

Universidad Simón Bolívar
CI5437 - Inteligencia Artificial I
Kevin Mena 13-10869
Leonardo López 14-10576
Pedro Samuel Fagundez 15-10460

Proyecto 4 - Mundo Pokémon

Descripción

Para este proyecto decidimos escoger el tema de **Artificial Intelligence Planning**, además decidimos utilizar **STRIPS** como **solver** para los problemas y **PDDL** como el lenguaje para describir el dominio de nuestro universo y la representación de los problemas.

Nuestro universo consiste en un mundo **Pokémon**, en el cual el **NPC** podrá abrir cofres para **recolectar pokebolas**, **moverse** y **capturar** los pokemones legendarios. La idea de los problemas que van a ser definidos dentro de este universo, consisten en que un NPC llamado **Ash** deberá capturar 3 pokemones legendarios, para ello necesita tener al menos 1 pokebola para capturar 1 pokémon, y dicha pokebola quedará usada e inservible para capturar otro pokémon legendario.

Lo que buscamos es que la **IA** pueda ser capaz de encontrar el mejor camino que lo lleve a poder capturar 3 pokemones legendarios, para lo cual es necesario haber hallado las 3 masterball que están sueltas alrededor del mapa.

Implementación

Decidimos utilizar la librería de **STRIPS** que vive en npm para **nodejs**, para ello creamos un archivo **index.js** que al ser ejecutado (con el comando `npm start <domain_file> <problem_file>`) carga un dominio y un problema y lo trata de resolver

con el solver, en caso de que exista la solución, va a ser impresa en un archivo **Results.txt**, en caso de no existir la solución, se imprime un mensaje en consola.

El archivo **Results.txt** consiste en enumerar y describir paso por paso cada una de las operaciones necesarias para llegar al estado final del problema.

Luego definimos y construimos el **dominio**, el cual está dividido en **predicates**, **types** y **actions**. Entremos más en detalle sobre estos elementos:

- **Types:** Aquí se define el tipo de las entidades que van a existir dentro de nuestro universo, para nuestro universo pokémon existen los siguientes:
 - **player:** definido para el NPC.
 - **location:** representa las posiciones del juego.
 - **masterball:** representa el elemento necesario para capturar a los pokemones legendarios.
 - **legendarypokemon:** representa la entidad objetivo del juego, capturar los pokemones es el objetivo principal.
 - **chest:** esta entidad existe para darle diversidad al juego, una masterball se ubica dentro de un cofre.
- **Predicates:** Los predicados representan la declaración de estados para las entidades del universo, un predicado permite establecer posiciones, estados del mundo y de sus entidades.
 - **at ?p ?l:** define la posición de p en l.
 - **border ?l1 ?l2:** define que de l2 es un borde de l1 (y no lo contrario).
 - **wall ?l:** define el atributo wall a la ubicación l, que será utilizado como pared o casilla a la cual no se puede trasladar.
 - **open ?c:** define el atributo open para los cofres.
 - **empty ?c:** define el atributo empty para los cofres.
 - **in ?o ?c:** define que o se encuentra dentro de c.

A continuación se describen predicados que relacionan el número de masterballs y de pokemones legendarios que posee el jugador.

- **has-no-masterball ?p**
 - **has-one-masterball ?p**
 - **has-two-masterball ?p**
 - **has-three-masterball ?p**
 - **has-no-legendary-pokemon ?p**
 - **has-one-legendary-pokemon ?p**
 - **has-two-legendary-pokemon ?p**
 - **has-three-legendary-pokemon ?p**
- **Actions:** Las acciones definen las operaciones que pueden ser realizadas dentro del juego, a continuación se mencionan y describen brevemente:
 - **move:** permite el movimiento del jugador actualizando las posiciones.
 - **open:** permite aperturar un chest marcándolo como open.
 - **collect-first-masterball:** Permite tomar una masterball de un chest.
 - **collect-second-masterball:** Permite tomar la masterball número 2 en el inventario de un chest abierto.
 - **collect-third-masterball:** Permite tomar la masterball número 3 en el inventario de un chest abierto.
 - **capture-first-legendary-pokemon-having-one-masterball:** Permite capturar el primer pokémon legendario teniendo 1 masterball, actualiza el estado y ahora se posee 0 masterballs.
 - **capture-first-legendary-pokemon-having-two-masterball:** Permite capturar el primer pokémon legendario teniendo 2 masterballs, actualizado el estado y ahora solo se poseerá 1 masterball.
 - **capture-first-legendary-pokemon-having-three-masterball:** Permite capturar el primer pokémon legendario teniendo 2 masterballs y realiza la misma operación anterior.
 - **capture-second-legendary-pokemon-having-one-masterball:** Permite capturar el segundo pokémon legendario teniendo 1 masterball.
 - **capture-second-legendary-pokemon-having-two-masterball:** Permite capturar el segundo pokémon legendario teniendo 2 masterballs.

- **capture-third-legendary-pokemon-having-one-masterball:** Permite capturar el tercer pokémon legendario.

De forma siguiente, construimos **problemas** para el dominio, principalmente generamos un problema que consistiera en un mapa, con paredes, donde estuvieran esparcidos los cofres con las masterballs y los pokémones legendarios, los movimientos del jugador podrán ser efectuados en cualquier posición donde no exista una pared o un pokémon.

La estructura del problema consiste en lo siguiente:

- **Objects:** Aquí se define el nombre que tendrán las entidades del problema relacionado a los **types** del dominio, por ejemplo se define que **ash** será el player.
- **Init:** Aquí se define el estado inicial del mundo para arrancar a jugar, básicamente con **border** se definen las fronteras entre las posiciones del juego, con **at** la ubicación inicial de ash, de los pokémones y de los cofres, con **wall** las paredes (sitios por los cuales no se puede caminar) y con **in** se insertan las masterballs en el cofre.
- **goal:** Aquí se define la condición que se debe cumplir para satisfacer el problema.

Finalmente esta es la implementación de lo que representa nuestro mundo pokémon y con el cual puede ser construido cualquier mundo, la representación con la cual hemos iniciado nosotros fue un mundo 4x4 de 16 casillas, 3x3 era extremadamente sencillo y 5x5 tuvimos problemas para correrlo y con el tiempo reducido que tuvimos lo desestimamos.

Luego de lograr encontrar el mejor resultado para el problema que teníamos decidimos aumentar el problema y ver si era posible conseguir un resultado con un problema más complicado con mayor número de variables en juego. Para ello planteamos que no solamente hay que atrapar los legendarios sino que cada legendario tiene un Pokémon guardián al que Ash debe vencer para poder

atraparlos, y para poder derrotar a estos guardianes Ash necesita atrapar Pokémons que venzan por tipos. Los guardianes existentes, son Pokémons de agua, fuego y hierba por lo que Ash debe capturar a sus contrapartes.

Se agregaron los siguientes predicados:

(fire-pokemon ?p - pokemon)
(water-pokemon ?p - pokemon)
(grass-pokemon ?p - pokemon)
(is-guardian ?p - pokemon ?lp - legendarpokemon)
(has-fire-type ?p - player)
(has-water-type ?p - player)
(has-grass-type ?p - player)
(defeated-no-guardian ?p - player)
(defeated-one-guardian ?p - player)
(defeated-two-guardian ?p - player)
(defeated-three-guardian ?p - player)

Y se agregaron las siguientes acciones:

- catch-fire-pokemon: Para capturar un pokémon de fuego que no sea un guardián.
- catch-water-pokemon: Para capturar un pokémon de agua que no sea un guardián.
- catch-grass-pokemon: Para capturar un pokémon de hierba que no sea un guardián.
- defeat-fire-guardian-first: Permite derrotar al guardián de fuego de primero, cada guardián está relacionado a un solo legendario.
- defeat-fire-guardian-second: Permite derrotar al guardián de fuego luego de derrotar a otro legendario, cada guardián está relacionado a un solo legendario.
- defeat-fire-guardian-third: Permite derrotar al guardián de fuego después de derrotar a los otros dos guardianes, cada guardián está relacionado a un solo legendario.
- defeat-water-guardian-first: Permite derrotar al guardián de agua de primero, cada guardián está relacionado a un solo legendario.

- defeat-water-guardian-second: Permite derrotar al guardián de agua luego de derrotar a otro legendario, cada guardián está relacionado a un solo legendario.
- defeat-water-guardian-third: Permite derrotar al guardián de agua después de derrotar a los otros dos guardianes, cada guardián está relacionado a un solo legendario.
- defeat-grass-guardian-first: Permite derrotar al guardián de hierba de primero, cada guardián está relacionado a un solo legendario.
- defeat-grass-guardian-second: Permite derrotar al guardián de hierba luego de derrotar a otro legendario, cada guardián está relacionado a un solo legendario.
- defeat-grass-guardian-third: Permite derrotar al guardián de hierba después de derrotar a los otros dos guardianes, cada guardián está relacionado a un solo legendario.

Lamentablemente, esta última modificación no fue posible de resolver dado el tiempo de corrida de STRIPS que al punto en el que se detuvo la ejecución iban 1 hora y media de tiempo, profundidad 9 y más de 900 estados hijos utilizando A*.

Resultados

Primera corrida oficial con dfs

- Solution found in 38 steps!

```
1. move ash grassfield13 grassfield9
2. move ash grassfield9 grassfield8
3. move ash grassfield8 grassfield4
4. open ash item1 grassfield4
5. move ash grassfield4 grassfield8
6. move ash grassfield8 grassfield9
7. move ash grassfield9 grassfield13
8. move ash grassfield13 grassfield12
9. open ash item2 grassfield12
10. move ash grassfield12 grassfield8
11. move ash grassfield8 grassfield4
12. collect-first-masterball ash item1 grassfield4
13. move ash grassfield4 grassfield8
14. move ash grassfield8 grassfield9
15. move ash grassfield9 grassfield13
16. move ash grassfield13 grassfield12
17. collect-first-masterball ash item2 grassfield12
18. move ash grassfield12 grassfield8
19. move ash grassfield8 grassfield4
20. capture-first-legendary-pokemon-having-one-masterball ash grassfield4 grassfield0 suicune
21. move ash grassfield4 grassfield8
22. move ash grassfield8 grassfield9
23. move ash grassfield9 grassfield13
24. move ash grassfield13 grassfield14
25. move ash grassfield14 grassfield15
26. move ash grassfield15 grassfield11
27. move ash grassfield11 grassfield7
28. move ash grassfield7 grassfield3
29. capture-second-legendary-pokemon-having-one-masterball ash grassfield3 grassfield2 raikou
30. move ash grassfield3 grassfield7
31. move ash grassfield7 grassfield11
32. move ash grassfield11 grassfield15
33. open ash item3 grassfield15
34. collect-first-masterball ash item3 grassfield15
35. move ash grassfield15 grassfield11
36. move ash grassfield11 grassfield7
37. move ash grassfield7 grassfield3
38. capture-third-legendary-pokemon-having-one-masterball ash grassfield3 grassfield2 raikou
```

Acá podemos notar que todas las operaciones que realizó y los cambios de estado fueron correctos, lo que le permitió llegar al goal, solo que el camino no era óptimo además de que no abrió el cofre incluso cuando estaba encima de él, por lo cual fue necesario algunos ajustes al dominio.

Cabe destacar que la solución óptima de este mapa se realiza en **20 pasos**.

Corrida con dfs luego de los ajustes al dominio

Solution found in 37 steps!

```
1. move ash grassfield13 grassfield9
2. move ash grassfield9 grassfield8
3. move ash grassfield8 grassfield4
4. open ash item1 grassfield4
5. move ash grassfield4 grassfield8
```

6. move ash grassfield8 grassfield9
7. move ash grassfield9 grassfield13
8. move ash grassfield13 grassfield12
9. open ash item2 grassfield12
10. move ash grassfield12 grassfield8
11. move ash grassfield8 grassfield4
12. collect-first-masterball ash item1 grassfield4 masterball1
13. move ash grassfield4 grassfield8
14. move ash grassfield8 grassfield9
15. move ash grassfield9 grassfield13
16. move ash grassfield13 grassfield12
17. collect-second-masterball ash item2 grassfield12 masterball2
18. move ash grassfield12 grassfield8
19. move ash grassfield8 grassfield4
20. capture-first-legendary-pokemon-having-two-masterball ash grassfield4 grassfield0 suicune
21. move ash grassfield4 grassfield8
22. move ash grassfield8 grassfield9
23. move ash grassfield9 grassfield13
24. move ash grassfield13 grassfield14
25. move ash grassfield14 grassfield15
26. move ash grassfield15 grassfield11
27. move ash grassfield11 grassfield7
28. move ash grassfield7 grassfield3
29. capture-second-legendary-pokemon-having-one-masterball ash grassfield3 grassfield2 raikou
30. move ash grassfield3 grassfield7
31. move ash grassfield7 grassfield11
32. move ash grassfield11 grassfield15
33. open ash item3 grassfield15
34. collect-first-masterball ash item3 grassfield15 masterball3
35. move ash grassfield15 grassfield14
36. move ash grassfield14 grassfield13
37. capture-third-legendary-pokemon-having-one-masterball ash grassfield13 grassfield14 entei

Vemos que ahora la solución cobra un poco más de sentido porque puede recolectar las masterballs justo después de aperturar el cofre, pero aún sabemos que el camino óptimo se logra en 20 pasos, por lo cual decidimos cambiar el algoritmo de resolución del solver por bfs y por A*.

Solución utilizando bfs y A* con una función de costos probablemente deficiente

- Solution found in 18 steps!

1. move ash grassfield13 grassfield12
2. open ash item2 grassfield12
3. collect-first-masterball ash item2 grassfield12 masterball2
4. move ash grassfield12 grassfield8
5. move ash grassfield8 grassfield4
6. open ash item1 grassfield4
7. collect-second-masterball ash item1 grassfield4 masterball1
8. capture-first-legendary-pokemon-having-two-masterball ash grassfield4 grassfield0 suicune
9. move ash grassfield4 grassfield8
10. move ash grassfield8 grassfield9
11. move ash grassfield9 grassfield13
12. capture-second-legendary-pokemon-having-two-masterball ash grassfield13 grassfield14 entei
13. move ash grassfield13 grassfield14
14. move ash grassfield14 grassfield15
15. move ash grassfield15 grassfield11
16. move ash grassfield11 grassfield7
17. move ash grassfield7 grassfield3
18. capture-third-legendary-pokemon-having-one-masterball ash grassfield3 grassfield2 raikou

Notemos que esta solución se aproxima mucho a la solución óptima, solo que es 2 pasos menor, ocurre que existe un bug en el paso 8 donde el estado de tener 2

masterballs no es reducido a 1, entonces en el paso 12 se tienen aún 2 masterballs y luego se procede a capturar el tercer pokémon legendario porque aún se posee una masterball después de capturar el segundo, los 2 pasos que faltan son los de abrir y recoger la masterball en el grassfield15, por lo cual esto nos llevó a proceso de debugging bastante interesante del cual aún no hemos podido salir.

Solución final con bfs y A* con función de costos probablemente no óptima

```
- Solution found in 20 steps!  
1. move ash grassfield13 grassfield9  
2. move ash grassfield9 grassfield8  
3. move ash grassfield8 grassfield4  
4. open ash item1 grassfield4  
5. collect-first-masterball ash item1 grassfield4 masterball1  
6. capture-first-legendary-pokemon-having-one-masterball ash grassfield4 grassfield0 suicune  
7. move ash grassfield4 grassfield8  
8. move ash grassfield8 grassfield12  
9. open ash item2 grassfield12  
10. collect-first-masterball ash item2 grassfield12 masterball2  
11. move ash grassfield12 grassfield13  
12. capture-second-legendary-pokemon-having-one-masterball ash grassfield13 grassfield14 ente  
13. move ash grassfield13 grassfield14  
14. move ash grassfield14 grassfield15  
15. open ash item3 grassfield15  
16. collect-first-masterball ash item3 grassfield15 masterball3  
17. move ash grassfield15 grassfield11  
18. move ash grassfield11 grassfield7  
19. move ash grassfield7 grassfield3  
20. capture-third-legendary-pokemon-having-one-masterball ash grassfield3 grassfield2 raikou
```

Luego de un proceso de debuggeo y de entender cómo se manejan los efectos en STRIPS logramos ajustar el dominio para poder conseguir la solución óptima del problema. Seguimos intentando conseguir una heurística que baje los tiempos de corrida del solver.

Conclusiones

STRIPS es un buen generador de planes para problemas que se puedan representar en estados, con condiciones para cada uno y llegar a una meta. La mejor forma de calcular el plan es utilizando el algoritmo de A*, colocando de buena forma las precondiciones y los efectos de cada acción así como tener una buena heurística dependiendo del problema a resolver. Esta es la parte más difícil del problema dado que conseguir una heurística consistente y admisible es una tarea de gran dificultad. Aún así con una heurística relativamente buena se pudieron conseguir buenos resultados en un tiempo relativamente corto en comparación con BFS y DFS.