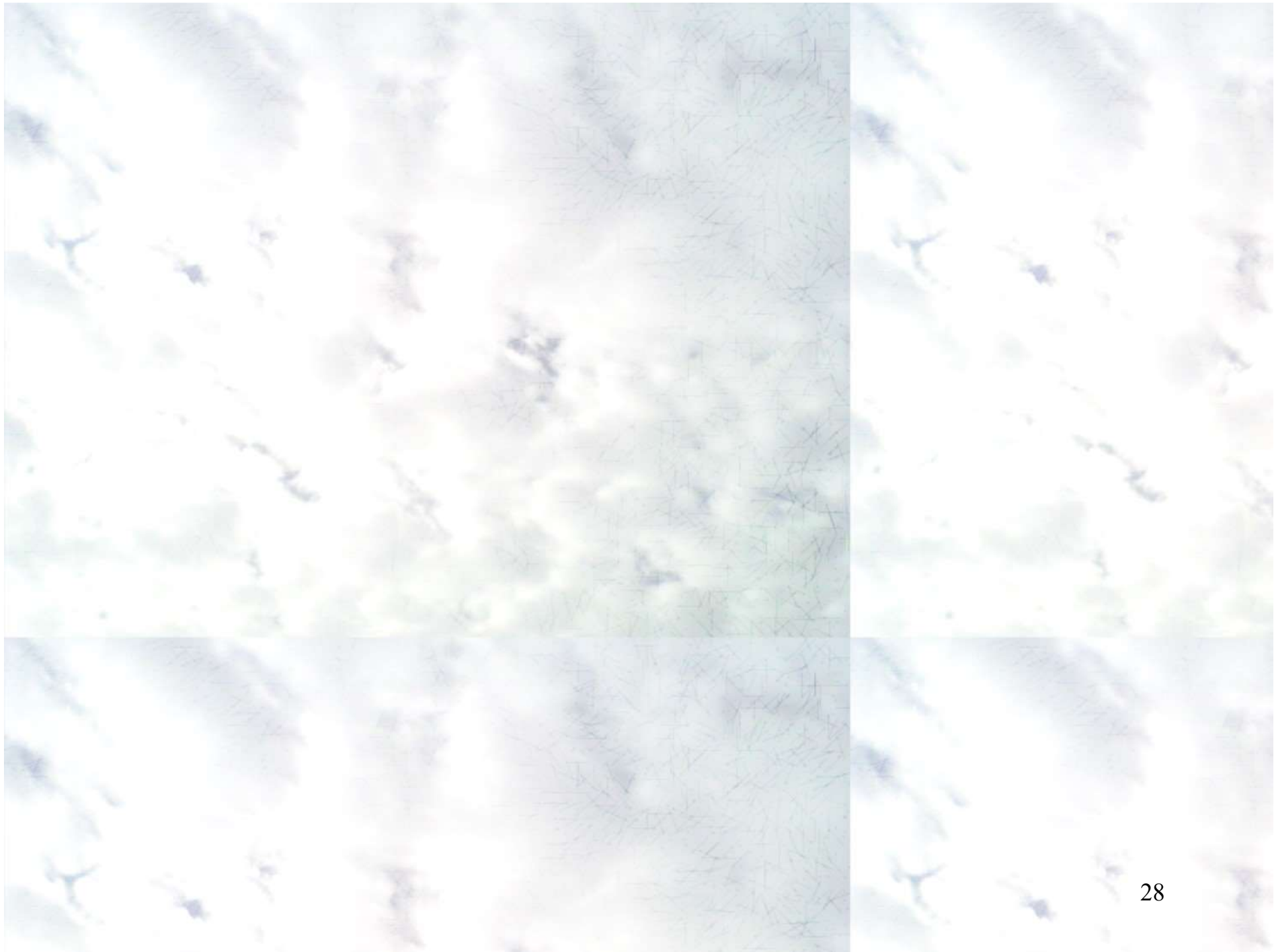
I 12a 12b 3c 8d 8c 8a 8b 3c 3d 4c 4a 3b 3c 4b 12a 4b 14c 14a 16a 16c 4d 14a 8d 2a F

En General

La trayectoria no es una línea es una brama en múltiples dimensiones

Cada punto de la brama forma un espacio caótico, o sea que cuando seguimos una trayectoria no seguimos una línea sino que cada punto forma un espacio de caos y podemos tomar múltiples posibilidades en ese espacio.

La trayectoria no solo puede ser espacial, también puede ser temporal o espacio temporal o sea puede formar un espacio de recuerdos



Recorrido de espacios con objetos

Los espacios en general no están vacíos, tienen objetos, obstáculos y marcas o cosas que permiten distinguir entre los diferentes lugares del espacio,

por lo que la idea es que el sistema también perciba estos objetos, los integre a su imagen del entorno, los manipule y los use para navegar en el espacio y tomar decisiones

Recorrido de espacios con objetos

Los espacios en general no están vacíos, tienen objetos, obstáculos y marcas o cosas que permiten distinguir entre los diferentes lugares del espacio, por lo que en general la idea es que el sistema también perciba estos objetos, los integre a su imagen del entorno, los manipule y los use para navegar en el espacio y tomar decisiones

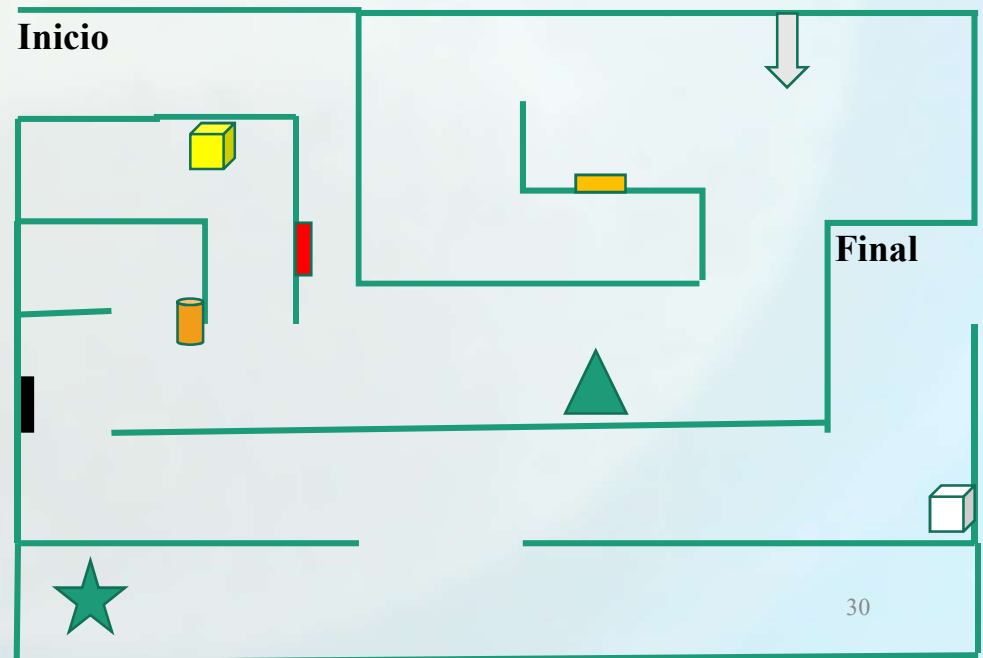
Por ejemplo los siguientes objetos y marcas:

Inicio          Final

I Mr Ca Mg Mn Ta Fg Mn Ea Cb F

Aparecen en el laberinto

Y se integran dentro de la imagen del entorno como rutinas semánticas, donde por ejemplo F indica Final, Mr marca roja, Cb cubo blanco,...



Cuando el seguidor del laberinto o en general un reconocedor del entorno recorre este espacio encuentra los objetos y va marcando dentro de la oración que representa al entorno con las **etiquetas semánticas** donde encontró las marcas

Semántica

La semántica estudia el
significado de las cosa

Y en principio la podemos ver como una relación que se establece entre
varias cosas

$a \ R \ b$

Existen múltiples formas para manejar la semántica

Por ejemplo desde que se crea el sistema se establece una tabla de reescritura con 2 columnas, en la primera se coloca los objetos (imágenes, sonidos, descripciones, etc.) y en la segunda se coloca el significado que le queremos asignar (etiqueta, rutina semántica, otro objeto, etc.) entonces cuando el sistema (seguidor, navegador, programa, robot,...) encuentra algo que corresponda a la primera columna asigna el significado presente en la segunda columna

Objeto	Significado
	Triangulo azul
	rectángulo
	estrella
	Triangulo blanco
.....	
	Flecha roja

Otra opción consiste en que cuando el sistema esta navegando en el entorno y encuentra algo que no reconoce, pregunta su significado y lo asigna

Métodos supervisados, semi supervisados, no supervisados

Por ejemplo, en el laberinto se establece un acuerdo entre los objetos y un conjunto de etiquetas, el cómo se establece el acuerdo depende del tipo de método:

en un extremo alguien definió las etiquetas,

en el otro, cada que el sistema encuentra un objeto le pone una etiqueta aleatoriamente sin que nadie participe en el proceso

y en un punto intermedio el sistema pone etiquetas y las modifica conforme la interacción con el ambiente va asignando o nuevas relaciones características a los
objetos

Semántica se construye mediante interacción con el entorno

Por ejemplo, la palabra perro está relacionada con la imagen de un perro o más generalmente con lo que entendemos como perro, ya sea porque alguien nos lo explico

o porque hemos visto muchos ejemplos de perro

Por ejemplo, al ir en la calle en varias ocasiones hemos escuchado la palabra perro y hemos visto un animal de 4 patas que ladra, menea la cola,, y que tiene ciertas características que vamos asociando con la palabra perro.

Una técnica usada actualmente consiste en **sistemas que navegan en la red para construir su imagen semántica**, como si fueran personas navegando por una ciudad.

Por ejemplo se lanza un buscador en la red que asocia imágenes de perros con sonidos que dicen perro y con la palabra perro

En 1995 en la ESCOM del IPN en México, varios alumnos investigadores desarrollaron sistemas semánticos que tomaban el significado asociando información de varias fuentes como sonidos, imágenes y texto

Pragmática

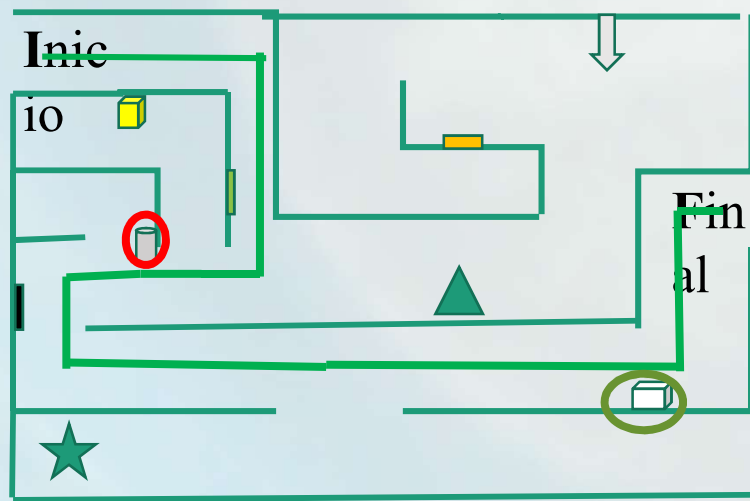
Cada organismo, persona, sistema, bicho tiene su propia imagen de la realidad, por lo que, cuando se percibe algo y se integra a la imagen de la realidad, cada ente lo puede integrar de diferente manera

Por ejemplo si alguien escucha la palabra perico, en principio un mecánico entendería que se refieren a una herramienta, pero otras personas pueden entender que se refieren a un animal

Mediante la imagen del entorno representada por la oración semántica es relativamente fácil ir a diferentes partes del laberinto, o sea que el sistema empieza a ser capaz de visualizar su entorno mediante una representación interna

por ejemplo se le puede decir que vaya al final y el sistema recorre el laberinto siguiendo su representación interna hasta que llega a F

O se le puede decir que vaya al cubo blanco y el sistema se movería en el laberinto siguiendo la oración hasta llegar a la marca de Cb

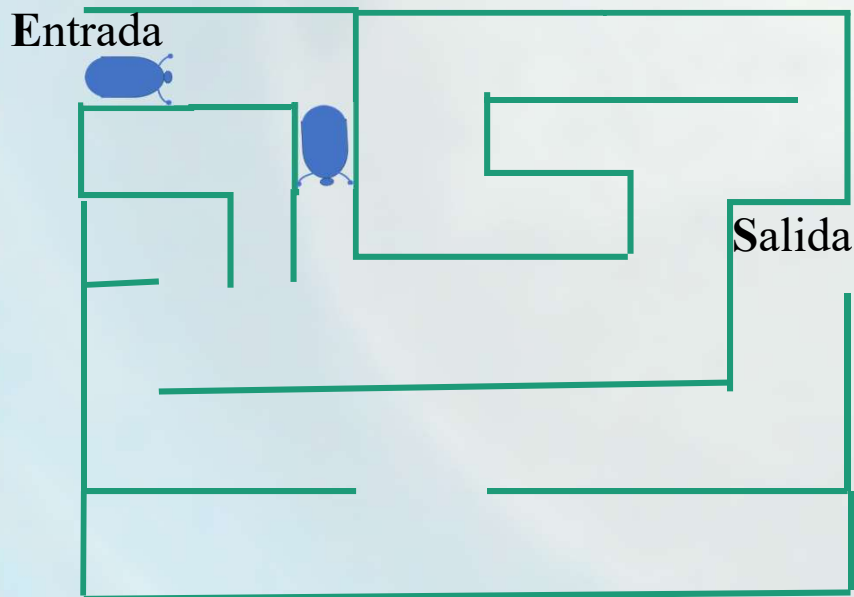
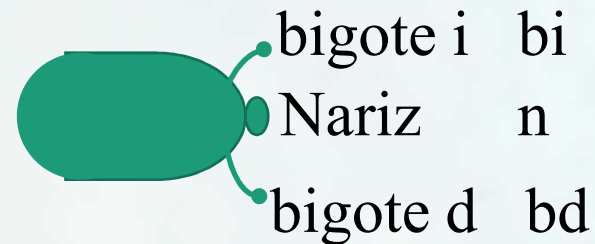


Aun mas, *si el sistema esta recorriendo el laberinto y se encuentra por ejemplo en la marca gris **Mg** y se le dice que vaya a al cubo blanco **Cb** seguiría el camino de **Mg** a **Cb**, o sea que ya “se empieza a dar cuenta de donde esta, a donde va y como llegar”*

Topos 1990-

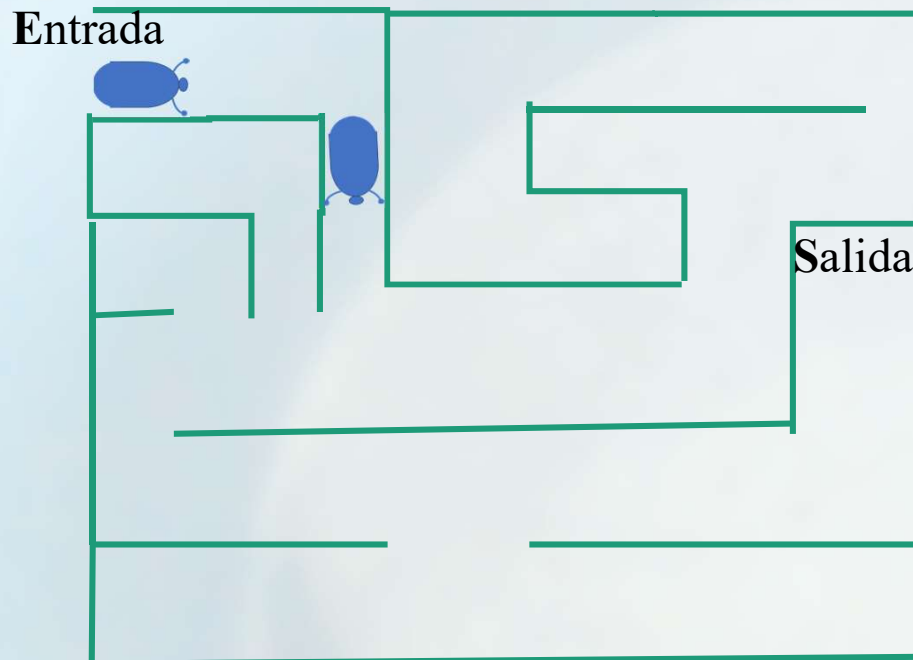
UPIICSA del IPN, Celanese Mexicana

Un topo es un robot que tiene al menos 3 mecanismos de percepción 2 bigotes, una nariz, y un reloj



El Topo se suelta en un laberinto comienza a caminar y con sus perceptores: bigotes, nariz y reloj va construyendo la imagen del laberinto.

Con los bigotes va detectando las paredes laterales, con la nariz detecta cuando choca con una pared y con el reloj marca el tiempo transcurrido



La imagen del entorno consiste en 4 vectores que se llenan con ceros y unos, el primer vector almacena el tiempo T, cada que pasa una unidad de tiempo se almacena un 1 en el vector T,

al mismo tiempo en los otros vectores se almacena lo que detecta el bigote d, el bigote i y la nariz en ese tiempo

Cuando el bigote d detecta una pared almacena 1 y cuando no detecta almacena 0, lo mismo ocurre con el otro bigote y la nariz

T	1	1	1		1	1	1	1	1	1	1	1	1	1	
bd	1	1	1		0	0	0	0	0	0	0	1	1	1	
bi	0	0	0		0	0	0	0	0	0	0	1	1	1	
n	0	0	0		0	0	0	1	1	1	1	0	0	0	

El primer topo de nuestro grupo de investigación fue construido por estudiantes investigadores de la Licenciatura en Ciencias de la Informática de la UPIICSA del IPN en 1990.

El método para recorrer el laberinto y construir la imagen de la realidad consistía básicamente en lo siguiente:

- Mientras alguno de los bigotes detectara pared y la nariz no continuaba caminando

- Cuando la nariz detectaba pared

 - Si los dos bigotes detectaban pared comenzaba a retroceder hasta que alguno no detectara pared

 - Si solo uno de los dos bigotes no detectaba pared giraba hacia ese lado

 - Si los dos bigotes no detectaban pared, giraba a la derecha o a la izquierda (dependiendo del método de recorrido fijado por los diseñadores)

- Y así hasta que llegaban al final

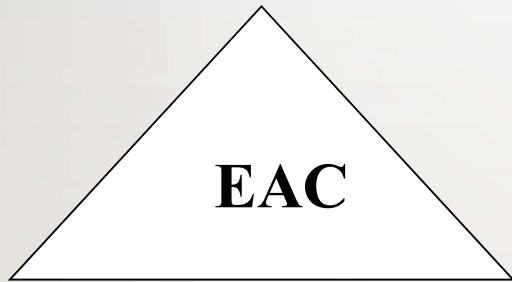
A la información resultante en los vectores se les aplicaba Álgebra Lingüística para simplificar y encontrar el mejor recorrido.

Recorrido en una ciudad

Algebra para seleccionar el recorrido mas corto

Fragmento sobre Recorrido de Laberintos, tomados de las presentaciones
Evolución y Sistemas Evolutivos Sev www.fgalindosoria.com/eac/2017/doc/sevolutivos17_a.pptx
SISTEMAS CONSCIENTES Scon www.fgalindosoria.com/eac/2017/doc/sconscientes17_a.pptx
(Que les recomiendo que vean)

Consciencia



Evolución **Afectividad**
<http://www.fgalindosoria.com/eac/>

Del
Seminario de Investigación
Sistemas Evolutivos,
Afectivos y Conscientes

www.fgalindosoria.com/seac2017/seac.pdf

Fernando Galindo Soria

www.fgalindosoria.com

Paola Neri Ortiz

www.eventbrite.com/o/paola-neri-ortiz-12877179662

Itztli García Salas

[http://www.fgalindosoria.com/informaticos/investigadores/Itztli_\(Horacio_Alberto\)_Garcia_Salas/](http://www.fgalindosoria.com/informaticos/investigadores/Itztli_(Horacio_Alberto)_Garcia_Salas/)

Auditorio Sor Juana (edificio C)
Tecnológico de Estudios Superiores de Ecatepec (TESE)
Del Lunes 13 al Jueves 16 de Marzo
de 2017 de 12:00 a 17:00 horas de la Ciudad de México
<http://www.fgalindosoria.com/seac2017/>

Número de Forma, Número de Freeman, Código de cadena
Shape Number, Freeman Chain Code, chain code

chain code

Código de cadena

https://en.wikipedia.org/wiki/Chain_code

https://es.wikipedia.org/wiki/Chain_code

SI-VE: SISTEMA DE VISIÓN EXPERTO

Ma. Sandra Camacho V., G. Patricia Gómez R., Jesús M. Olivares C.

“Los autores realizaron este trabajo como parte del curso de Lenguajes de Nivel Alto en el 4º semestre de la Licenciatura en Ciencias de la Informática en la UPIICSA del IPN, concluyéndolo en julio de 1986 y reportado en agosto del mismo año. El Sistema de Visión Experto funciona en una computadora HP-3000 con sistema operativo MPE.”

http://fgalindosoria.com/eac/evolucion/libro_sistemas_evolutivos/I7-SIVE.pdf

Existen múltiples convenciones para los Número de Forma

De los mejores son los desarrollados por el Dr Adolfo Guzman Arenas y por el Dr. Ernesto Bribiesca en el IIMAS de la UNAM

<http://turing.iimas.unam.mx/~ernesto/shape.htm>

Algunos trabajos de Adolfo Guzmán Arenas y Ernesto Bribiesca

Adolfo Guzmán Arenas

Ph. D in Computer Science, December of 1968, MIT.

Computer Recognition of Three Dimensional Objects in a Visual Scene.

(Reconocimiento por Computadora de Objetos de 3 Dimensiones en un Escenario Visual

Ph. D. en Ciencias de la Computación, diciembre de 1968, MIT.)

Uploaded by Adolfo Guzman Arenas

http://www.academia.edu/504993/19_Computer_recognition_of_three-dimensional_objects_in_a_visual_scene

Shape Numbers

The theory of shape numbers was proposed by E. Bribiesca and A. Guzman in 1978. The shape number of a curve is derived for two-dimensional (2D) non-intersecting closed curves that are the boundary of simply connected regions. This description is independent of their size, orientation and position, but it depends on their shape. Each curve carries "within it" its own shape number. The order of the shape number indicates the precision with which that number describes the shape of the curve. For a curve, the order of its shape number is the length of the perimeter of a "discrete shape" (a closed curve formed by vertical and horizontal segments, all of equal length) closely corresponding to the curve. A procedure is given that deduces, without table look-up, string matching or correlations, the shape number of any order for an arbitrary curve. To find out how close in shape two curves are, the degree of similarity between them is introduced. Informally speaking, the degree of similarity between the shapes of two curves tells how deep it is necessary to descend into a list of shapes, before being able to differentiate between the shape of those two curves.

References:

- Bribiesca E. and Guzmán A., Shape Description and Shape Similarity Measurement for Two Dimensional Regions, **Proceedings of The 4th International Conference on Pattern Recognition**, pp. 608-612, Kyoto, Japan (1978).
- Bribiesca E. and Guzman A., How to Describe Pure Form and How to Measure Differences in Shapes Using Shape Numbers, **IEEE Computer Society, Conference on Pattern Recognition and Image Processing**, Chicago, Illinois, U.S.A. (1979).
- Bribiesca E. and Guzmán A., Shape Description and Shape Similarity Measurement for Two dimensional Regions, **Geo-Processing**, Vol. 1, No. 1 pp. 129-144 (1979).
- Bribiesca E. and Guzmán A., How to Describe Pure Form and How to Measure Differences in Shape Using Shape Numbers, **Pattern Recognition**, Vol. 12, No. 2, pp. 101-112 (1980).
- Bribiesca E., Arithmetic Operation Among Shapes Using Shape Numbers, **Pattern Recognition**, Vol. 13, No. 2, pp. 123-137 (1981).
- Bribiesca E., Unsupervised and Supervised-classification on Digital Images Using Shape and Color, **Advances in Information Sciences and Technology**, Vol. I: Pattern Recognition and Digital Technique, Calcutta (1982).
- Bribiesca E., Shape Classification on Digital Images, **Proceedings of the International Society of Photogrammetry and Remote Sensing**, Commission IV, Symposium: Environmental Assessment and Resource Management, Crystal City, Virginia, U.S.A. (1982).

<http://turing.iimas.unam.mx/~ernesto/shape.htm>

Shape description and shape similarity measurement for two-dimensional regions

Ernesto Bribiesca and Adolfo Guzmán Arenas, 1978

http://www.academia.edu/2373106/33_Shape_description_and_shape_similarity_measurement_for_two-dimensional_regions

Ernesto Bribiesca and Adolfo Guzmán Arenas, 1978

Shape Numbers: A Notation to Describe Pure Form and to Measure Resemblance and Difference in Shape

Technical Report PR-78-20 (Orange Series 178), IIMAS. In Spanish.

http://www.academia.edu/550240/34_Shape_Numbers_A_Notation_to_Describe_Pure_Form_and_to_Measure_Resemblance_and_Difference_in_Shape