



**UNIVERSIDAD
DE GRANADA**

TÉCNICAS DE LOS SISTEMAS INTELIGENTES

Práctica 3, Grupo 3.
StarCraft en PDDL

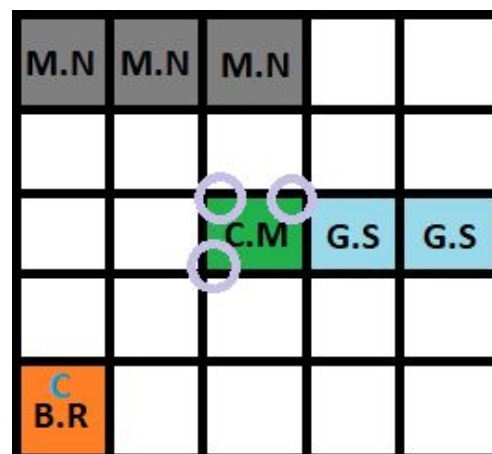
Lukas Häring García

1. Ejercicio 1.

Los predicados utilizados para esta práctica son los especificados en el enunciado de la misma, además se ha utilizado unos predicados que especifican el tipo ("EsTipoX") de las variables genéricas, por ejemplo, un edificio ("EsTipoE"), puede ser un Centro de Mando o un Barracón ("Asociación Variable-Constante"). De la misma forma, para identificar una variable en una localización, utilizaremos "EnX" "Tipo variable" "Localización". Un camino es definido por dos localizaciones, para ello he utilizado un plugin del editor online "PDDL Editor" que crea el mapa de manera automática. Un predicado ("Necesita TipoEdificio Recurso") para evitar los condicionales y especificarse estas propiedades en el problema. Otro predicado que asocia un recurso con un nodo (localización).

- **Navegar:** Requiere que la unidad no se encuentre "Extrayendo", una vez encontrada la localización "EnU(nidad)", miraremos un nodo adyacente, el efecto será modifica dicho hecho "EnU" a la nueva localización y "eliminar" el rastro ("EnU") anterior.
- **Asignar "Extraer":** Comprobamos que se trata de una entidad VCE, aunque se en el primer ejercicio solo exista este tipo de unidad, nos será útil para el futuro. Buscamos un recurso en una localización y el tipo de recurso, además debemos asegurarnos que la entidad esté en la localización del recurso. El resultado es un hecho que indica que la unidad está extrayendo y que tenemos un recurso que es el que hemos extraído.
- **Construir.** Para construir debemos encontrar una unidad que no se encuentre ocupada ("no Extrayendo"), buscaremos una variable Edificio que no se encuentre construida y la localización asociada, en el problema se ha especificado las posiciones de los elementos donde van a construirse dichos edificios. Comprobaremos el tipo de edificio a construir y el recurso necesitado y lo tengamos. El resultado será una etiqueta que indica que este edificio ha sido construido.

1. El mapa representa el archivo del problema .
2. M.N Representa la localización de las menas.
3. C.M Representa el centro de mando. En la misma localización, 3 VCE.
4. G.S El Gas
5. B.R Barracón a construir, definido en el problema.



El objetivo del programa es la "Construcción" de un barracón (variable).

2. Ejercicio 2.

Se trata de una modificación simple en el problema, se añade una nueva constante TipoEdificio para mantener información sobre el extractor. En el problema creamos 2 variables para los 2 nodos Gas, además situamos sobre ambos nodos los extractores, pero estos **no** van a estar contruidos, serán contruidos en la resolución del problema. La modificación en el dominio es simple, añadimos una implicación utilizando el paquete “:adl” que se convierte en un condicional, aunque $a \Rightarrow b \equiv \neg a \vee b$, Si hay Gas, entonces debe existir un edificio contruido en la misma posición y además debe ser de tipo Extractor.

Recurso Gas \Rightarrow Hay Extractor Contruido donde el Gas

Además como se indica, para construir un extractor hace falta mineral, por lo que añadiremos un hecho en el problema que indique “Necesita Extractor Mineral” para así poder construirse.

(navegar unidad1 l3_3 l3_2)
(navegar unidad1 l3_2 l3_1)
(asignar unidad1 minerales3 mineral l3_1)
(navegar unidad2 l3_3 l4_3)
(construir extractor2 extractor unidad2 l4_3 mineral)
(asignar unidad2 gas2 gas l4_3)

Figura 1: Resultado de hacer “HayRecurso Gas”

Aunque el “goal” de este ejercicio era el mismo que el ejercicio 1, para comprobar que funciona correctamente, se ha modificado el goal para coger Gas.

Observamos que este funciona, la unidad 1 viaja hasta el mineral, se ocupa y la unidad 2 se encarga de construir un extractor sobre la posición del gas, (4, 3) y finalmente lo recolecta.

3. Ejercicio 3.

En este problema tenemos más de un material para construir un tipo de edificio, para ello, vamos a utilizar la regla creada en el ejercicio 1. “Necesita TipoEdificio Recurso” para indicar también que necesita otro recurso, por ejemplo “Necesita CentroDeMando Gas” y “Necesita CentroDeMando Mineral”, como se indica en la práctica.

En la construcción de un edificio, vamos a eliminar la regla (HayRecurso) ya que nos limitaba a la construcción de un edificio con un único recurso, para ello, vamos a mirar si para todo tipo de recurso existente, si es necesario, entonces lo tenemos. De manera matemática.

$$\forall TipoRecurso (Necesita(TipoRecurso) \Rightarrow HayRecurso(TipoRecurso))$$

Si esto es cierto, entonces podremos construir el edificio requerido. Vemos que también he utilizado la implicación “:adl” aunque pudiera haberse realizado con la equivalencia lógica.

(navegar unidad1 l3_3 l3_2)
(navegar unidad1 l3_2 l3_1)
(asignar unidad1 minerales3 mineral l3_1)
(navegar unidad2 l3_3 l4_3)
(construir extractor2 extractor unidad2 l4_3)
(asignar unidad2 gas2 gas l4_3)
(construir centro centrodemandando unidad3 l3_3)

Figura 2: Resultado tras hacer “Construir base”

Aunque de manera equivalente al anterior, debemos construir un barracón y no se muestra el resultado del ejercicio, se ha modificado para “Construir la primera base”. Vemos que ciertamente la primera unidad recogerá mineral, que será utilizado para la creación del extractor (Véase ejercicio 2) y finalmente una vez recolectado, como la unidad 3 se encuentra sobre la posición donde se requiere construir la base, especificado en el problema, se construye utilizando gas y mineral.

4. Ejercicio 4.

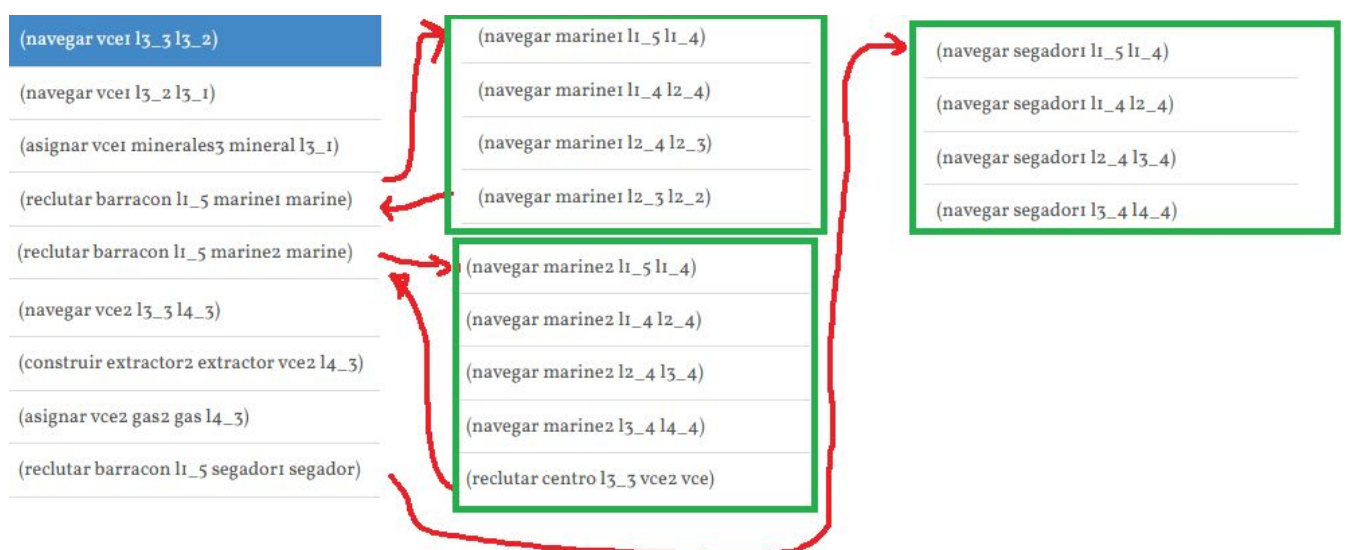
En este ejercicio se aplicarán cambios fuertes, en primer lugar añadimos dos nuevos tipos de unidades en TipoUnidad, Marines y Segadores. Un nuevo predicado “Necesita TipoUnidad Recurso” ya que ahora podremos generar nuevas unidades, utilizando o no varios recursos.

Este nuevo predicado será utilizado en la definición del problema, añadiendo los requisitos necesarios para la creación de las unidades expuestas en el enunciado de la práctica. Además un predicado que diferencia entre una unidad reclutada (que será además añadida a las demás acciones que hacen uso de una unidad y para funcionar, deban estar reclutada).

Deberemos crear una nueva acción:

- **Reclutar:** Como se indica en el enunciado, debemos comprobar el tipo de edificio desde la que se va a reclutar una unidad ya que un centro de mando podrá reclutar un VC, por lo que deberá buscar un VCE en la lista de candidatos **no reclutado** y si este fuera un barracón, buscar de la lista o un Marine o un Segador, **no reclutado**. Recopilamos la ubicación del edificio así como la información requerida para el reclutamiento de la unidad, finalmente, utilizaremos de manera equivalente al ejercicio 3, una comprobación de que tenemos los recursos requeridos para la “creación” (reclutamiento) de este. El resultado será el reclutamiento del mismo (se añade el predicado “Reclutada Unidad” y se mueve la localización de la unidad a la posición del edificio).

El problema ahora mantendrá una “lista” de unidades (variables) no reclutadas para su uso, en cuestión, 4 VCEs, 2 Marines y 1 Segador, el cual según el problema, solo existe un único VCE construido en el comienzo. El goal como su propio nombre indica, será mantener un marine, “marine1” por ejemplo, en la localización (2,2) Marine 2 (4,4), Segador (4,4).



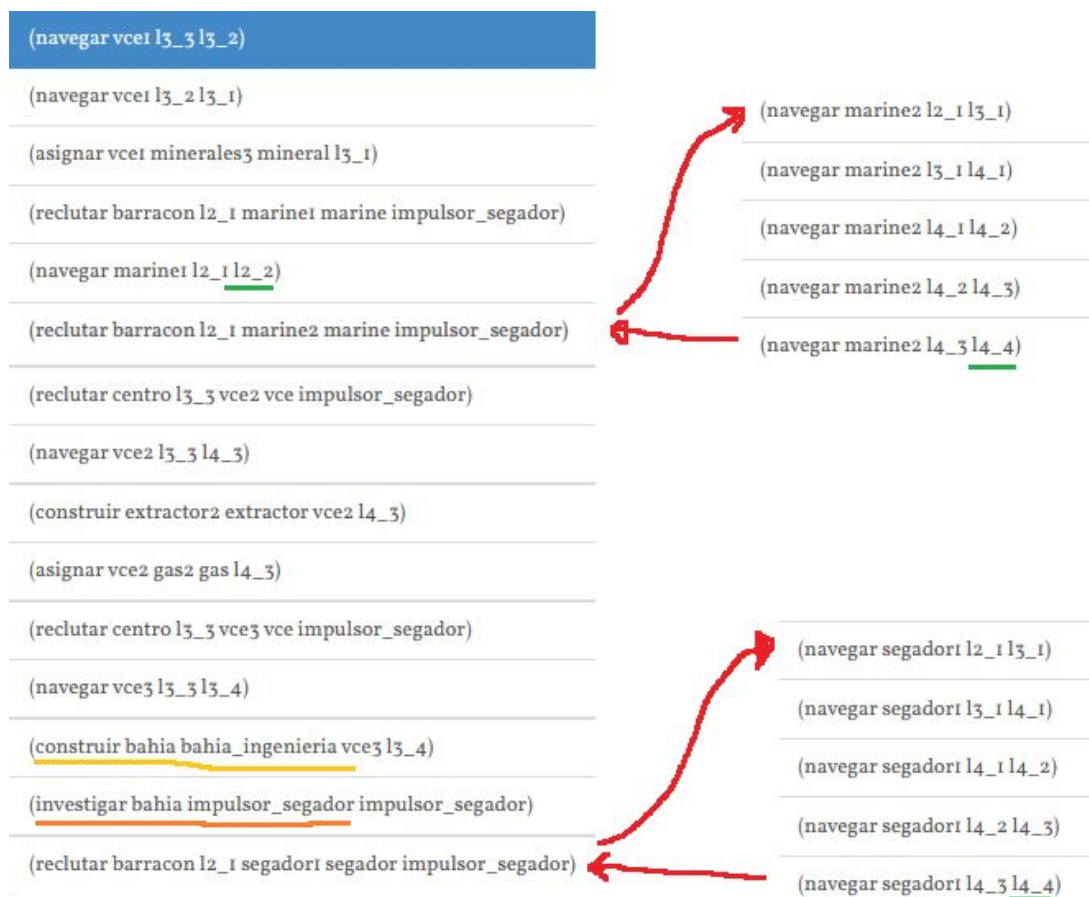
5. Ejercicio 5.

Crearemos una constante Impulsor_Segador para el tipo “TipoInvestigacion” y junto con otro tipo investigación. De esto, un predicado “EsTipoI” y dos nuevos que realizan funciones similares a los ejercicios anteriores “InvestigacionRequiere” y “Investigado Investigacion”, ya que como se indica en el foro, para realizar una investigación para el impulsor, hace falta gas y mineral.

Por otro lado, necesitamos un nuevo edificio, “Bahia_ingenieria”, que será necesaria para la nueva acción:

- **Investigar:** Como hemos comentado, hace falta de un edificio de tipo Bahia_Ingenieria, el cual debe estar construido y una investigación la cual no hayamos investigado. Comprobaremos de manera similar a los problemas anteriores que recursos necesitamos, que han sido definidos en el problema. El resultado será un nuevo hecho, “Investigado Investigación”.

Este último hecho será introducido en la acción predefinida anteriormente, “Reclutar”, la cual en el apartado donde se comprueba que se va a generar un segador, también comprobaremos que tenemos una investigación realizada y su tipo es “Impulsor_Segador”. No hace falta hacer referencia a la Bahía ya que es necesaria para crear una investigación. En el problema se ha añadido bahia_ingenieria y los materiales requeridos.



6. Ejercicio 6.

Vamos a manipular el código anterior para la utilización de cálculo numérico, para ello vamos a mover aquellos predicados a “functions” para poder así ser utilizados numéricamente, estos serán “Requiere, Necesita, InvestigacionRequiere, MaximoRecurso, HayRecurso”, donde MaximoRecurso es un nuevo “hecho” que almacena la cantidad máxima de recursos a guardar. En la definición del problema y modificando la cláusula que indicaba el material necesario, ahora mantendrá información sobre un entero que indica la cantidad necesaria.

Crearemos un nuevo edificio “Depósito” que será añadido en el problema 3 de ellos en la localizaciones (1,2),(2,2),(3,2) sin construir y definiremos como MaximoRecurso en 100 junto a HayRecurso Gas,Mineral en 0. Modificaremos las siguientes acciones.

- **Asignar:** Incrementa en 10 el material recogido, si supera el MaximoRecurso, entonces asigna el material en la cantidad correspondiente a MaximoRecurso, así podemos recolectar $\min(95 + 10, 100 = \text{MaximoRecurso}) = 100 = \text{MaximoRecurso}$. **Se ha modificado a recolectar en 20 ya que si no este problema no acababa.**
- **Desasignar:** Encuentra una unidad “Extrayendo” y elimina dicho hecho.
- **Construir, Reclutar e Investigar:** De manera similar a como estaba, el forall que comprobaba la existencia de recurso ahora va a comprobar si para todo TipoRecurso, hay suficiente recurso que sea requerido.

$$\forall \text{TipoRecurso} (\text{HayRecurso}(\text{TipoRecurso}) \geq \text{Requiere}(\text{TipoRecurso}))$$

Si esto es cierto, podemos decrementar cada elemento que tenemos en lo que se requiere.

$$\forall \text{TipoRecurso} (\text{HayRecurso}(\text{TipoRecurso}) = \text{HayRecurso}(\text{TipoRecurso}) - \text{Requiere}(\text{TipoRecurso}))$$

- **Construir:** Aunque hemos expuesto los cambios más generales arriba, este tiene una modificación especial, en la actuación del efecto, cuando (“when”) se trata de un edificio de tipo Depósito, se incrementará el MaximoRecurso en 100.

1. Vemos el mapa final, con las localizaciones y los elementos construidos y a construir (C).
2. D.P hace referencia a los depósitos.
3. B.I a la bahía de ingeniería.
4. E.X son edificios sobre los recursos gas que no han sido aún construidos.
5. Comenzamos con un único VCE.

M.N	M.N	M.N		
D.P	D.P	D.P		
		C.M	G.S	G.S
		B.I	E.X	E.X
B.R				