Intelligent Systems Assignment 1

Christoph Royer [12004184]

May 7, 2025

1 Scotland Yard

See submission files.

7. Reach of X: I have proven that X cannot reach every location after exactly 2 steps.

2 Cat World

See submission files.

4. Environment properties

- Partially observable: The location of the cat is not directly shown.
- Deterministic: no randomness is employed - cat, pawprints, pots and hiding spots are either at a place or they are not.
- Episodic: at the time of observation, the cat does not move and the environment does not depend on a previous time step.
- Static: The environment does not change.
- Discrete: The placement of the tiles is in discrete increments (n by n board), and the placement of cat, pots and hiding spots is a binary statement.
- Single agent: The cat is the only agent that can be said to be active, we are just observing its (past) actions.

3 Background Knowledge

3.1 1. Monotonicity of ASP

The program:

```
fast(X) :- car(X), not vw_golf(X).
car(louise).
#show fast/1.
```

Produces the answer set:

```
fast(louise).
```

However, when expanding the program with the knowledge that Louise is a VW Golf:

```
fast(X) :- car(X), not vw_golf(X).
car(louise).
vw_golf(louise).
#show fast/1.
```

The answer set is empty.

Thus, by adding an assumption (`vw_golf(louise)`), we could no longer prove `fast(louise)`; ASP is non-monotonic.

3.2 Predicate completion

I mainly used predicate completion (`<=>`) in exercise 1. With this, I ensured that a negative statement was also defined (e.g. that a player does not start on a location).

I did however omit this in some cases where the reverse is not necessarily true. For example, a connection between two variables entails that they are locations, but both being locations does not entail them being connected.

If I did not use predicate completion, I would have to define some of the following formulas differently not to rely on the definition of negative statements, rather only on true ones.