IASD 2022/23 Project Assignment #2: ‘Roll the Ball’ Slide Puzzle

Luis Custódio and Rodrigo Ventura
Instituto Superior Técnico, University of Lisbon

(Version 1.0, October 7, 2022)

Introduction

Free the Ball, or Roll the Ball, is a classic tile puzzle where the goal is to move the sliding tiles to unblock a path for the ball to roll to the exit.

The goal of the project is to develop a program using Python (version 3) to search for a solution, if any, for a ‘Roll the Ball’ puzzle. The project will be divided in three parts, and three deliverables. This document defines the second assignment and deliverable.

Notice that you should read again Assignment #1 to remember the rules of the puzzle and the input file format.

1 Second Assignment and Deliverable

The second assignment is to develop a Python program capable of reading a puzzle (see Assignment #1) and solving it using an uninformed search algorithm.

Associated with the IASD course book, there is a Python code repository with many Python files for the different chapters of the book[1].

In the repository you have the file `search.py` where you can find the definition of the class `Problem`, the superclass of the class RTBProblem, examples for other problems, and the functions that implement each one of the search algorithms mentioned in the IASD lectures. For this assignment, you only need to import search.py, which in turn imports utils.py. Both files are available in the repository. Copy both files to your work directory while developing and testing on your computer. Nothing else is needed.

The Python program to be delivered should be called `solution.py` and include (at least) a class with name `RTBProblem` containing (at least) the following methods (see Annex A for the class template):

`result(state,action)` returns the state that results from executing the given action in the given state.

`actions(state)` returns the actions that can be executed in the given state.

`goal_test(state)` returns `True` if the given state is a goal.

`load(fh)` loads a puzzle (initial configuration) from a file object fh

`setAlgorithm()` sets the uninformed search algorithm to be used

`solve()` calls the uninformed search algorithm that was set.

In order to solve a puzzle you need to:

- read the initial configuration of the puzzle from a file object like you did for Assignment #1,
- decide and implement a representation for the state of the problem. You may change the representation used for Assignment #1 if you find a better one, and represent the initial state with that representation, and saved in the object RTBProblem (`self.initial`),
- implement the methods `result`, `actions` and `goal_test` as described above,
- choose one uninformed search algorithm that the group considers best fits the problem at hand. See `search.py` for the set of uninformed search algorithms available.

[1] The link for the repository is https://github.com/aimacode/aima-python.

2 Evaluation

The deliverable for this assignment is shown through DEEC Moodle, with the submission of a single python file, called `solution.py`, implementing the modules mentioned above. Instructions for this platform are available on the course webpage. Finally, the grade is computed in the following way:

- 50% from the public tests;
- 50% from the private tests; and
- -15% from the code structure, quality and readability.

Deadline: **21-October-2022**. Projects submitted after the deadline will not be considered for evaluation.

Closing Remarks on Ethics:

- Any kind of sharing code outside your group is considered plagiarism;
- Developing your code in any open software development tool is considered sharing code;
- You can use GitHub. Make sure you have private projects and remove them afterward;
- If you get caught in any plagiarism, either by copying the code/ideas or sharing them with others, you will not be graded; and
- The scripts and other supporting materials produced by the instructors cannot be made public!

A Class Template

```
import search
""" For this assignment, you just need to import search.py,
which in turn imports utils.py; both files are available
in the repository mentioned on page 2 of the Assignment"""

class RTBProblem(search.Problem):

        def __init__(self):
                """Method that instantiate your class.
                     You can change the content of this.
                     self.initial is where the initial state of
                     the puzzle should be saved."""
        self.initial = None
        self.algorithm = None

    def result(self, state, action):
        """Return the state that results from executing the given act
        pass

    def actions(self, state):
        """Return the actions that can be executed in the given state
        pass

    def goal_test(self, state):
        """Return True if the state is a goal."""
        pass

    def load(self, fh):
        """loads a RTB puzzle from the file object fh.
            You may initialize self.initial here."""
        pass

    def setAlgorithm(self):
        """Sets the uninformed search algorithm chosen."""

        self.algorithm = search. ...
        # example: self.algorithm = search.breadth_first_tree_search
        # substitute by the function in search.py that
        # implements the chosen algorithm.
        # You can only use the algorithms defined in search.py

    def solve(self):
        """Calls the uninformed search algorithm chosen."""

        return self.algorithm(self, ...)
        # You have to provide the arguments for the
        # chosen algorithm if any.
        # For instance, for the Depth Limited Search you need to
        # provide a value for the limit L, otherwise the default
        # value (50) will be used.
```