CIS 700 - Interactive Fiction and Story Generation
HW 4 Write-up
3 March 2022
[redacted]

Creating the PDDL files

For this assignment, we converted a wikiHow article titled, “How to Make Papyrus”, into PDDL files. Specifically, we translated this article into a single PDDL domain consisting of 3 problems: gathering the tools required to make papyrus at home, obtaining the papyrus plant from a local river, and making papyrus. We chose this article because we thought the topic was interesting and because it followed a singular and fairly unambiguous process that would be conducive to PDDL translation. Also, unlike many other articles on wikiHow, this did not consist of many instructions saying what *not* to do (which is less relevant and interesting for our goal of translating a task to PDDL).

To carry out this translation, we used parts 1, 2, and 3 from the article. The 3 problem PDDL files roughly correspond to each of the parts from the articles. However, we made some slight modifications. We expanded on part 1 as we implemented it into PDDL problem 1 by adding in actions relating to locations. Although location is relatively unimportant for this task and could have been left out, by including it, it gave us extra practice with incorporating the location type into our action functions and problem goals. We left out part 4, *Making Papyrus with Children*, as it was largely a rehashing of steps we had already implemented, but with slight alterations to make it easier for children to carry out the task.

The overarching goal of this domain is to create papyrus. The goal is achieved by completing each of the 3 problems in sequential order. The first problem consists of gathering the tools used to process the papyrus plant into papyrus paper. The player must be at their home and gather water, linen sheets, a rolling pin, a knife, and a wooden board. The second problem consists of going to a river and searching for and gathering the papyrus plant. Finally,

the third problem consists of utilizing the tools and materials to process the papyrus plant into a sheet of papyrus paper.

We made this goal obtainable, we wrote a series of action functions and preconditions. We relied heavily on predicates, especially in part 3, to track the state of the raw materials as they are being processed. For example, there are a series of actions that have an effect of changing the predicate for the raw papyrus materials. The action “soak_strips” has an effect of changing the predicate “soaked” to true for the item “papyrus_strips”. Similarly, “roll_strips” has an effect on the “dried” predicate, “weave_strips” have an effect on the “woven” predicate, and “bundle_strips” has an effect on the “finished” predicate. For problems 1 and 2, we also incorporated location variables and interacted with them through action functions such as “get” and “search_location”.

Text-Adventure Game Application

We think our PDDL files could easily be used to create a challenge in a text adventure game. For example, if you have an RPG game where an agent (NPC or player) asks another agent for papyrus in exchange for a reward, the files that we wrote could be used to carry out that task and check whether the task is completed. The assignee of the task would first go to their home to gather the tools, then to the river to gather the materials, and finally carry out the task of crafting the papyrus.

PDDL Limitations

One limitation of PDDL seems to be that it is not easy to keep track of a history of states. For our problem, this was not much of an issue, but for a problem where you need to keep track of how many times you carried out a task, or whether a player's actions follow a certain sequence in the correct order, this becomes much more difficult in PDDL. It also seems that PDDL would

make it difficult to keep track of continuous variables. For example, if you need to keep track of a player's health or how much water the player has in their canteen, this would be difficult.

Utilizing OpenAI API

We think it would be feasible to use the OpenAI API to convert wikiHow articles into PDDL provided that we prompt GPT-3 adequately. We would likely need to finetune GPT for different parts of the PDDL generation, similar to how we trained GPT for use cases in Action Castle. For example, one use case would be the generation of an action in PDDL format. To do this, the inputs to the GPT-3 prompt would be the different steps from wikiHow articles. The outputs would be the respective actions in PDDL format.

Another use case could be generating locations and the connections between them in PDDL format. For this, we could input full articles of wikiHow (those in which location is a significant component of completing the task at hand), and the outputs would be the part of the init section of PDDL problem file that specifies how different locations are connected to one another.

We are less confident that GPT-3 would be able to generate entire domain files in PDDL format. Although plausible, it seems that it would be difficult for GPT-3 to pick up on all of the nuances required to translate potentially complex problems into a syntax as unforgiving as PDDL.