CIS 700 HW4: Convert WikiHow to PDDL Report

Weiqiu You, Yifei Li

March 3, 2022

1 What WikiHow article did you pick and why?

We select the WikiHow article *How to Survive on a Desert Island*[1] since:

- There are different locations so it's more natural and logical to be converted to planning format.
- The domain knowledge is not alien.

2 What portions of the article did you select to translate to PDDL?

There are three portions (w.r.t three problems):

- Portion (Problem) 1 - Hunt Animal for Food: "Find a fresh water source", step-3 method-1 of WikiHow
- Portion (Problem) 2 - Make a Fire: "Make a fire", step-4 method-2 of WikiHow
- Portion (Problem) 3 - Construct a Sturdy Shelter: "Construct a sturdy shelter.", step-3 method-2 of WikiHow

3 Give some examples of the actions, types, and predicates you used in your domain.

Examples:

- Actions (selected):

```
(:action catch_fish ; catch fish from a location that has fish.
 :parameters (?p - player ?loc - location ?fish - fish ?stick - stick)
 :precondition (and (at ?p ?loc) (has_fish ?loc) (inventory ?p ?stick)
                    (sharpened ?stick))
 :effect (and (inventory ?p ?fish) (edible ?fish))
)

(:action cook ; cook to make animal edible
 :parameters (?p - player ?loc - location ?animal - animal)
 :precondition (and (at ?p ?loc) (inventory ?p ?animal) (has_fire ?loc))
 :effect (and (inventory ?p ?animal) (edible ?animal))
)

(:action get_stick ; get a stick from bosk
 :parameters (?p - player ?loc - location ?stick - stick)
 :precondition (and (at ?p ?loc) (at ?stick ?loc))
 :effect (and (inventory ?p ?stick) (not (sharpened ?stick)))
)

(:action start_fire ; start a fire at a location
 :parameters (?p - player ?loc - location ?stick - stick)
 :precondition (and (at ?p ?loc) (inventory ?p ?stick) (sharpened ?stick))
 :effect (and (has_fire ?loc))
)
```

[1] https://www.wikihow.com/Survive-on-a-Desert-Island

- Types:

```
(:types
  water stick tarp leave - item
  player direction location
      fish bird insect shellfish - animal
)
```

- Predicates:

```
(:predicates
 (has_water_source ?loc - location) ; this location has fresh water.
 (treated ?water - water) ; True if the water decontaminated by boiling
 (at ?obj - object ?loc - location) ; an object is at a location
 (inventory ?player - player ?item - item) ; item in the inventory
 (connected ?loc1 - location ?dir - direction ?loc2 - location) ; direction
 (blocked ?loc1 - location ?dir - direction ?loc2 - location) ; block
 (edible ?item); an item is edible
 (has_fish ?loc - location) ; this location has fish.
 (has_bird ?loc - location) ; this location has bird.
 (has_insect ?loc - location) ; this location has insect.
 (has_shellfish ?loc - location) ; this location has shellfish.
 (has_fire ?loc - location) ; this location has fire.
 (sheltered ?loc - location) ; this location is sheltered.
)
```

4 Explain what goal you selected for your problem, and give the initial state and solution that you created.

- Problem 1: We set the goal that NPC should (pick a stick to) collect different food, fish, bird, insect and shellfish to diverse his/her food resources. Further, NPC should (pick another stick to start fire to) cook the food, making them edible and therefore avoid poison.

```
(:init
 (connected camp west path)
 (connected path east camp)
 (connected camp north beach)
 (connected beach south camp)
 (connected path west cliff)
 (connected cliff east path)
 (connected cliff up clifftop)
 (connected clifftop east bushwood)
 (connected bushwood east clifftop)
 (connected clifftop down cliff)
 (at npc camp)
 (at canteen camp)
 (at stick bushwood)
 (at stick2 camp)
 (has_fish beach)
 (has_bird clifftop)
 (has_insect bushwood)
 (has_insect path)
 (has_shellfish beach)
)

(:goal (and (inventory npc fish) (edible fish) (inventory npc bird)
            (edible bird) (inventory npc insect) (edible insect)
            (inventory npc shellfish) (edible shellfish))
)
```

with solution

```
Time: 0.7989721298217773s
plan:
get stick2 npc camp
go north npc camp beach
sharpen_stick npc stick2
catch_fish npc beach fish stick2
get_shellfish npc beach shellfish
go south npc beach camp
go west npc camp path
hunt_insect npc path insect
go west npc path cliff
go up npc cliff clifftop
hunt_bird npc clifftop bird stick2
start_fire npc clifftop stick2
cook npc clifftop shellfish
```

- Problem 2: We set the goal that NPC should (pick a stick to) start fire in the camp and therefore s/he can cook food and keep warm.

```
(:init
 (connected camp west path)
 (connected path east camp)
 (connected camp north beach)
 (connected beach south camp)
 (connected path west cliff)
 (connected cliff east path)
 (connected cliff up clifftop)
 (connected clifftop east bushwood)
 (connected bushwood east clifftop)
 (connected clifftop down cliff)
 (at npc camp)
 (at canteen camp)
 (at stick bushwood)
)

(:goal (and (has_fire camp)))
```

with solution

```
Time: 0.01341700553894043s
plan:
go west npc camp path
go west npc path cliff
go up npc cliff clifftop
go east npc clifftop bushwood
get stick npc bushwood
go east npc bushwood clifftop
go down npc clifftop cliff
go east npc cliff path
go east npc path camp
sharpen_stick npc stick
start_fire npc camp stick
```

- Problem 3: We set the goal that NPC should (use stick and tarp to) create shelter in the camp where s/he can sleep inside.

```
(:init
 (connected camp west path)
 (connected path east camp)
 (connected camp north beach)
 (connected beach south camp)
 (connected path west cliff)
```

```
(connected cliff east path)
(connected cliff up clifftop)
(connected clifftop east bushwood)
(connected bushwood east clifftop)
(connected clifftop down cliff)
(at npc camp)
(at canteen camp)
(at stick bushwood)
)

(:goal (and (sheltered camp)))
```

with solution

```
Time: 0.025891780853271484s
plan:
go west npc camp path
go west npc path cliff
go up npc cliff clifftop
go east npc clifftop bushwood
get stick npc bushwood
go east npc bushwood clifftop
go down npc clifftop cliff
go east npc cliff path
go east npc path camp
get_tarp npc camp tarp
construct_shelter npc camp stick tarp leave
```

5 What limitations of PDDL did you encounter that makes it difficult to precisely convert a wikiHow description into PDDL?

Limitations of PDDL:

- Linguistic Ambiguity: The wikiHow is written by natural language, making it ambiguous to convert to the unambiguous planning symbols.
- Implicit Logic: The wikiHow article doesn't always come with clear logic, thus it's hard to convert them to unidirectional, explitic logic
- Inconsistent Granularity: The actions mentioned in wikiHow article are not consistent in granularity, the size of gap among actions are various procedurally, temporally, and spatially.

6 Could your PDDL be used as an interesting challenge for a text-adventure-style game? If so, how? If not, what would needed to create an interesting challenge?

Our PDDL could be an interesting challenge:

- Our PDDL could be for a survival kind of text-adventure-style game where you need to survive in a desert island while accomplishing multiple tasks.
- Our PDDL is based on real life survival skills so it makes it more realistic when you are using actual physics to solve problems.
- One limitation is that our current problems are all about survival. We might need to add more problems related to fighting the big boss in an adventure game to make people more engaged.

7 Discuss how you might use GPT-3 to automatically or semi-automatically convert a wikiHow article to PDDL?

Suggestions:

- Similar to what we did in homework 2, prompt GPT-3 with some examples with the context and generate mentions based on the context. And then prompt GPT-3 again with context, mentions, and generate different predicates. Then we prompt GPT-3 with context, mentions, and predicates and generate actions.
- We can also finetune GPT-3 to do the same thing as the previous bullet point.