PRIMER TAREA PROGRAMADA, INTELIGENCIA ARTIFICIAL

JOSÉ CASTRO

Contents

1. Introducción 1
2. Especificación de la tarea a programar 1
3. Calificación 2
4. Fecha de Entrega: 28 de Setiembre 2012 2

1. Introducción

Los algoritmos de búsqueda se utilizan constantemente en aplicaciones de IA tales como robótica. En estos contextos, sin embargo, es importante adaptar dichos algoritmos a un entorno dinámico. Es común, por ejemplo, que el agente tenga que responder en un marco de tiempo limitado, que tenga que movilizarse f'ísicamente para encontrar las rutas, y que en general se encuentre trabajando *en linea*.

En esta tarea ud. debe modificar los algoritmos clásicos de búsqueda, para que trabajen en un contexto en linea.

2. Especificación de la tarea a programar

En la página del tec digital existe una aplicación en erlang que despliega una ventanan con una cuadrícula. La aplicación compila pero al hacerlo genera una lista de warnings. Esto es debido a que utiliza el móvulo `gs` el cual será eliminado de la distribución erlang en el release 16.

El programa se ejecuta con el comando:

```
1 > board:start().
```

La cuadrácula que el programa despliega corresponde al espacio en que se va a mover su agente. El punto verde corresponde al punto inicial, el punto rojo el punto final. Dichos valores puedes ser rebuscados en la cuadrácula, además de que se puede poner obstculos en el camino de su agente, y guardar las configuraciones que genere.

Para interactuar con este ambiente su programa deberá enviar mensajes al proceso `board`, los mensajes que dicho proceso entiende los puede ver en el código fuente `board.erl` en la función `eventLoop`. Estos son:

- *get_pos* se ejecuta con el comando:

  ```
  > board ! {get_pos, self()}.
  ```

  El proceso contesta enviando un mensaje, así que para verlo hay que revisar el buzón del proceso utilizando el receive
- *get_neighbors* obtiene los vecinos del nodo actual (el que se obtiene con *get_pos*. llegan en dos listas, la primer lista son los identificadores de la posición (traen la información de la cuadrículo, pero su tarea no debe manipular ni utilizar esta información). Y la segunda lista corresponde al valor de la heurística para cada uno de las posiciones que se encuentran en la lista anterior.

  Esta información debe ser utilizada por su algoritmo.

  Note que cuando ejecuta el mensaje *get_neighbors* el despliegue le cambia de color a aquellos recuadros que le enva como vecinos. Esto indica que las celdas que acaba de obtener pasan de ser celdas no visitadas, a celdas que se encuentran en la frontera o *fringe*.
- *move* este comando se ejecuta enviándole al `board` el siguiente mensaje:

  ```
  > board ! {move, {Row, Col}}.
  ```

  Este comando no responde nada, y se debe verificar con *get_pos*, el comando solo se ejecuta si la celda escogida se encuentra en el *fringe* o frontera.

Ud. debe implementar tres algoritmos de búsqueda de IA. Estos son:

(1) Greedy algorithm
(2) A^*
(3) Online Graph Prunning (artículo en la sección de artículos del tec digital)

3. Calificación

- Programa 50%
- Presentación 35%
- Documentación 5%
- Anáisis de resultados 10%

4. Fecha de Entrega: 5 de Octubre 2012