

CIS-700 HW4 Writeup



March 3, 2022

1 What wikiHow article did you pick and why?

I picked an article on survival on a deserted island. I picked this article because it had some tasks that seemed fairly well-defined and straightforward to implement, as opposed to some other articles that had more open-ended goals and steps. Problems like getting water or building a fire have a clear goal. There may be multiple ways to accomplish the goal, but the end result is largely the same.

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

I translated the portions for finding water, building a fire, getting food, and making a signal. I picked these problems because finding water and making fire are logical starting points for survival, and cooking food and making a smoke signal come fairly naturally after making fire.

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

One action I used was bagging the leaves of a plant to collect water. I considered adding a 'wait' action to pass time for water to gather, but I ultimately decided to only use a single action to wait and gather the water, since my framework did not, and in this implementation would not, reflect the other consequences of waiting, such as hunger increasing or the time of day changing, which would likely require a more robust framework than simple types and predicates. Other than that, I made types for different kinds of wood needed to start a fire: tinder to form the base of the fire, and softwood and a hardwood stick to ignite through friction. The action to create a campfire required only tinder, and the action to light it required the two types of wood as well as being in proximity of the campfire.

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

The first goal was to collect freshwater, represented by having water in your inventory. The initial state had multiple connected locations, including a forest with plants, and a river. A player could solve the problem by collecting water from the river, or by using a bag to collect water from a plant. The second goal was to create a fire, represented by a campfire being lit. The initial state had similar locations, with a forest with tinder, softwood, and hardwood sticks. A player would have to collect all the kinds of wood from the forest, then choose somewhere to build and light the campfire with the wood they gathered. The third goal was to collect food, represented by having a cooked fish in inventory. The initial state included, in addition to the states of previous problems, a sharp rock and water sources with fish at the shore and river. To solve the problem, a player would need to make a fire as outlined before, use the sharp stone to make a spear, catch a fish at the shore or river with the spear, and cook it at the campfire. The final problem was to make a smoke signal. To solve this problem, a player would have to make a fire, then gather more tinder and wet it at a water source to make it produce more smoke when ignited, and then use it to make a signal at the campfire.

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

A major limitation for me was PDDL's restriction to boolean predicates. The world is represented as a set of distinct states, which makes it difficult to check for a range of acceptable preconditions. Another limitation is that everything that can exist in a problem must be defined at the start, which means that items cannot really be created in the process of solving a problem. For example, making a spear involves losing a stick and gaining a spear which has always existed but has been inaccessible up to that point. To use more than one item, multiple items must be predefined, or the item must be reusable. Another limitation is the lack of "chaining" effects. For example, waiting for water to gather has the effect of time passing, which in turn should increase hunger and thirst and progress the time of day. In PDDL, not only is it difficult to effectively increment a value for time, we must also directly implement all the effects of such an action, instead of having variables that change automatically as others do.

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?

As it is now, I do not think my PDDL could be used as an interesting challenge for a text-adventure game. The tasks are not very hard to solve, and there is no real sense of danger. Multiple improvements are needed to create an interesting challenge. First, there need to be failure conditions, foremost among which are hunger, thirst, and the need for shelter, all of which would benefit greatly from a system to track the passing of time. A game could also benefit from adversarial agents or enemies to fight, but PDDL also does not work well for autonomous agents, as their actions would need to be coded in as part of the solution, which does not make sense, or they would need to happen as results of other actions, which is hard to implement.

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

GPT-3 could realistically be used to parse wikiHow articles into distinct steps, which could then be converted to event representations, possibly including the use of WordNet synsets. Given such event representations, such as (*entity1, verb, entity2*), we could create PDDL types for unique entities and create actions based on the events. Another possibility is creating a predicate based on the title of a step in wikiHow, and setting that as a goal. These approaches are still limited by the need to fill in details not necessarily included in the articles, as well as the existence multiple solutions to a problem that may be hard to separate if they appear in the same block of text in the article.