

---

# Laboratorio de Robots Móviles

## Práctica No. 3

### Búsqueda de Caminos Usando Mapas Simbólicos

Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE100821

---

**Objetivo:** Familiarizar al alumno en el conocimiento de la búsqueda de caminos usando mapas simbólicos.

**Desarrollo:** Para cada uno de los siguientes apartados, realizar los programas que se piden.

**Duración:** Tres semanas

1.- El directorio: ~/data\_pumasimbot contiene los mapas topológicos de los medios ambientes en donde opera el robot en los archivos \*.top

Estos archivos contienen los nodos, su posiciones en el medio ambiente, las distancias entre ellos. Esta información es generada tomando en cuenta los vértices aumentados de los polígonos de los mapas simbólicos, el apéndice A muestra un ejemplo de este tipo de mapas topológicos.

2. - En el simulador para ver un red topológica de un medio ambiente, oprima el botón "Plot Topological". La selección número 7 de los comportamientos muestra la navegación del robot utilizando una red topológicas con búsqueda primera en profundidad o el algoritmo de Dijkstra. El Check Button en el GUI del simulador "Topological Search First/Dijkstra" selecciona el algoritmo de busqueda, sin ser seleccionado utiliza el algoritmo de búsqueda primera en profundidad, seleccionando utiliza el algoritmo de Dijkstra. Una vez encontrados los nodos para ir de un nodo a otro el robot simulado utiliza un algoritmo de evasión de obstáculos y dirigirse a una fuente luminosa.

Entienda el funcionamiento de estas dos selecciones en el código de GoTo\_State\_Machine.cpp

3.- Dada la posición origen de un robot móvil  $x_0, y_0$ , y la posición final  $x_d, y_d$ , seleccionados en el GUI por el mouse encuentre:

a) Usando el algoritmo de búsqueda en profundidad, encuentre un grupo de nodos, los cuales unen el origen y el destino usando cualquiera de los mapas topológicos \*.top. Recorra estos nodos usando el algoritmo de Bug 1 desarrollado en la práctica anterior para ir de un nodo a otro.

4. - Repita el mismo procedimiento usando el algoritmo de búsqueda de Dijkstra.

5.- Incluya obstáculos desconocidos no contemplados por el planeador de movimientos, modificando el archivo del medio ambiente \*.wrl en donde se hacen las pruebas. Estos archivos se encuentran en el directorio ~/ data\_pumasimbot. En el apéndice B se muestra el archivo que contiene el medio ambiente obstacle.wrl. Combinando un algoritmo de búsqueda y los comportamientos reactivos desarrollados en las prácticas anteriores encuentre un camino de un origen a un destino evitando los obstáculos desconocidos.

## APENDICE A

### Ejemplo de Mapa Topológico \*.top

Num nodes indica el número de nodos del mapa topológico:  
( num nodes 4)

name nodes indica los nombres de los nodos, estos pueden ser números, letras, palabras, etc:  
( name nodes 1 2 3 4 )

node indica la posición en el mapa del nodo:  
( node 1 16.905001 23.813334 )  
( node 2 17.135000 26.946667 )  
( node 3 17.020000 20.131666 )  
( node 4 19.780001 20.053333 )

connection indica la distancia o peso que hay entre los nodos:  
( connection 1 2 19.126621 )  
( connection 2 1 19.126621 )  
( connection 2 3 17.171452 )  
( connection 2 4 24.633667 )  
( connection 3 2 17.171452 )  
( connection 3 4 40.814728 )  
( connection 4 3 40.814728 )

## APENDICE B

### Mapa Simbólico obstacle.wrl

En este apéndice se muestra las siguientes directivas de la representación de los mapas simbólicos:

1. Se pueden incluir comentarios o comentar una línea si se coloca un “;” al principio del renglón.

```
;* File: obstacle.wrl *
```

2. “dimensions” indica las dimensiones del medio ambiente, las cuales están indicadas en el sistema métrico decimal. En el siguiente ejemplo se presentan las dimensiones del medio ambiente “room” de 1m x 1m:

```
( dimensions room 1.000 1.000 )
```

3. “polygon” indica los vértices de un polígono, se indica si es de tipo obstáculo o pared, después viene el nombre del obstáculo y a continuación los vértices. Los vértices de los polígonos se indican con las coordenadas Xi y Yi, ordenados hacia el sentido horario de las manecillas del reloj:

```
( polygon obstacle obs1 .40 .55 .60 .55 .60 .35 .40 .35 )
```

```
( polygon wall wall1 0.0 0.0 0.0 1.0 0.01 1.0 0.01 0.0 )
```

## obstacle.wrl

```
.*****  
,  
; * File: room.wrl *  
; * Definition of the forbidden and allowed areas in the Robot's *  
; * world. These areas are derivated from the objects in the *  
; * world. *  
*****  
  
( dimensions room 1.000 1.000 )  
( polygon obstacle obs1 .40 .55 .60 .55 .60 .35 .40 .35 )  
( polygon wall wall1 0.0 0.0 0.0 1.0 0.01 1.0 0.01 0.0 )  
( polygon wall wall2 0.0 0.99 0.0 1.0 1.0 1.0 1.0 0.99 )  
( polygon wall wall2 0.99 1.0 1.0 1.0 1.0 0.0 0.99 0.0 )  
( polygon wall wall2 1.0 0.01 1.0 0.0 0.0 0.0 0.0 0.01 )
```