Artificial Intelligence - Work in Energy Planning

The assignment asks to create a plan using the PDDL language that solves 3 problems that have the structure of a game. In this game the player enters a network of caves and his goal is to return having collected all the treasures he finds in the caves. At the same time, he must avoid dangers, such as monsters and pits, by using tools he meets along the way.

The design of the domain was done in the following editor:

http://editor.planning.domains/

Field Analysis

For the implementation of the Cave domain, the following assumptions have been made since they are not clear from the speech: a) There is no limit to the number of objects - treasures, weapons, shoes - that the player can carry and b) there cannot be a monster and a pit simultaneously in the same cave.

The**entities**of the field are the following:

1. location (represents the different locations, i.e. the caves)
2. person (represents the person entering the cave - the player)
3. treasure (the treasures found in the caves)
4. monster (the monsters found in the caves)
5. pit (the pits that the player can encounter in the caves)
6. shoes (the magical shoes that the player can find, which he can use to pass the pits)
7. weapon (the weapons found in the caves, which are used by the player to kill the monsters he will encounter)

These entities are linked to the following**relationships**:

1. at-person ?person1 ?location1 (declares that person1 is in cave location1)
2. at-treasure ?treasure1 ?location1 (declares that treasure1 is in cave location1)
3. at-shoes ?shoes1 ?location1 (declares that shoes1 is in cave location1)
4. at-weapon ?weapon1 ?location1 (declares that weapon weapon1 is in cave location1)
5. at-pit ?pit1 ?location1 (declares that pit1 is in cave location1)

6. at-monster ?monster1 ?location1 (declares that the monster monster1 is in the cave location1)
7. hasPath ?location1 ?location2 (declares that there is a path leading from cave location 1 to cave location2)
8. IsSafe ?location1 (declares location1 cave is safe, ie no monster or pit)
9. carries ?person1 ?treasure1 (declares that player person1 carries treasure1)
10. holds ?person1 ?weapon1 (declares player person1 holds weapon1)
11. wears ?person1 ?shoes1 (declares player person1 wears shoes1)

The**operators**which were needed for the implementation of the domain are the following:

1. Doer**Move**(?P ?L1 ?L2)
   - Accepts parameters (?P ?L1 ?L2) and moves player P from cave L1 to L2.
   - The operatorMove assumes that the player is in L1, that there is a path from L1 to L2, and that the L2 cave is safe, i.e. that the relationships (at-person ?P ?L1) hold (hasPath ?L1 ?L2) (isSafe ?L2).
   - The execution of the operatorMove results in the player now being in cave L2 rather than L1, so the new state events add (at-person ?P ?L2) and subtract (at-person ?P ?L1).

2. Operator**PickTreasure**(?P?L?T)
   - Accepts parameters (?P ?L ?T) and player P collects treasure T located in cave L.
   - The operatorPickTreasure assumes that the player is in L and that the treasure T is in L, that is, the relations (at-person ?P ?L) (at-treasure ?T ?L) hold.
   - The execution of the operatorPickTreasure results in the player carrying the treasure and the treasure ceases to be in cave L, so (carries ?P ?L) is added to the new state events and (at- treasure ?T ?L) is subtracted.

3. Performer**PickWeapon**(?P?L?W)
   - Accepts parameters (?P ?L ?W) and player P collects the weapon W found in cave L.
   - The operatorPickWeapon assumes that the player is in L and that the weapon W is in L, i.e. the relations (at-person ?P ?L) (at-weapon ?W ?L) hold.
   - The execution of the operatorPickWeapon causes the player to hold the weapon and the weapon ceases to be in cave L, so (holds ?P ?W) is added to the new state events and (at- weapon ?W ?L) is removed.

4. Executor**PickShoes**(?P?L?S)
   - Accepts parameters (?P ?L ?S) and player P collects shoes S located in cave L.
   - The operatorPickShoes assumes that the player is in L and that the shoes are in L, that is, the relations (at-person ?P ?L) (at-shoes ?S ?L) hold.
   - The execution of the operatorPickShoes results in the player wearing the shoes and they stop being in cave L, so (wears ?P ?S) is added to the new state events and (at-shoes ?S ?L) is subtracted.

5. Executor**UseWeapon**(?P ?L1 ?L2 ?M ?W)

- Accepts parameters (?P ?L1 ?L2 ?M ?W) and player P, while in cave L1, uses weapon W to kill monster M located in cave L2.

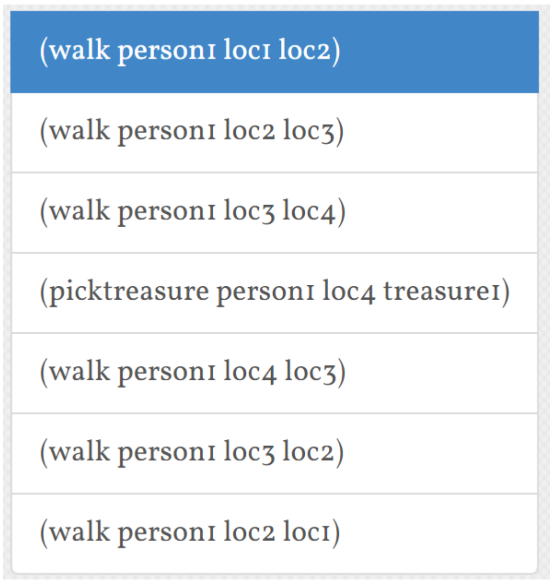
- The operatorUseWeapon assumes the player is at L1, holding the W weapon, there is a path from the L1 cave to L2, and there is a monster at L2. That is, the relations (at-person ?P ?L1) (holds ?P ?W) (hasPath ?L1 ?L2) and (at-monster ?M ?L2) must hold.
- The execution of the operatorUseWeapon results in the L2 cave now being safe, the player losing the weapon since it is disposable, and the monster not being in the L2 cave. Therefore, (isSafe ?L2) is added to the new state events and (holds ?P ?W) and (at-monster ?M ?L2) are removed.

6. Executor**Fly**(?P ?L1 ?L2 ?PIT ?S)
   - Accepts parameters (?P ?L1 ?L2 ?PIT ?S) and player P, located in cave L1, jumps over the PIT pit of L2 using magic shoes S.

   - The operatorFly assumes that player P is in cave L1, that there is a path from L1 to L2, that there is a PIT pit in L2, and that he is wearing shoes S. So, the relations (at-person ?P ?L1) must hold (hasPath ?L1 ?L2) (at-pit ?PIT ?L2) and (wears ?P ? S).
   - The execution of the operatorFly results in player P moving from L1 to L2, but losing his shoes. Therefore, (at-person ?P ? L2) is added to the new state events and (at-person ?P ?L1) and (wears ?P ?S) are subtracted.

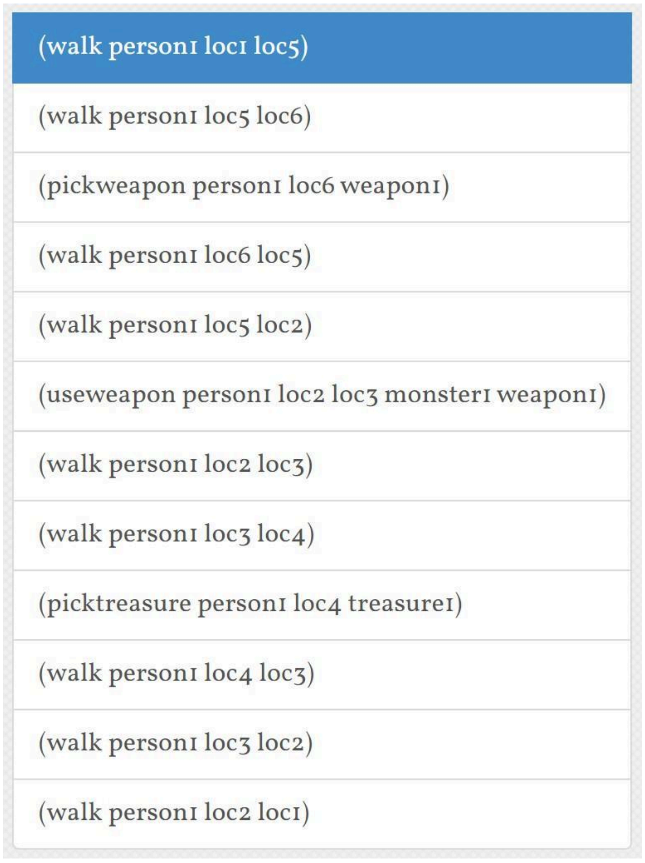
Solutions - Shots of the problems

1. For the Monster1 problem, the following 7-step solution was found. During the search, a total of 14 nodes were created, 9 of which were expanded. The execution time was instantaneous (the tool shows a negative value in the time so I think the reading is wrong).



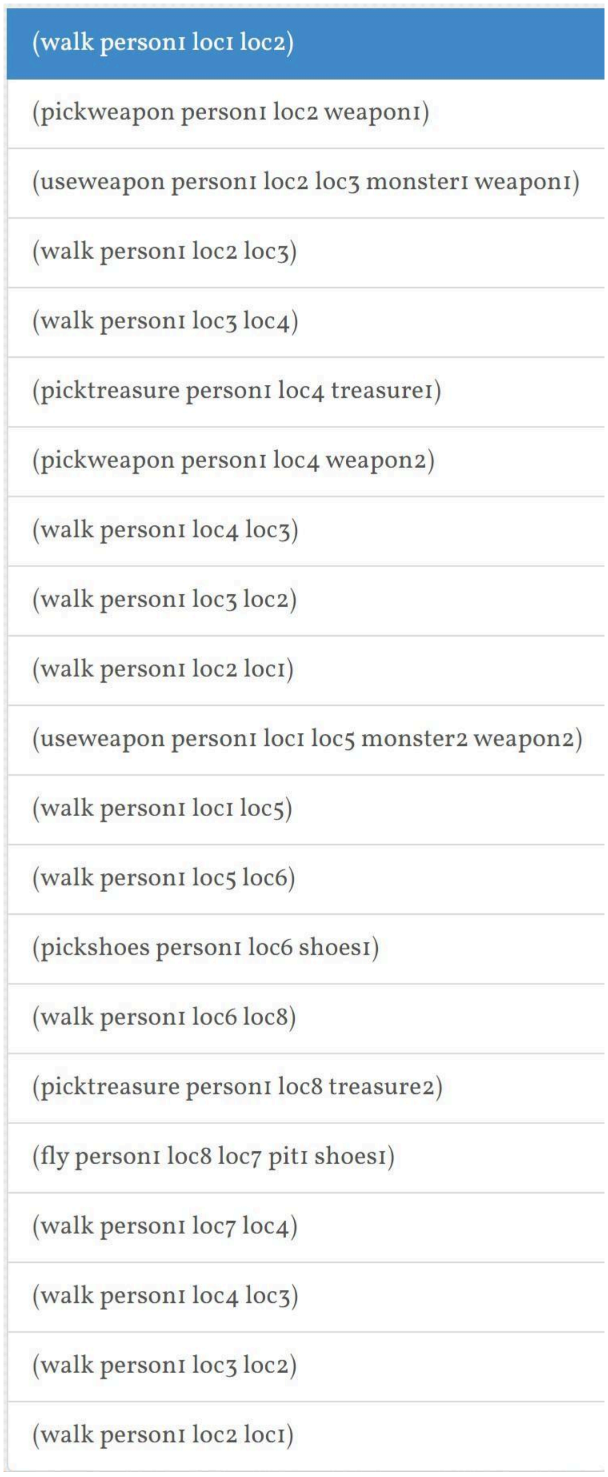
```
(walk person1 loc1 loc2)
(walk person1 loc2 loc3)
(walk person1 loc3 loc4)
(picktreasure person1 loc4 treasure1)
(walk person1 loc4 loc3)
(walk person1 loc3 loc2)
(walk person1 loc2 loc1)
```

2. For the Monster2 problem, the following 12-step solution was found. During the search, a total of 44 nodes were created, 21 of which were expanded. The execution time was instantaneous in this case as well.



```
(walk person1 loc1 loc5)
(walk person1 loc5 loc6)
(pickweapon person1 loc6 weapon1)
(walk person1 loc6 loc5)
(walk person1 loc5 loc2)
(useweapon person1 loc2 loc3 monster1 weapon1)
(walk person1 loc2 loc3)
(walk person1 loc3 loc4)
(picktreasure person1 loc4 treasure1)
(walk person1 loc4 loc3)
(walk person1 loc3 loc2)
(walk person1 loc2 loc1)
```

3. Finally, for the Monster3 problem a solution was found with 21 steps. A total of 55 nodes were generated during the search, 46 of which were expanded with instantaneous execution time.



```
(walk person1 loc1 loc2)
(pickweapon person1 loc2 weapon1)
(useweapon person1 loc2 loc3 monster1 weapon1)
(walk person1 loc2 loc3)
(walk person1 loc3 loc4)
(picktreasure person1 loc4 treasure1)
(pickweapon person1 loc4 weapon2)
(walk person1 loc4 loc3)
(walk person1 loc3 loc2)
(walk person1 loc2 loc1)
(useweapon person1 loc1 loc5 monster2 weapon2)
(walk person1 loc1 loc5)
(walk person1 loc5 loc6)
(pickshoes person1 loc6 shoes1)
(walk person1 loc6 loc8)
(picktreasure person1 loc8 treasure2)
(fly person1 loc8 loc7 pit1 shoes1)
(walk person1 loc7 loc4)
(walk person1 loc4 loc3)
(walk person1 loc3 loc2)
(walk person1 loc2 loc1)
```