Northwestern | McCORMICK SCHOOL OF ENGINEERING

COM_SCI 371 and MSAI 371: Knowledge Representation and Reasoning

---

Project Report

“**Murder Mystery Mansion (a.k.a. Clue)**”

March 14th, 2022

Professor:
Kenneth Forbus

Team Members:
Ammar Gilani (*Student ID- 3186176*)
Ayushi Mishra (*Student ID- 3292451*)
Azizakhon Mirsaidova (*Student ID-3386398)*
Shraddha Bangad (*Student ID- 3376340*)

Goal and motivation of the project

The main goal of the project was to build “Murder Mystery Mansion” game that was highly inspired by the board game Clue/Cluedo, a classical mystery game. The game was built and presented using P.D.D.L. (Planning Domain Definition Language) scripts.

The primary objective of the game was to identify who murdered the game's victim, what was the murderer’s motivation for killing, where the crime took place, and which weapon was used in the crime scene. Similar to the board game, the game we built has various characters, tools (weapons), and rooms within the mansion.

● ***The initial starting point*** of the game is the crime scene where the heiress of the mansion has been murdered.

● ***The goal state*** of the game is for the detective to find out who the actual murderer is and what was his/her motivation behind commiting the murder.

● ***The detective*** is the game player (the querier) trying to solve the murder case by visiting the crime scene, identifying clues, and interrogating the characters of the game.

There will be some known facts and events that are true related to the murder of the heiress in the game, and the rest will be possibilities. To deduce the details of the murder we have,

Project details

Planning tasks specified in PDDL are separated into two file categories:

- **A domain file for defining the predicates and actions.** Here the knowledge about all potential objects, entities, logical actions and reasoning flows are defined and encoded. Some examples of predicate definitions are- what type of Alibis are possible, what are the possible goal states or moods or interpersonal relationships,etc.

```
;;Types of Alibis
(interrogatedPerson ?p - person)
(personInDifferentRoomAlibi ?p - person ?r - room)
(personWithDifferentPersonAlibi ?p - person ?corroborator - person)
(noLocationAlibi ?p - person)
(noCorroborationAlibi ?p - person)

;;Moods
(happy ?p - person)
(sad ?p - person)
(calm ?p - person)
(worried ?p - person)
(angry ?p - person)
(scared ?p - person) ;;this is also required as scared shows that this person could have done something which is why he is scared

;;Interpersonal Relationships
(hates ?hater - person ?hated - person)
(loves ?lover - person ?loved - person)
(envies ?envier - person ?envied - person)
(loyal ?loyal - person ?loyalized - person)
```

  Some examples of action definitions are- defining potential means and motive of murder, discovering key owner, detective moving between rooms, love triangles, logical reasoning to identify murderer, etc.

```
:action MoveBetweenRooms
  :parameters (?currentroom - room ?nextroom - room)
  :precondition (and (detectiveCurrentlyInRoom ?currentroom)
                     (roomConnected ?currentroom ?nextroom))
  :effect (and (not (detectiveCurrentlyInRoom ?currentroom))
               (detectiveCurrentlyInRoom ?nextroom))
)


; Access to the murder weapon gives means
; Specifies murder weapon
; Mango has the poison thus making mango a murder weapon
:action PoisonedMangoIsMurderWeapon
   :parameters (?w - weapon)
   :precondition (and (poisoned ?w) (knownWeapon ?w))
   :effect (weaponIsMurderWeapon ?w)
)
```

- **A problem file for objects, initial state and goal specification.** This is our test file that has certain preconditions specified that are queried,logically reasoned

and verified by the knowledge encoded in the domain file. Each test file implies a storyline with details that could potentially frame the interrogated suspect for the murder. We have six test problem pddl files for six suspects of murder where we feed the following information for the detective regarding various murder suspects:

- Relationship between characters
- Current mood state of characters
- Current location of the character
- Alibis of the character
- Mansion room access that the characters had
- Objects in the room

```
(:init
    ;;detective's location
    (detectiveCurrentlyInRoom diningroom)

    ;;person interrogated
    (interrogatedPerson cook) ;; explicitly mentioning here since there are multiple people

    ;;relationships between characters
    (loves heiress duke)
    (loves cook duke)

    ;;current state of characters
    (dead heiress)
    (worried cook)
    (sad duke)

    ;;location of characters
    (personCurrentlyInRoom heiress diningroom)
    (personCurrentlyInRoom duke diningroom)
    (personCurrentlyInRoom cook kitchen)
    (personInRoomDuringMurder duke diningroom)

    ;;character alibis
    (personInDifferentRoomAlibi cook kitchen)
    (noLocationAlibi duke)

    ;;room access that characters have
    (ownsKey cook kitchenkey)
    (keyUnlocksRoom kitchenkey kitchen)
```

The murderer of the game is identified by our reasoner (coded in domain.pddl file) based on the goal state reached for each interrogated character (coded in test_problem.pddle files), which is categorized in 4 ways:

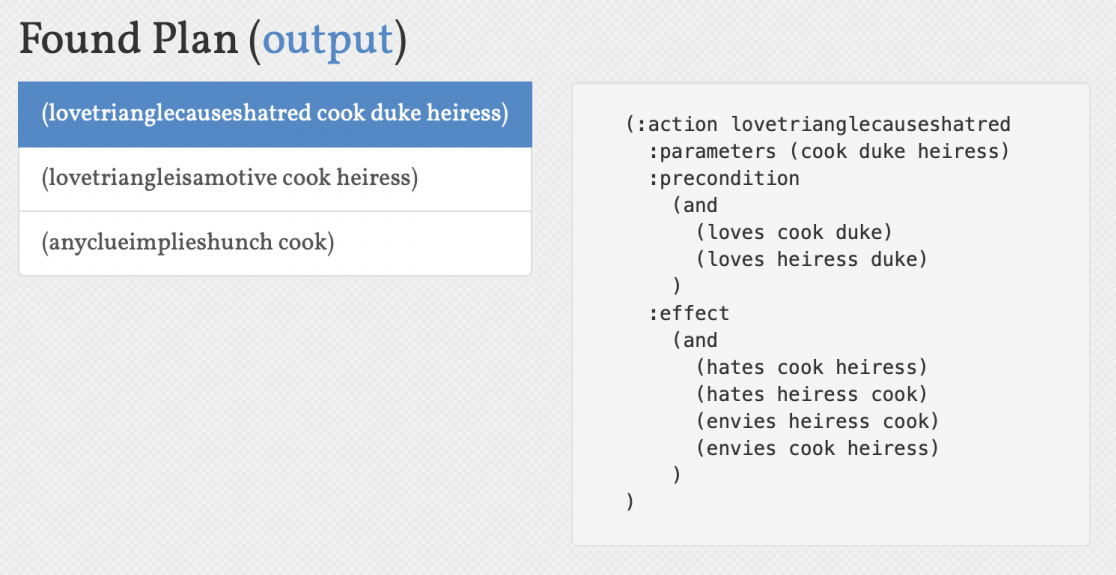
- **Hunch:** When the suspect checks out any one aspect of murder (mean, motive, suspicion).
- **Probable:** When the suspect checks out atleast two aspects of murder (i.e. means and motive OR motive and suspicion OR means and suspicion).
- **Solved:** When the suspect checks out all three aspects of murder (mean, motive and suspicion).

- **Illogical:** When suspect checks out neither of the three aspects of murder (mean, motive and suspicion) OR there is no logical or rational reasoning to the detective's hypothesis.

Reaching the goal state would involve logical reasoning-driven identification of which facts are true and which aren't, along with verification of whose alibi checks out to be true and whose doesn't at the time of the murder. Please note that in order to automate the process and make it less time consuming for the game player, the detective, too, will have some set moves, actions, and interrogation rules defined for verifying the alibis.

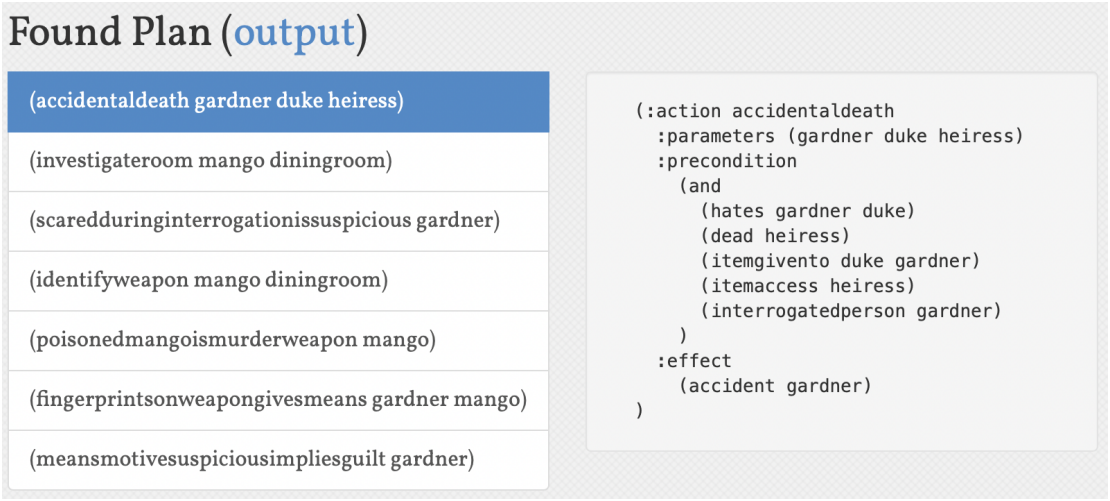
Following are the sequential logical deductions made by the reasoner (left side of the output image) based on the inputs of the test case (right side of the output image) for some of the suspects of our game:

- **Test case1- Love triangle between cook, duke and heiress leads to the heiress's death:** In this scenario the cook is assumed to be the murderer by poisoning the heiress since she is having an affair with the duke. Love triangle causes hatred which does give cook a motive but there are no means or suspicion to further support this theory. Hence the goal state reached is a hunch (anyclueimplieshunch) thereby indicating that our reasoner disproves the cook as a suspect.

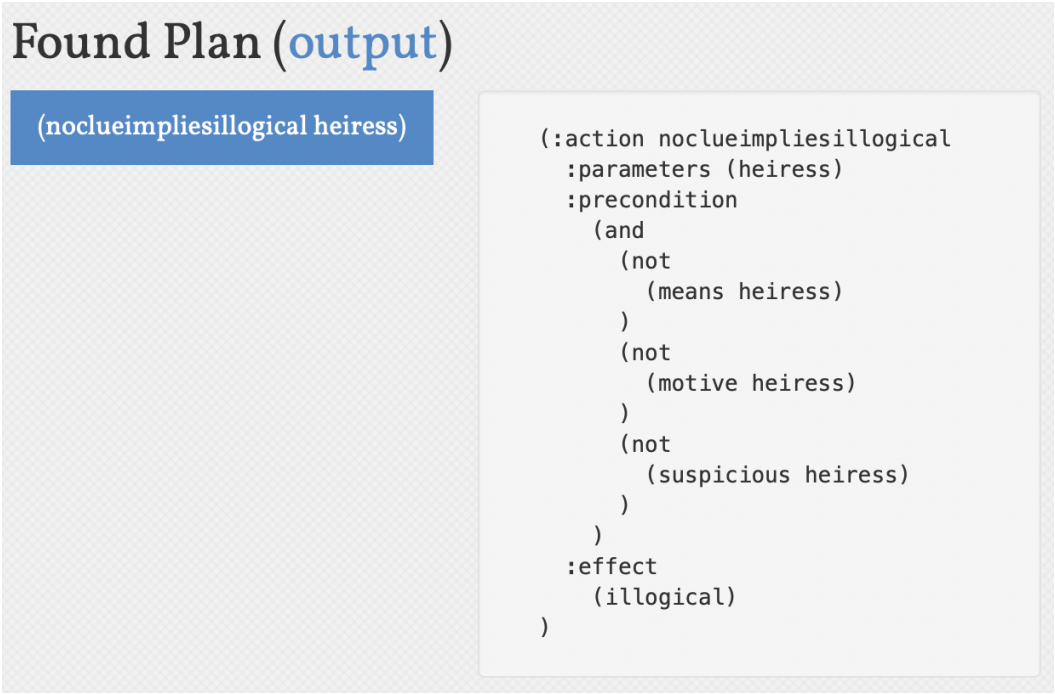


- **Test case2- Accidental death of heiress by gardener:** In this scenario the gardener is assumed to be the killer since he sent mangoes for the duke which were poisoned and eaten by the heiress. Thereby accidentally killing the heiress. Our reasoner reaches the goal state of this test case as solvable. It recognizes

the gardner as the murderer since the gardner checks out the means (fingerprints on the mango were his and the mango was poisoned), motive (loved his wife the cook and wanted to take the duke out of the picture since he was having an affair with his wife) and suspicion (scared during interrogation).



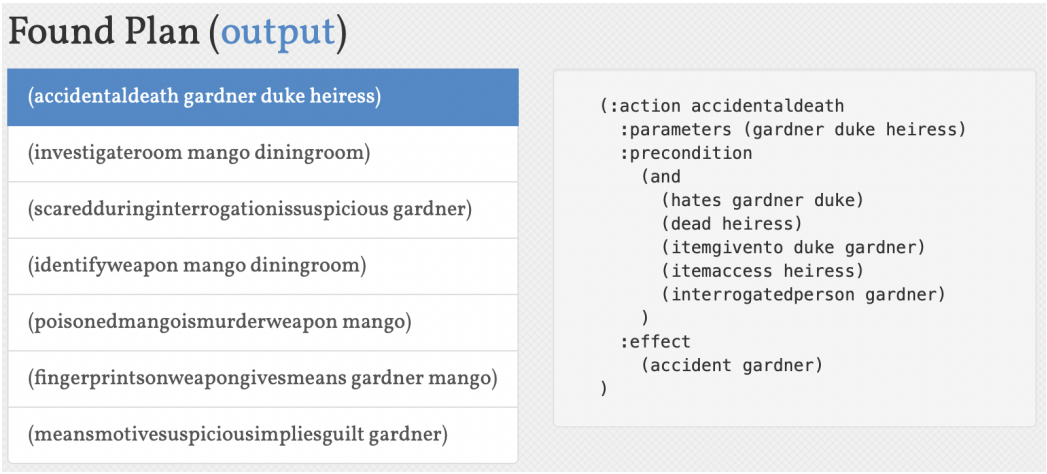
- **Test case 6- The maid is the suspect:** In this scenario the maid is considered to be a suspect. Since the maid was in the ballroom at the time of murder and was loyal to the heiress the reasoner couldn't find any means, motive or suspicion attached to the maid that would result in her committing the murder. Thereby declaring this hypothesis as illogical resulting in the maid being innocent.



Strengths and weaknesses of our approach

Strengths:

- As we use PDDL script for implementation,we inherit its computation properties thereby making derivation of inferences very fast.
- Since we used PDDL, we were able to observe how the inferences were made (stepwise logical deduction as can be seen in the left side of the output screenshot below).



- The mystery game could be adapted to different crime scenes by simply changing the initial starting point. Thereby making additions of test cases, characters and possibilities easier. For eg- We could create a scenario where the heiress is killed by a new character Miss Beams (the neighbour) through candlestick by defining the character along with the murder object in the domain file and providing her whereabouts in the test_problem file.

Weaknesses: As we used PDDL for implementation, our approach inherits disadvantages of PDDL-

- We would not be able to use functions which makes it impossible to express certain facts.
- We would not be able to use universal, existential quantifiers which makes it impossible to express rules.
- There is no room for representing that some things are unknown. With the closed world assumption, whatever is not mentioned, is assumed to be false.

What you might do differently next time, or what next steps would be interesting to do.

If we were to do this project again, we would like to further expand the murder mystery mansion world by introducing multiple complex stories. We chose a relatively simple idea with only a few complex story arcs. We believe this was a wise decision based on the time constraint we had. But we would still enjoy the challenge of working on more characters, weapons and scenarios. Following are certain ways in which this could be done:

- To make the story more complex, **we can add new characters** to increase the number of suspects. For example, add some guests for perhaps the duke's birthday, introduce family members like step-mother, step-sisters, mother in-laws, certain unknown characters like heiress's secret admirer who is dangerously obsessed with her, etc. This would have increased the number of suspects and made the story more interesting.
- We could **introduce more potential weapons** for muder. For example, there's a head wound to the heiress's head- was it a knife, candlestick, garden shovel, etc. that was used to kill the heiress. Adding multiple weapons would make the game stakes high and increase accessibility of these weapons among characters (for example, as almost everyone can access a knief).
- We could **add more test cases** scenarios. For example, Greed_of_Money: Duke might have killed the heiress for money; Anger_DueTo_Disrepect: The secret admirer could have been a potential murder as the heiress had disrespected him; Fear_Of_Truth_ComingOut: heiress must have known a dark secret of the family for which she must have been killed, etc.
- Apart from making changes in storyline, we would also like to try implementing the same project in **different knowledge representation** languages.