**MEMORIA PRÁCTICA 2**

**Autores: Victoria Pelayo e Ignacio Rabuñal**

**Grupo: 2301 pareja 8**

**Ejercicio1:**

En este ejercicio nos piden implementar dos funciones para evaluar el valor de la heurística, según el tiempo o el precio, de un estado.

**Pseudocódigo de las funciones:**

**F-H-TIME (state sensors)**

input:

state: nombre de la ciudad (estado) actual

sensors: lista de los estados y sus correspondientes heurísticas

los elementos de la lista son (state (h-time h-price))

Output:

Res: el valor de la heurística del tiempo

Proceso:

i = 0

para i < longitud de sensors:

si state = valor-estado(sensors[i]):

devolver valor-heurisitica-time (sensors[i])

**F-H-PRICE (state, sensors)**

input:

state: nombre de la ciudad (estado) actual

sensors: lista de los estados y sus correspondientes heurísticas

los elementos de la lista son (state (h-time h-price))

Output:

Res: el valor de la heurística del precio

Proceso:

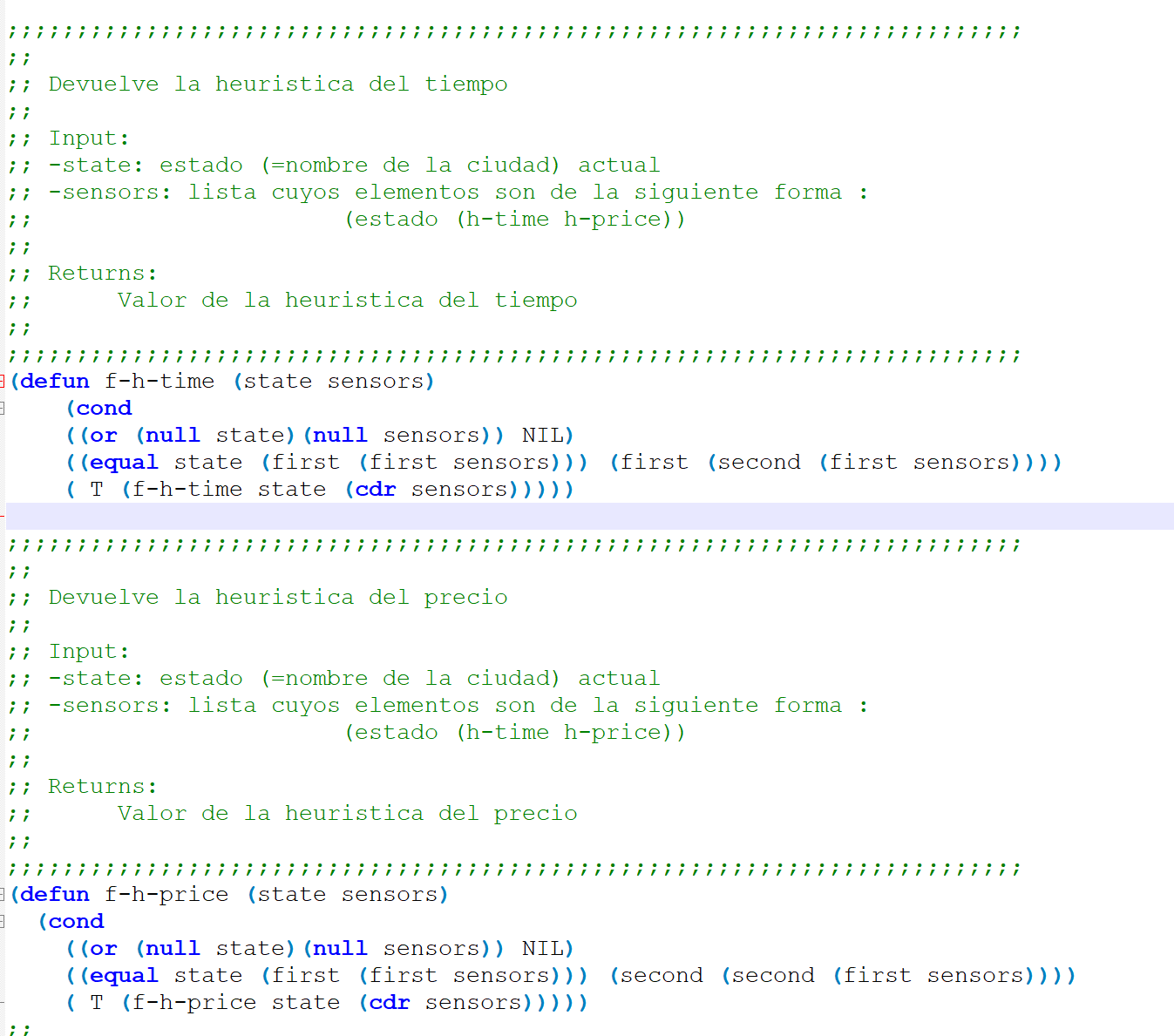
i = 0

para i < longitud de sensors:

si state = valor-estado(sensors[i]):

devolver valor-heurisitica-price (sensors[i])

Capturas del Código implementado:



**Ejemplos de ejecución ejercicio 1:**

CG-USER(21): (f-h-time 'Nantes \*estimate\*)

75.0

CG-USER(22): (f-h-time 'Marseille \*estimate\*)

145.0

CG-USER(35): (f-h-price 'Lyon \*estimate\*)

0.0

CG-USER(36): (f-h-price 'Madrid \*estimate\*)

NIL

CG-USER(37): (f-h-time 'Madrid \*estimate\*)

NIL

**Ejercicio2:**

**Ejemplos de ejecución ejercicio 2:**

CG-USER(139): (navigate-canal-time 'Avignon \*canals\*)

(#S(ACTION :NAME NAVIGATE-CANAL-TIME :ORIGIN AVIGNON :FINAL MARSEILLE :COST 35.0))

CG-USER(140): (navigate-train-price 'Avignon \*trains\* '())

(#S(ACTION :NAME NAVIGATE-TRAIN-PRICE :ORIGIN AVIGNON :FINAL LYON :COST 40.0)

#S(ACTION :NAME NAVIGATE-TRAIN-PRICE :ORIGIN AVIGNON :FINAL MARSEILLE :COST 25.0))

CG-USER(141): (navigate-train-price 'Avignon \*trains\* '(Marseille))

(#S(ACTION :NAME NAVIGATE-TRAIN-PRICE :ORIGIN AVIGNON :FINAL LYON :COST 40.0))

CG-USER(142): (navigate-canal-time 'Orleans \*canals\*)

NIL

CG-USER(143): (navigate-canal-time 'Lyon \*canals\*)

(#S(ACTION :NAME NAVIGATE-CANAL-TIME :ORIGIN LYON :FINAL NANCY :COST 150.0) #S(ACTION :NAME NAVIGATE-CANAL-TIME :ORIGIN LYON :FINAL ROENNE :COST 40.0)

#S(ACTION :NAME NAVIGATE-CANAL-TIME :ORIGIN LYON :FINAL AVIGNON :COST 50.0))

CG-USER(144): (navigate-train-time 'Lyon \*trains\* \*forbidden\*)

(#S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN LYON :FINAL TOULOUSE :COST 60.0) #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN LYON :FINAL ROENNE :COST 18.0))

CG-USER(145): (navigate-train-price 'Lyon \*trains\* \*forbidden\*)

(#S(ACTION :NAME NAVIGATE-TRAIN-PRICE :ORIGIN LYON :FINAL TOULOUSE :COST 95.0) #S(ACTION :NAME NAVIGATE-TRAIN-PRICE :ORIGIN LYON :FINAL ROENNE :COST 25.0))

**Ejemplos de ejecución ejercicio 3**

CG-USER(104): (f-goal-test node-calais '(Calais Marseille) '(Paris Nancy))

T

CG-USER(105): (f-goal-test node-calais '(Calais Marseille) '(Paris Limoges))

NIL

CG-USER(106): (f-goal-test node-paris '(Calais Marseille) '(Paris))

NIL

CG-USER(107): (f-goal-test node-calais '(Calais Marseille) '(Paris Nancy))

T

CG-USER(108): (f-goal-test nil '(hola hola) '(ninguno))

NIL

\*Como decision hemos tomado que node-nevers no puede ser un nodo de estado final, pues representa que no ha habido ningún camino

CG-USER(109): (f-goal-test node-nevers nil nil)

NIL

\*\*function auxiliar

CG-USER(110): (get-camino node-calais nil)

(CALAIS REIMS NANCY PARIS)

CG-USER(111): (get-camino node-nevers nil)

NIL

CG-USER(116): (get-camino node-paris '(madrid barcelona))

(PARIS MADRID BARCELONA)5

**Ejemplos de ejecución ejercicio 4:**

CG-USER(144): (f-search-state-equal node-calais node-calais-2 '())

T

CG-USER(145): (f-search-state-equal node-calais node-calais-2 '(Reims))

NIL

CG-USER(146): (f-search-state-equal node-calais node-calais-2 '(Nevers))

T

CG-USER(147): (f-search-state-equal node-nancy node-paris '())

NIL

\*\*Ejemplos de la funcion auxilair

CG-USER(148): (no-visitados nil '(madrid) nil)

(MADRID)

CG-USER(149): (no-visitados '(madrid barcelona) '(madrid paris lyon) nil)

(PARIS LYON)

CG-USER(150): (no-visitados '(madrid) nil nil)

NIL

**Ejemplos de ejecución ejercicio 6**

\*\*Ejemplo de la función auxiliar expand-node-action

(Con este ejemplo comprobamos que esta función funcione bien)

Hemos ido pidiendo cada uno de los atributos de nodo porque en la terminal no salía el nodo al completo

CG-USER(95): (node-f (expand-node-action node-marseille-ex6 #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0) \*travel-fast\*))

205.0

CG-USER(96): (node-g (expand-node-action node-marseille-ex6 #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0) \*travel-fast\*))

75.0

CG-USER(97): (node-h (expand-node-action node-marseille-ex6 #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0) \*travel-fast\*))

130.0

CG-USER(98): (node-depth (expand-node-action node-marseille-ex6 #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0) \*travel-fast\*))

13

CG-USER(99): (node-action (expand-node-action node-marseille-ex6 #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0) \*travel-fast\*))

#S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0)

CG-USER(100): (node-parent (expand-node-action node-marseille-ex6 #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0) \*travel-fast\*))

#S(NODE :STATE MARSEILLE :PARENT NIL :ACTION NIL :DEPTH 12 :G ...)

CG-USER(101): (node-state (expand-node-action node-marseille-ex6 #S(ACTION :NAME NAVIGATE-TRAIN-TIME :ORIGIN MARSEILLE :FINAL TOULOUSE :COST 65.0) \*travel-fast\*))

TOULOUSE

Ejemplos de ejecución de la función expand-node:

**\*ejemplo del enunciado**

CL-USER(53): (defparameter lst-nodes-ex6

(expand-node node-marseille-ex6 \*travel-fast\*))

LST-NODES-EX6

CL-USER(54): (print lst-nodes-ex6)

(#S(NODE :STATE TOULOUSE

:PARENT #S(NODE :STATE MARSEILLE

:PARENT NIL

:ACTION NIL

:DEPTH 12

:G 10

:H 0

:F 20)

:ACTION #S(ACTION :NAME NAVIGATE-TRAIN-TIME

:ORIGIN MARSEILLE

:FINAL TOULOUSE

:COST 65.0)

:DEPTH 13

:G 75.0

:H 130.0

:F 205.0))

(#S(NODE :STATE TOULOUSE

:PARENT #S(NODE :STATE MARSEILLE

:PARENT NIL

:ACTION NIL

:DEPTH 12

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-TRAIN-TIME

:ORIGIN MARSEILLE

:FINAL TOULOUSE

:COST 65.0)

:DEPTH 13

:G ...))

**\*introduciendo un problema no valido**

CL-USER(77): (expand-node node-paris nil)

NIL

**\*introduciendo un nodo no valido**

CL-USER(78): (expand-node nil \*travel-cheap\*)

NIL

**\*Paris con la heuristica sobre el precio.**

**Hemos elegido parís porque tiene muchas conexiones con otras ciudades.**

CL-USER(79): (expand-node node-paris \*travel-cheap\*)

(#S(NODE :STATE REIMS

:PARENT #S(NODE :STATE PARIS

:PARENT #S(NODE

:STATE

NEVERS

:PARENT

NIL

:ACTION

NIL

:DEPTH

0

:G

...)

:ACTION NIL

:DEPTH 0

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-CANAL-PRICE

:ORIGIN PARIS

:FINAL REIMS

:COST 10.0)

:DEPTH 1

:G ...)

#S(NODE :STATE NANCY

:PARENT #S(NODE :STATE PARIS

:PARENT #S(NODE

:STATE

NEVERS

:PARENT

NIL

:ACTION

NIL

:DEPTH

0

:G

...)

:ACTION NIL

:DEPTH 0

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-CANAL-PRICE

:ORIGIN PARIS

:FINAL NANCY

:COST 10.0)

:DEPTH 1

:G ...)

#S(NODE :STATE CALAIS

:PARENT #S(NODE :STATE PARIS

:PARENT #S(NODE

:STATE

NEVERS

:PARENT

NIL

:ACTION

NIL

:DEPTH

0

:G

...)

:ACTION NIL

:DEPTH 0

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-TRAIN-PRICE

:ORIGIN PARIS

:FINAL CALAIS

:COST 60.0)

:DEPTH 1

:G ...)

#S(NODE :STATE NANCY

:PARENT #S(NODE :STATE PARIS

:PARENT #S(NODE

:STATE

NEVERS

:PARENT

NIL

:ACTION

NIL

:DEPTH

0

:G

...)

:ACTION NIL

:DEPTH 0

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-TRAIN-PRICE

:ORIGIN PARIS

:FINAL NANCY

:COST 67.0)

:DEPTH 1

:G ...)

#S(NODE :STATE NEVERS

:PARENT #S(NODE :STATE PARIS

:PARENT #S(NODE

:STATE

NEVERS

:PARENT

NIL

:ACTION

NIL

:DEPTH

0

:G

...)

:ACTION NIL

:DEPTH 0

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-TRAIN-PRICE

:ORIGIN PARIS

:FINAL NEVERS

:COST 75.0)

:DEPTH 1

:G ...)

#S(NODE :STATE ORLEANS

:PARENT #S(NODE :STATE PARIS

:PARENT #S(NODE

:STATE

NEVERS

:PARENT

NIL

:ACTION

NIL

:DEPTH

0

:G

...)

:ACTION NIL

:DEPTH 0

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-TRAIN-PRICE

:ORIGIN PARIS

:FINAL ORLEANS

:COST 38.0)

:DEPTH 1

:G ...)

#S(NODE :STATE ST-MALO

:PARENT #S(NODE :STATE PARIS

:PARENT #S(NODE

:STATE

NEVERS

:PARENT

NIL

:ACTION

NIL

:DEPTH

0

:G

...)

:ACTION NIL

:DEPTH 0

:G ...)

:ACTION #S(ACTION :NAME NAVIGATE-TRAIN-PRICE

:ORIGIN PARIS

:FINAL ST-MALO

:COST 70.0)

:DEPTH 1

:G ...))