— GUIDE —

Assignment 5: Escape the Wumpus Cave

AI-1 Systems Project (Winter Semester 2024/2025)

Jan Frederik Schaefer

Friedrich-Alexander-Universität Erlangen-Nürnberg, Department Informatik

This document is intended to help you solve the assignment "Assignment 5: Escape the Wumpus Cave" [AS]. You do not have to read it, but we do recommend to at least take a look at the tips and common issues.

1 A few tips

- Don't try to support all cell types at once. It is probably easier to start with the easier maps to get a feeling for PDDL.
- The actions in your PDDL file don't have to directly correspond to the "solution" actions described in **??**. The only condition is that the "solution" actions can easily be generated from the PDDL actions.
- Be aware that planners usually support only a subset of the PDDL format.
- Make your own minimal test maps to debug your implementation.
- There is no need to find an optimal (e.g. shortest) plan.
- You do not need to use advanced PDDL features. Generally, a key challenge is to adopt the "PDDL thinking" in terms of predicates, objects and sets of facts. For example, you do not have to use something like counter variables for items that you picked up. Instead, you can make an object for each item and then use a predicate to indicate that it is available for use.
- Keep actions simple and rather use multiple actions for different variations. Some students use lots of nested `when`s, which gets really difficult to understand and debug.
- Have fun!

2 Planners

You should use a PDDL planner for this problem. PDDL planners require two files: a domain file, which we will call `wumpus.pddl`, and a problem file, which we will call `map.pddl`. The domain file should describe the rules of the Wumpus world in general and the problem file specifies the details of a particular map. A planner can then find a plan that solves the problem.

You can use the `fast-downward` planner or `pyperplan` (see below). `pyperplan` is less efficient and has fewer features, but has the advantage that it is easier to install. It might make sense to use pyperplan for getting started, but you might have to switch to `fast-downward` if efficiency becomes an issue. If you want to use a different planner, please ask in the problem channel first.

2.1 `fast-downward` (recommended)

`fast-downward` [FD] is a heavily optimized planner that was quite successful in the international planning competition (IPC). You can try it out online at [FDW]. To run it on your machine, you can build it yourself (which worked very well for me, but was a problem for Windows users in the past) or run it using docker.

2.1.1 Building and running `fast-downward` yourself

After cloning the repository, you can build `fast-downward` with

```
python3 build_configs.py
python3 build.py
```

Then you should be able to run it with

```
python3 fast-downward.py --alias lama-first \
    --plan-file /path/for/outputfile \
    /path/to/wumpus.pddl \
    /path/to/map.pddl
```

You can pick other modes than `lama-first`. Note that the `lama-first` mode does not attempt to find an optimal plan (i.e. a plan with minimal cost/a minimal number of actions).

2.1.2 Using Docker

If you have docker installed, you can simply run `fast-downward` with

```
docker run --rm -v "/path/to/my/files:/files" \
    aibasel/downward --alias lama-first \
    --plan-file /files/outputfile \
    /files/wumpus.pddl /files/map.pddl
```

The first line maps the directory `/path/to/my/files` to the docker volume `/files`. That means that `/path/to/my/files/xyz` on your machine will correspond to `/files/xyz`.

2.2 `pyperplan`

`pyperplan` [Alk+20] is implemented in Python and used as a teaching tool. As such, it is not heavily optimized, but it is easier to install and use. The README file on github provides good instructions for installation and use.

References

[Alk+20] Yusra Alkhazraji et al. *Pyperplan*. https://doi.org/10.5281/zenodo.3700819. 2020. DOI: 10.5281/zenodo.3700819. URL: https://doi.org/10.5281/zenodo.3700819.

[AS] *Assignment 5: Escape the Wumpus Cave*. URL: https://kwarc.info/teaching/AISysProj/WS2425/assignment-1.5.pdf.

[FD] *Fast Downward*. URL: https://www.fast-downward.org/ (visited on 02/15/2022).

[FDW] *Fast-Downward Planning on the Web*. URL: https://lcas.lincoln.ac.uk/fast-downward/ (visited on 05/29/2024).