Advanced topics in AI2: Reverse Planning

Ounadi Ikram

February 28, 2024

Introduction

This project focuses on identifying reverse plans within a STRIPS domain. It involves the implementation of a STRIPS domain in PDDL (Planning Domain Definition Language) and a simplified version in the K language. The project utilizes tools and encodings introduced in lectures to discover reverse plans. Initially, the domain is outlined. Then, the implementations in both PDDL and K are detailed in section 2. Section 3 includes screenshots from the testing phase, demonstrating the functionality of the implementations. Finally, section 4 presents the set up I used.

1 Domain

This domain is modeled after the popular television show "Stranger Things," which shows a made-up universe in which Eleven lives in the small Indiana town of Hawkins with her close friends Mike Wheeler, Dustin Henderson, Lucas Sinclair, and Will Byers and others (ElevenSquad) . Their task is to make their way across a world full of bizarre creatures and frightening events.

One of the main things they have to do is send scary monsters to the mysterious dimension called the Upside Down. They also try to summon monsters from the Other Side so they may use them to fight other animals in Hawkins. These animals may be anything from little demodogs to powerful demogorgons, thus two people are needed to manage them safely.

Sometimes Eleven and a friend must seek out another pal to combine their powers to either bring the monster back to the Upside Down or use it as bait to catch additional monsters. During their dangerous travels, Eleven and her companions frequently find comfort and relief in The Byers House, a warm retreat owned by Will Byers and his family that doubles as a tactical headquarters for their operations.

2 Implementation

2.1 Implementation in PDDL

Action 1: This action means seeking refuge at the Byers' residence, without any predetermined conditions. It provides a safe haven for Eleven or one of her friends during moments of adversity or uncertainty.

Go to Byers

```
(:action go_to_byers
    :effect (and
        (elevensquad_relaxed)
    )
)
```

Action 2: luring a monster from the Upside Down into Hawkins. While it may seem peculiar at first glance, the rationale behind this approach lies in the complexity of combating these otherworldly creatures. By enticing one monster to confront another, it presents a strategic opportunity to manipulate their behavior and potentially neutralize a greater threat.

Bring a monster to Hawkins

```
(:action bring_monster
    :effect (and
        (monster_at_hawkings)
    )
)
```

Action 3: this action is for the squad member decides to return to Hawkins, understanding the parallel nature of the Upside Down and the real world. Before embarking on this journey, they find solace and preparation at the Byers' house, knowing they need to be mentally and emotionally ready for the challenges ahead.

Go the monster location

```
    (:action go_to_hawkings
        :precondition (and
            (elevensquad_relaxed)
        )
        :effect (and
            (elevensquad_at_hawkings)
            (not(elevensquad_at_upside_down))
        )
    )
```

Action 4: This action involves the pursuit and hunt of a monster by a member of the elevensquad in Hawkins. The member must be in a relaxed and prepared state for battle, while both the member and the monster must be present in Hawkins for the action to occur.

Hunt the monster

```
     (:action hunt_monster
        :precondition (and
            (elevensquad_at_hawkings)
            (elevensquad_relaxed)
            (monster_at_hawkings)
        )
        :effect (and
            (elevensquad_takedown_monster)
        )
    )
```

Action 5: This action pertains to subduing the monster and transporting it to the Upside Down, with the requirement that the member of the elevensquad involved must be adequately prepared.

Go to the upside down

```
(:action go_to_upside_down
     :precondition (and
         (elevensquad_relaxed)
         (monster_at_hawkings)
         (elevensquad_takedown_monster)
     )
     :effect (and
         (not (monster_at_hawkings))
         (elevensquad_at_upside_down)
         (not (elevensquad_at_hawkings))
     )
 )
```

Action 6: this action involves an elevensquad member reaching the Upside Down and releasing the captured monster, an endeavor that drains all of the member's energy. The term "taking down" signifies the act of capturing the monster rather than defeating it. Once in the Upside Down, the squad member initiates the release of the monster, a task that demands significant exertion and depletes the member's energy entirely.

Unleash the monster

```
(:action unleash_monster
     :precondition (and
         (not(monster_at_hawkings))
         (elevensquad_relaxed)
         (elevensquad_takedown_monster)
         (elevensquad_at_upside_down)
         (not(elevensquad_at_hawkings))
     )
     :effect (and
         (not(elevensquad_takedown_monster))
         (not (elevensquad_relaxed))
     )
 )
```

Action 7: dodging a monster is always possible if the member is in a good shape.

Dodging monster

```
 (:action dodge_monster
     :precondition (and
         (elevensquad_relaxed)
     )
     :effect (and
         (not (monster_attacked))
     )
 )
```

Action 8: Instead of capturing the monster, a squad member has the option to engage in direct confrontation and attack, provided they are adequately prepared.

Attack monster

```
(:action attack_monster
    :precondition (and
        (not(monster_attacked))
        (elevensquad_relaxed)
    )
    :effect (and
        (monster_attacked)
        (not (elevensquad_relaxed))
    )
)
```

Action 9: some monsters are stronger than other like Demogorgons, and to take down one of them , thesquad members should cooperate ,specially when they want for this monster to track the other monsters in Hawkings to where they actually belong.

Attack monster

```
(:action bring_demogorgon
    :effect (and
        (demogorgon_at_hawkings)
        (not(elevensquad_synergy))
    )
)
```

Action 10: this actions if for the two members to collide.

combine the power of two members in the elevensquad

```
(:action call_another_elevensquad
    :precondition (and
        (elevensquad_relaxed)
        (demogorgon_at_hawkings)
        (not(elevensquad_synergy))
    )
    :effect (and
        (elevensquad_sos)
    )
)
```

Action 11: this actions is for when the cooperation happens.

perform the cooperation between 2 elevensquads

```
(:action elevensquad_harmony
    :precondition (and
        (elevensquad_relaxed)
        (demogorgon_at_hawkings)
        (elevensquad_sos)
    )
    :effect (and
        (elevensquad_synergy)
        (not(elevensquad_sos))
    )
)
```

Action 12: This action is aimed at returning the Demogorgon to its native realm.

Take demogorgon to upside down

```
(:action take_demogorgon_to_upside_down
    :precondition (and
        (elevensquad_relaxed)
        (elevensquad_synergy)
        (demogorgon_at_hawkings)
    )
    :effect (and
        (not(demogorgon_at_hawkings))
    )
)
```

Action 13: Parting ways is inevitably difficult, but it often enhances productivity. However, this process can dampen the atmosphere and leave both parties feeling drained afterward.

The goodbye between two elevensquad members

```
(:action elevensquad_dismantle
    :precondition (and
        (elevensquad_relaxed)
        (not(demogorgon_at_hawkings))
        (elevensquad_synergy)
        (not(elevensquad_sos))
    )
    :effect (and
        (not(elevensquad_synergy))
        (not(elevensquad_relaxed))
    )
)
```

2.2 Implementation with K

I reduced the domain because it was difficult to implement; currently it reads: Elevensquad member hunt a monster in Hawkings and unleash him in the upside down.

I did not run the code since I couldn't prepare all the dependencies .

PDDL Domain Specification

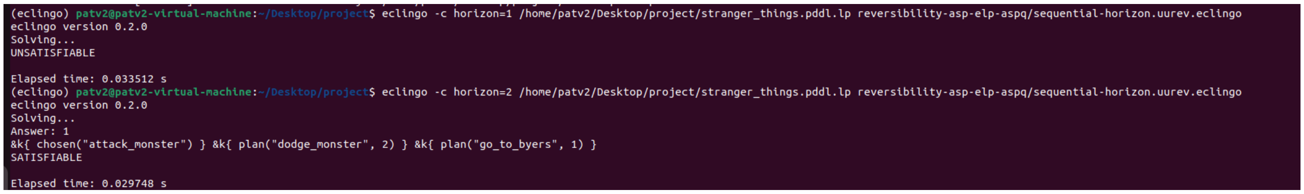
```
fluents:
        monster_at_hawkings.
        elevensquad_at_hawkings.
        elevensquad_at_upside_down.
        elevensquad_relaxed.
        elevensquad_takedown_monster.

actions:
        go_to_hawkings.
        hunt_monster.
        go_to_upside_down.
        unleash_monster.

always:
        executable go_to_hawkings if elevensquad_relaxed.
        caused elevensquad_at_hawkings after go_to_hawkings.
        caused -elevensquad_at_upside_down after go_to_upside_down.

        executable hunt_monster if elevensquad_at_hawkings.
        executable hunt_monster if elevensquad_relaxed.
        executable hunt_monster if monster_at_hawkings.
        caused elevensquad_takedown_monster after hunt_monster.

        executable go_to_upside_down if elevensquad_relaxed.
        executable go_to_upside_down if monster_at_hawkings.
        executable go_to_upside_down if elevensquad_takedown_monster.
        caused monster_at_hawkings after go_to_upside_down.
        caused elevensquad_at_upside_down after go_to_upside_down.
        caused elevensquad_at_hawkings after go_to_hawkings.

        executable unleash_monster if -monster_at_hawkings.
        executable unleash_monster if -elevensquad_at_hawkings.
        executable unleash_monster if elevensquad_at_upside_down.
        executable unleash_monster if elevensquad_relaxed.
        executable unleash_monster if elevensquad_takedown_monster .
        caused -elevensquad_takedown_monster after unleash_monster.
        caused -elevensquad_relaxed after unleash_monster.

initially:
        total monster_at_hawkings.
        elevensquad_at_hawkings.
        -elevensquad_at_upside_down.
        total elevensquad_relaxed.
        total elevensquad_takedown_monster.


goal: -elevensquad_takedown_monster? (4)
```

3 Results of the tests

During the testing phase, the range of horizons explored extended from 1 to 5.

Additionally, I attempted to use Clingo, but encountered issues when my virtual machine became unresponsive. Unfortunately, I neglected to capture any screenshots during this process.

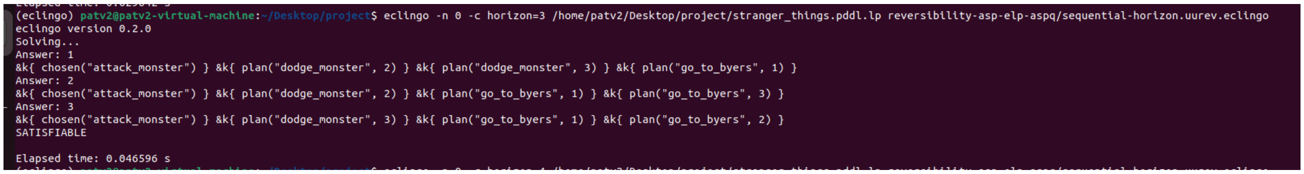
Horizon 1 and 2



```
(eclingo) patv2@patv2-virtual-machine:~/Desktop/project$ eclingo -c horizon=1 /home/patv2/Desktop/project/stranger_things.pddl.lp reversibility-asp-elp-aspq/sequential-horizon.uurev.eclingo
eclingo version 0.2.0
Solving...
UNSATISFIABLE

Elapsed time: 0.033512 s
(eclingo) patv2@patv2-virtual-machine:~/Desktop/project$ eclingo -c horizon=2 /home/patv2/Desktop/project/stranger_things.pddl.lp reversibility-asp-elp-aspq/sequential-horizon.uurev.eclingo
eclingo version 0.2.0
Solving...
Answer: 1
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("go_to_byers", 1) }
SATISFIABLE

Elapsed time: 0.029748 s
```

Figure 1: test using eclingo with horizon=1,2

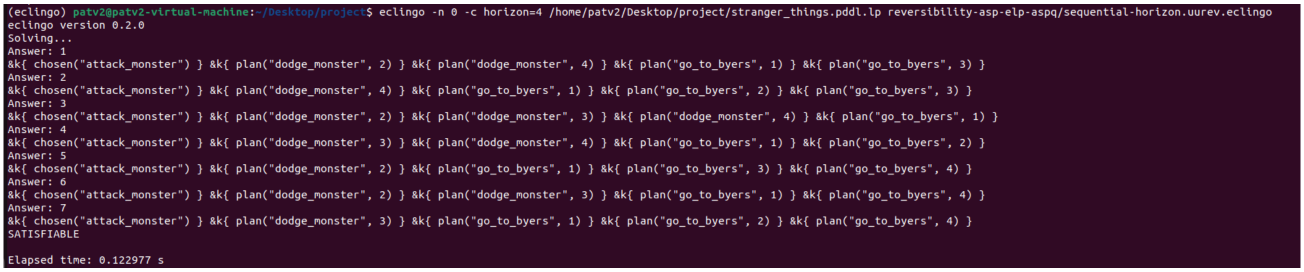
Horizon 3



```
(eclingo) patv2@patv2-virtual-machine:~/Desktop/project$ eclingo -n 0 -c horizon=3 /home/patv2/Desktop/project/stranger_things.pddl.lp reversibility-asp-elp-aspq/sequential-horizon.uurev.eclingo
eclingo version 0.2.0
Solving...
Answer: 1
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 3) } &k{ plan("go_to_byers", 1) }
Answer: 2
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 3) }
Answer: 3
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 3) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) }
SATISFIABLE

Elapsed time: 0.046596 s
```

Figure 2: test using eclingo with horizon=3

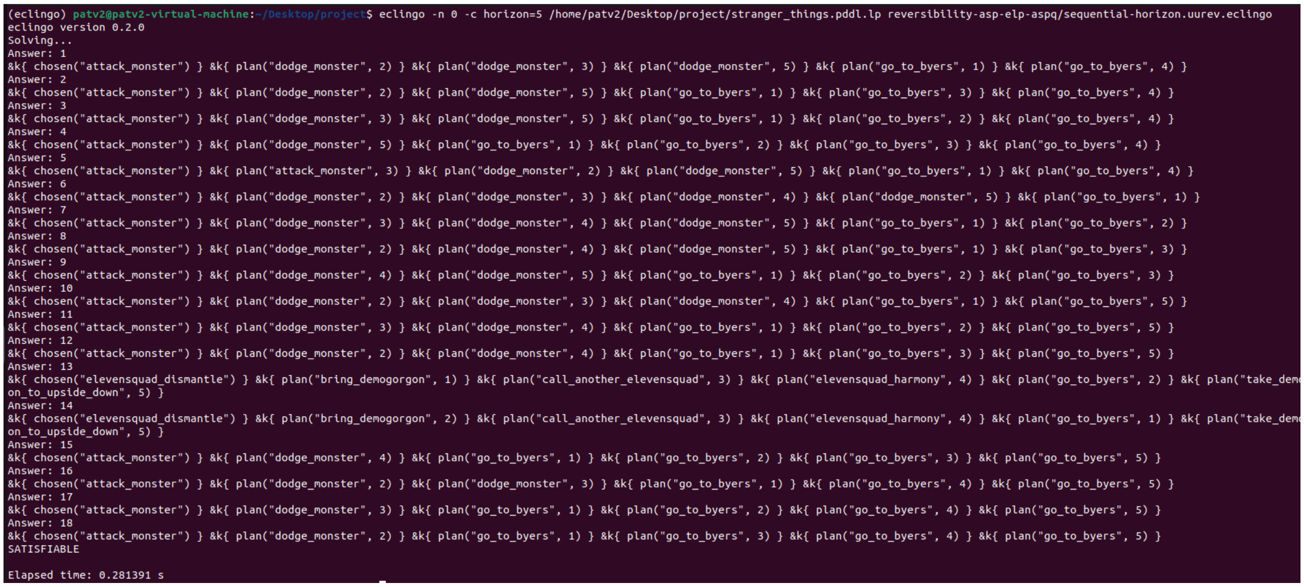
Horizon 4



```
(eclingo) patv2@patv2-virtual-machine:~/Desktop/project$ eclingo -n 0 -c horizon=4 /home/patv2/Desktop/project/stranger_things.pddl.lp reversibility-asp-elp-aspq/sequential-horizon.uurev.eclingo
eclingo version 0.2.0
Solving...
Answer: 1
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 3) }
Answer: 2
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 3) }
Answer: 3
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) }
Answer: 4
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) }
Answer: 5
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 3) } &k{ plan("go_to_byers", 4) }
Answer: 6
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 3) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 4) }
Answer: 7
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 3) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 4) }
SATISFIABLE

Elapsed time: 0.122977 s
```

Figure 3: test using eclingo with horizon=4

Horizon 5



```
(eclingo) patv2@patv2-virtual-machine:~/Desktop/project$ eclingo -n 0 -c horizon=5 /home/patv2/Desktop/project/stranger_things.pddl.lp reversibility-asp-elp-aspq/sequential-horizon.uurev.eclingo
eclingo version 0.2.0
Solving...
Answer: 1
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 4) }
Answer: 2
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 3) } &k{ plan("go_to_byers", 4) }
Answer: 3
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 4) }
Answer: 4
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 3) } &k{ plan("go_to_byers", 4) }
Answer: 5
&k{ chosen("attack_monster") } &k{ plan("attack_monster", 3) } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 4) }
Answer: 6
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 4) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) }
Answer: 7
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 4) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) }
Answer: 8
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 4) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 3) }
Answer: 9
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 4) } &k{ plan("dodge_monster", 5) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 3) }
Answer: 10
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 5) }
Answer: 11
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 3) } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 5) }
Answer: 12
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 3) } &k{ plan("go_to_byers", 5) }
Answer: 13
&k{ chosen("elevensquad_dismantle") } &k{ plan("bring_demogorgon", 1) } &k{ plan("call_another_elevensquad", 3) } &k{ plan("elevensquad_harmony", 4) } &k{ plan("go_to_byers", 2) } &k{ plan("take_dem
on_to_upside_down", 5) }
Answer: 14
&k{ chosen("elevensquad_dismantle") } &k{ plan("bring_demogorgon", 2) } &k{ plan("call_another_elevensquad", 3) } &k{ plan("elevensquad_harmony", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("take_dem
on_to_upside_down", 5) }
Answer: 15
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 4) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 3) } &k{ plan("go_to_byers", 5) }
Answer: 16
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("dodge_monster", 3) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 4) } &k{ plan("go_to_byers", 5) }
Answer: 17
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 3) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 2) } &k{ plan("go_to_byers", 4) } &k{ plan("go_to_byers", 5) }
Answer: 18
&k{ chosen("attack_monster") } &k{ plan("dodge_monster", 2) } &k{ plan("go_to_byers", 1) } &k{ plan("go_to_byers", 3) } &k{ plan("go_to_byers", 4) } &k{ plan("go_to_byers", 5) }
SATISFIABLE

Elapsed time: 0.281391 s
```

Figure 4: test using eclingo with horizon=5

4 Set up

Tools

- **Clingo**: Version 5.4.0
- **Eclingo**: Version 0.2.0
- **Plasp**: Version 3.1.1
- **Qasp**: Version 0.1
- **Python Version**: 3.10

System Specifications

- **CPU**: Intel® Core™ i7-10750H Processor
- **Memory (RAM)**: 8 GB
- **GPU**: NVIDIA GeForce RTX 2060 with 16 GB memory
- **Operating System**: Ubuntu 22.04