

Práctica 6- Implementación A*

1. En esta práctica vamos a implementar en LISP el algoritmo A*.

Para esta implementación, asumimos que, dado un problema concreto, tenemos definidos los siguientes elementos:

- La variable **ESTADO-INICIAL** que contiene el estado inicial.
- La función (ES-ESTADO-FINAL ESTADO) que dado un ESTADO, determina si ese estado es solución del problema o no.
- La variable **OPERADORES** que contiene la lista de operadores del problema considerado.
- Para cada OPERADOR la función (OPERADOR ESTADO-ACTUAL) que devuelve el estado obtenido aplicando el OPERADOR al ESTADO-ACTUAL, si el OPERADOR es aplicable y NIL, en caso contrario.
- La función (HEURISTICA ESTADO) que devuelve el valor de la función de evaluación heurística aplicada al ESTADO (es decir, el coste estimado para alcanzar una solución a partir del ESTADO).
- La función (COSTE-DE-APLICAR-OPERADOR ESTADO OPERADOR) que devuelve el coste de aplicar el OPERADOR al ESTADO.

Vamos a representar los nodos heurísticos por medio de estructuras que llamaremos nodos (nodos-ch) que constarán de cuatro campos:

- El estado
- Un camino (es decir, la lista de operadores necesarios para alcanzar el estado actual desde el inicial, escritos en orden inverso a como se han aplicado).
- El coste del camino actual
- El coste del camino más la heurística del nodo

Para la implementación se aconseja seguir el algoritmo en pseudocódigo que aparece en las transparencias de teoría. Además, se recomienda definir las siguientes funciones auxiliares:

- Función SUCESORES, que dado un nodo, determina los nodos sucesores de dicho nodo. Es recomendable que esta función se construya en términos de otra función auxiliar SUCESOR que dado un nodo y un operador, devuelve el nodo resultante de aplicar el operador a ese nodo o NIL si el operador no puede ser aplicado.
- Función ESTA-MEJOR, que dado un nodo N y una lista de nodos, comprueba si existe algún nodo en la lista con el mismo estado que N y tal que el coste del camino es menor o igual que el de N. En este caso, la función devuelve TRUE. Si no, devuelve NIL.
- Función ELIMINA-PEORES que dadas tres listas de nodos L0, L1 y L2, devuelve una lista con los elementos de L0 tales que no hay nodos ni en L1 ni en L2 con el mismo estado y menor coste de camino.

- Función NUEVOS-O-MEJORES-SUCESORES, que dado un nodo N y dos listas, devuelve el resultado de aplicar la función ELIMINA-PEORES a los sucesores de dicho nodo y a las dos listas.
- Función ORDENA-POR-COSTE-MAS-HEUR, que dada una lista de nodos, la reordena de forma creciente de acuerdo con los valores en el campo coste-mas-heurística.

Se recuerda que tanto la función de coste como la heurística deberán definirse para cada problema concreto.

Aplicar el algoritmo A* al problema del granjero.