Técnicas de los Sistemas Inteligentes

Práctica 3: Planificación Clásica (PDDL)

Rafael Vázquez Conejo

Mayo 2020

Índice

1. Introducción

El objetivo de esta práctica es el uso del lenguaje “pddl” para la representación de dominios de planificación clásicos. Para ello, a continuación se presenta la resolución de seis problemas de este tipo. Cada problema tiene definido su propio archivo dominio y problema.

2. Ejercicios

Ejercicio 1:

Diseñaremos nuestra primera versión del dominio de nuestro problema. Para ello comenzamos definiendo los diferentes tipos que encontraremos, unidades, edificios y localizaciones. Tras ello las constantes del problema, las cuales serán:

- VCE como unidades.
- Centro de mandos y barracones como edificios
- Minerales y gas como recursos

El siguiente paso es la definición de los predicados, para ello hemos incluido los predicados pedidos en el enunciado del ejercicio, estos son:

- localizacionEdificio → determina si un edificio está en una localización concreta.
- localizacionUnidad → determina si una unidad está en una localización concreta.
- hayCamino → indica que existe un camino entre 2 localizaciones
- asignarNodoRecurso → asigna un nodo de un recurso indicado a una localización concreta.
- extrayendo → indica si un VCE está extrayendo un recurso.
- recursoNecesario → indica que recurso necesita un edificio para ser construido

Además de los predicados indicados también he definido otros que me permiten controlar ciertas situaciones y parámetros.

- ocupadoTrabajando → indica que una unidad está ya asignada en un nodo de recurso.
- tipoUnidad → determina el tipo de una unidad concreta.
- tipoEdificio → determina el tipo de un edificio concreto.
- tipoRecurso → determina el tipo de un recurso concreto.
- recursosObtenidos → me permite saber que he conseguido un recurso de un tipo concreto.

Tras ello defino las diferentes accciones indicadas:

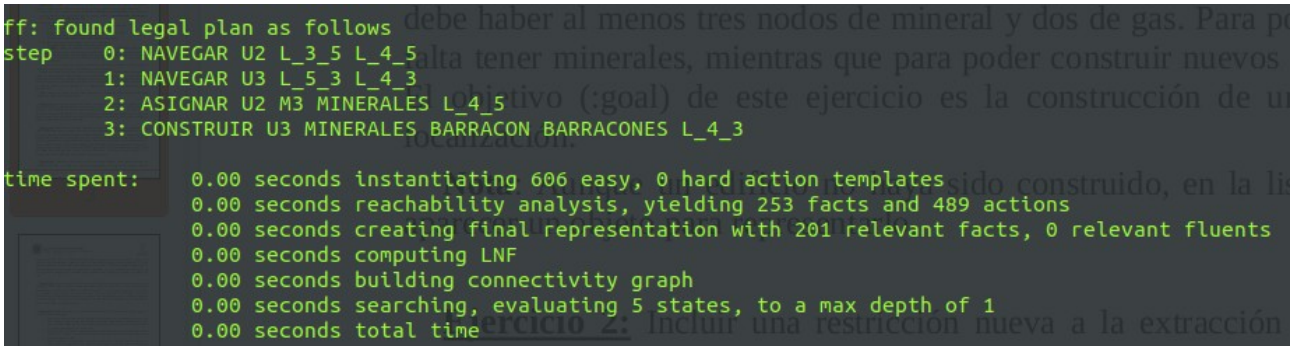
- Navegar → comprueba si se encuentran conectadas la localización de una unidad con otra localización, si esto es así, la unidad se moverá de la localización actual a la nueva localización.

- Asignar→ si una unidad no se encuentra trabajando y se encuentra en la misma posición que un nodo de recursos, se le asigna a esta unidad este nodo de trabajo y extrae el recurso que forme a dicho nodo. Esto provoca que tengamos recursos del material extraído.
- Construir→ para poder construir una unidad no debe estar trabajando y debe encontrase en una localización dónde no hay ningún edificio. Si se poseen los recursos necesario para la construcción del edificio se construye en dicha localización el edificio.

De esta forma hemos finalizado la definición del dominio, y tras ello definimos nuestra primera versión del problema. En ella definimos una matriz 5x5, además de los elementos indicados en el enunciado que deben presentarse en el mapa. Para la conexión de la matriz utilizaré el predicado "hayCamino", que me permitirá indicar los cuadrantes que se encuentran conectados. Una vez defino los tipos de elementos, los recursos necesarios para cada edificio y la localización en el mapa de los elementos necesarios obtengo la siguiente matriz:

unidad_1		nodo_mineral_1		nodo_gas_2
nodo_gas_1		centro_de_mando		unidad_2
nodo_mineral_2		barracón		nodo_mineral_3
		unidad_3		

El objetivo será construir un Barracón, que necesita Minerales para su construcción:



Ejercicio 2:

En este ejercicio partimos del mismo dominio que en el anterior, sólo que con algunas modificaciones. Estas modificaciones estarán centradas en hacer que para poder asignar un nodo del recurso "gas" primero se deberá construir un Extractor sobre dicho nodo. Para ello lo primero que haré será crear una nueva constante con el nuevo edificio:

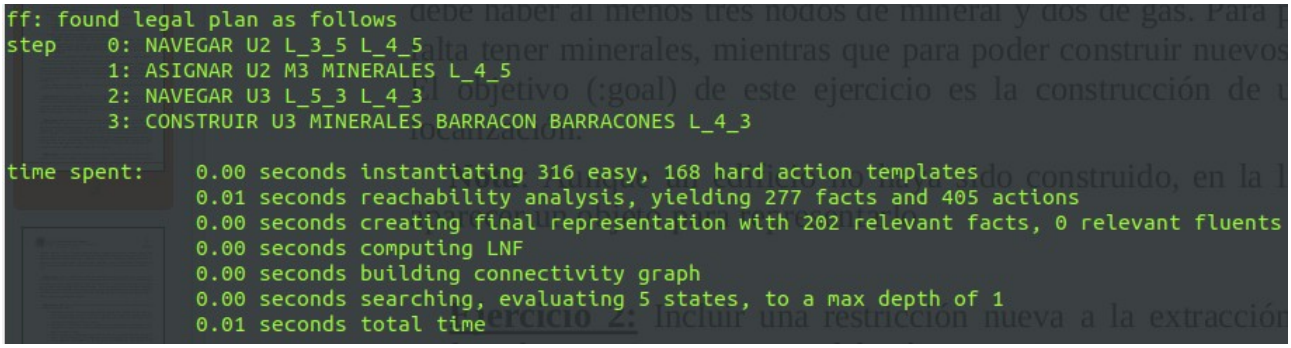
- Extractor como edificio.

Añadiré un nuevo predicado que me permita controlar la existencia o no de un extractor en una localización concreta (existeExtractor). A continuación modificaré la acción de Asignar, de forma que solamente podré asignar un trabajador de tipo VCE a un nodo de gas cuando exista un extractor en esa misma localización.

Para finalizar con el dominio, modifico la acción Construir, de manera que si el tipo de edificio a construir es un Extractor deberá existir un nodo de recursos de tipo gas en la localización, si no es el

caso no se podrá construir. Si es el caso de haber construido un Extractor se activará el predicado existeExtractor en dicha posición.

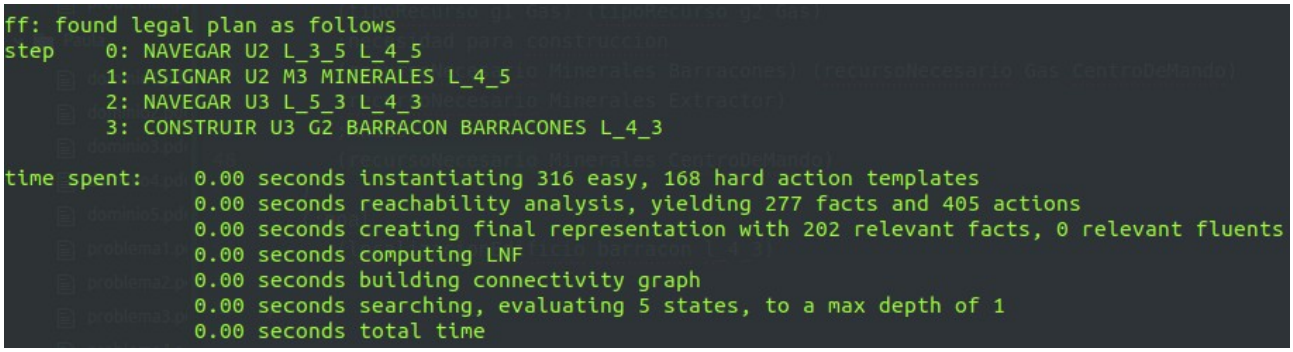
En el problema añadimos un nuevo tipo de objeto del tipo edificio correspondiente a nuestro Extractor. Tras ello lo inicializamos asignándole el tipo mediante “tipoEdificio”. El objetivo será el mismo que en el ejercicio anterior, contruir un barracón.



Ejercicio 3:

La idea de este ejercicio es permitir que al construir un edificio se tenga en cuenta que este puede necesitar más de un tipo de recursos. Para ello en el dominio modificaré la acción de Construir, de forma que añadiré una nueva precondición que mediante “forall” recorrerá todos los tipos de recursos y si se trata de un recurso necesario para la construcción del edificio actual, se observará si poseemos dicho recurso. De esta forma sólo podremos construir un edificio si tenemos todos los tipos de recursos que requiere.

En archivo del problema solamente añadiré un nuevo recursoNecesario para el Centro de Mando, necesitará tanto gas como minerales para su construcción. El objetivo será el mismo que en el ejercicio anterior.



Ejercicio 4:

En este ejercicio añadiremos dos nuevos tipos de unidades, además de la acción Reclutar, que nos permitirá crear nuevas unidades. Comenzando por el dominio lo primero será añadir dos nuevas constantes:

- Marine como unidad.
- Segador como unidad.

Tras ello creo dos predicados nuevos, uno de ellos para crear una nueva unidad en un edificio concreto que permita la creación de dicha unidad (creaUnidad), y un predicado que controlo los recursos necesarios para crear una unidad (recursoNecesarioUnidad), similar a “recursoNecesario” utilizado para los edificios.

El siguiente paso será la creación de la acción Reclutar, la cual para que se desarrollo tendremos en cuenta que no existe ya la unidad que vayamos a crear, que el edificio en la localización concreta sea el tipo de edificio que permita la creación del tipo de unidad concreta, y finalmente, como hicimos en el ejercicio anterior, repasaremos todos los recursos necesario para la creación de la unidad deseada, y que poseemos todos ellos. Si se dan estas situaciones se podrá crear una unidad en la misma localización del edificio que la crea.

Respecto al archivo del problema, añadiremos los objetos de tres unidades nuevas, dos marines y un segador. Inicializaremos sus tipos a unidades y estableceré los recursos necesarios para la creación de cada unidad, mediante “recursoNecesarioUnidad”. Finalmente mediante “creaUnidad” indicaré los tipos de edificios que permiten la creación de cada tipo de unidad.

Importante mencionar que esta vez en el mapa solamente habrá un trabajado al comienzo.En este caso el objetivo cambia, esta vez será la creación de un marine y un segador que se sitúen en la misma localización y otro marine que se sitúe en una localización distinta.

```
ff: found legal plan as follows
step    0: NAVEGAR U1 L_5_3 L_5_2
        1: NAVEGAR U1 L_5_2 L_4_2
        2: NAVEGAR U1 L_4_2 L_4_1
        3: ASIGNAR U1 M2 MINERALES L_4_1
        4: RECLUTAR U2 VCE CENTROMANDO CENTRODEMANDO L_3_3
        5: NAVEGAR U2 L_3_3 L_4_3
        6: NAVEGAR U2 L_4_3 L_5_3
        7: RECLUTAR U3 VCE CENTROMANDO CENTRODEMANDO L_3_3
        8: NAVEGAR U3 L_3_3 L_3_2
        9: NAVEGAR U3 L_3_2 L_3_1
       10: CONSTRUIR U3 G2 EXTRACTOR EXTRACTOR L_3_1
       11: ASIGNAR U3 G1 GAS L_3_1
       12: NAVEGAR U2 L_5_3 L_5_4
       13: CONSTRUIR U2 G2 BARRACON BARRACONES L_5_4
       14: RECLUTAR MA1 MARINE BARRACON BARRACONES L_5_4
       15: NAVEGAR MA1 L_5_4 L_5_3
       16: NAVEGAR MA1 L_5_3 L_5_2
       17: RECLUTAR MA2 MARINE BARRACON BARRACONES L_5_4
       18: RECLUTAR S1 SEGADOR BARRACON BARRACONES L_5_4

time spent:    0.02 seconds instantiating 711 easy, 318 hard action templates
               0.00 seconds reachability analysis, yielding 582 facts and 792 actions
               0.00 seconds creating final representation with 427 relevant facts, 0 relevant fluents
               0.00 seconds computing LNF
               0.00 seconds building connectivity graph
               0.01 seconds searching, evaluating 56 states, to a max depth of 8
               0.03 seconds total time
```

Ejercicio 5:

En este ejercicio añadiremos una nueva acción, Investigar, la cual sólo se podrá desarrollar si tenemos construido un nuevo tipo de edificio, Bahía de Ingeniería, acción que nos permitirá investigar “Impulsor Segador”, investigación necesario para poder crear unidades del tipo segador.

Para ello en mi dominio añadiré un nuevo tipo, “investigaciones”, además de dos nuevas constantes:

- BahiaDeIngenieria como edificio.

- ImpulsorSegador como investigación.

Necesitaré una serie de predicados que me permitan controlar las investigaciones:

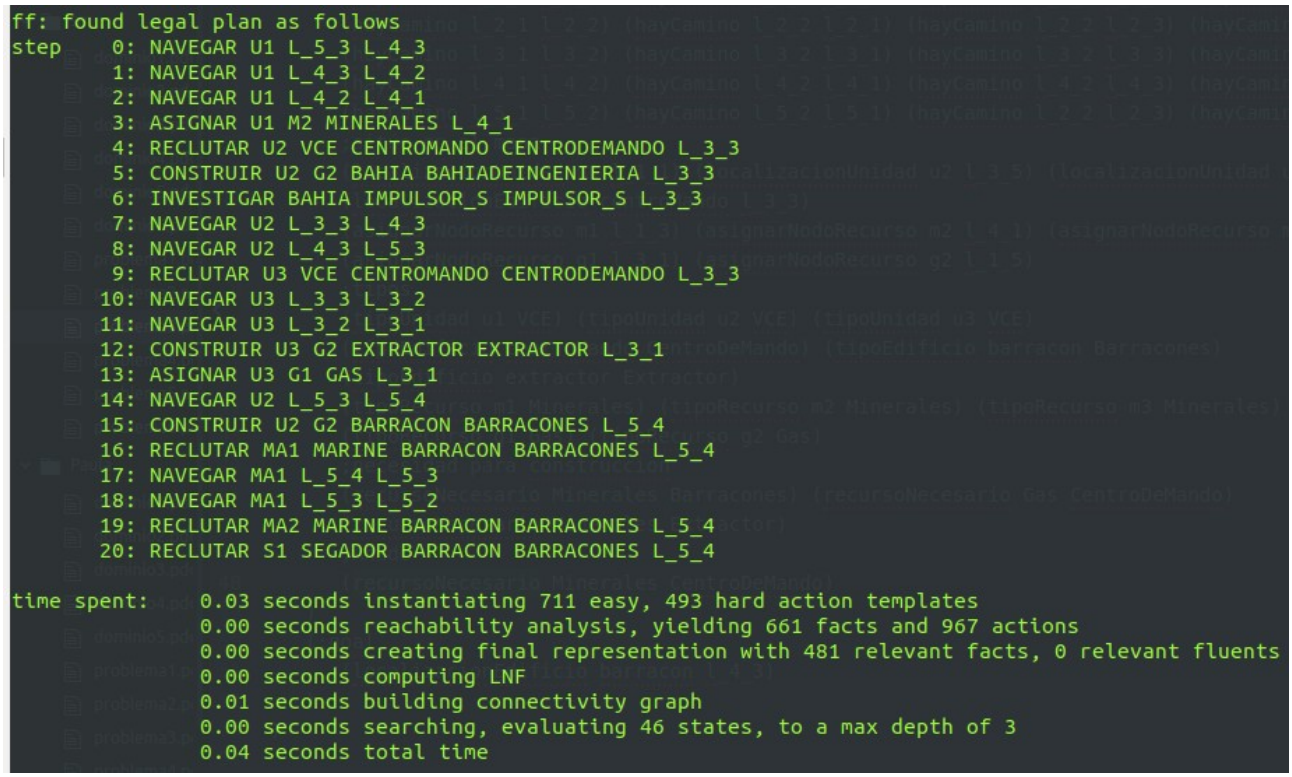
- investigacionCompleta→ me permitirá controlar cuando se ha llevado a cabo un tipo de investigación y cuando esta aún no se ha producido.
- TipoInvestigacion→ determina el tipo de una investigación concreta.
- RecursoNecesarioInvestigar→ indica que recurso necesita una investigación para realizarse.

Crearemos a continuación la nueva acción, Investigar. Para que se desarrolle tendrá que existir un edifico de tipo Bahía de Investigación en una localización concreta, además de que la investigación no se haya completado todavía (ya que no tiene sentido investigar lo mismo varias veces). Finalmente, igual que en los ejercicios anteriores, repasaremos todos los recursos necesario para realizar la investigación deseada, y que poseemos todos ellos. Una vez se han dado todas estas condiciones se establecerá la investigación como completa, y se podrá reclutar segadores.

El siguiente paso será modificar la acción de Reclutar, de forma que si la intención es crear una unidad de tipo segador se deberá dar como precondición que se haya completado la investigación de ImpulsorSegador. Si no es el caso, no se podrá crear dicha unidad.

Una vez hemos modificado nuestro dominio, pasaremos al archivo del problema, en el que añadiremos dos nuevos objetos para el nuevo edificio, Bahía, y la investigación de Impulsor Segador. Inicializaremos los tipos de estas nuevos objetos, añadiremos los recursos necesarios para la construcción del nuevo edificio mediante “recursoNecesario”, y los recursos necesarios para la investigación, mediante “recursoNecesarioInvestigar”

El objetivo será el mismo que en el ejercicio anterior, la creación de un marine y un segador que se sitúen en la misma localización y otro marine que se sitúe en una localización distinta.



Ejercicio 6:

La idea de este ejercicio será ir acumulando un número de recursos de cada tipo, pues ahora los edificios, unidades e investigaciones tienen un coste número de recursos. Además con el extra de tener un límite de recursos totales que podemos almacenar, límite que aumentaremos con un nuevo edificio, Depósito.

Comenzaremos definiendo una nueva constante con nuestro nuevo edificio:

- Depósito como edificio.

Para desarrollar este ejercicio tendremos que establecer funciones que nos permitan ir incrementando o disminuyendo el valor de ciertas variables. Estableceremos varias:

- cuentaRecursos → cantidad de recursos de un tipo que poseo.
- cuentaTrabajadores → cantidad de trabajadores en un nodo de recursos concreto.
- cuentaLimite → límite de almacenamiento de un recurso concreto.
- costeEdificios → cantidad de recursos de un tipo concreto necesarios para construir un edificio.
- costeUnidad → cantidad de recursos de un tipo concreto necesarios para construir una unidad.
- costeInvestigacion → cantidad de recursos de un tipo concreto necesarios para hacer una investigación.

Modifico Asignar de manera que ya no indicará que tenemos el recurso que se le ha asignado al trabajador, esta acción tendrá las mismas precondiciones, pero el efecto esta vez no será darnos un recurso para su uso, sino que aumentará en 1 el número de trabajadores en un tipo de nodo de recurso concreto cada vez que se desarrolle. Seguirá indicando que la unidad asignada se encuentra trabajando y extrayendo del nodo concreto.

Modifico Construir de manera que cada vez que desee construir un edificio comprobaré si poseo los recursos necesarios para ello. Gracias a la función “forall” recorreré todos los tipos de recursos y veré si tengo una cantidad mayor o igual a la necesaria mediante la función cuentaRecursos. Si tengo los recursos necesarios construiré el edificio y restaré a cuentaRecursos los recursos de cada tipo que han sido utilizados. Además añadiré que si se trata de un Depósito aumente la capacidad de almacenamiento de cada recurso en 100, es decir, aumento cuentaLimite de cada tipo de recurso en 100.

Tanto en la acción Reclutar e Investigar realizo lo comentado en el punto anterior en Construir, cada vez que desee reclutar una unidadad o investigar comprobaré si poseo los recursos necesarios para ello. Gracias a la función “forall” recorreré todos los tipos de recursos necesarios y veré si tengo una cantidad mayor o igual a la necesaria mediante la función cuentaRecursos. Si tengo los recursos necesarios realizaré la acción y restaré a cuentaRecursos los recursos de cada tipo que han sido utilizados.

Añadiré la acción Desasignar, la cual permite que un trabajador VCE que se encuentra extrayendo un nodo de recursos de un tipo concreto deje de hacerlo pasando a estar libre y dejando de extraer dicho recurso , además de reducir el número de trabajadores en ese nodo de recursos en uno.

Añadiré una nueva acción Recolectar, con la que un trabajador que se encuentre trabajando podrá recolectar de un recurso si el almacén no está lleno, esto hará que aumente la cantidad de recursos de un tipo de material.

En el archivo del problema crearé un nuevo objeto para deposito, al que le asignaré su tipo y recursos necesarios para su construcción. Añadiré para todos los edificios, unidades e investigaciones su coste, mediante costeEdificios, costeUnidad, y costeInvestigacion respectivamente. Tras ello inicializo cuentaLimite de cada recurso a 100, cuentaRecursos de cada recurso a 0 y cuentaTrabajadores a 0.

El objetivo será el mismo que en el ejercicio anterior, la creación de un marine y un segador que se sitúen en la misma localización y otro marine que se sitúe en una localización distinta.

Sin embargo no he sido capaz de obtener una solución correcta al problema, obtengo una solución que no tiene en cuenta los recursos de gas. He intentado solucionar el error y lo he acotado bastante, sabiendo que se debe a un error en la acción Recolectar, seguramente por un fallo a la hora de controlar el límite del almacén. He intentado arreglarlo añadiendo un nuevo predicado que me controle el límite de cada almacén pero me tarda mucho tiempo, seguramente por algún erro mio.

3. Bibliografía

http://users.cecs.anu.edu.au/~patrik/pddlman/writing.html

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiN8ZaU9N3pAhVvBWMBHTIPC2EQFjABegQIARAB&url=http%3A%2F%2Fdecsai.ugr.es%2F~faro%2FPractica2%2FLinkedDocuments%2FManual-HTN-PDDL.pdf&usg=AOvVaw3ca-I16rpaImYMjTVdxkID