CIT CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Assignment 1 - Planning & Scheduling

Due date:

Assignment should be submitted to Canvas before 11PM **Sunday March 8th**

The Wolf, Goat, and Cabbage -- State-transition systems

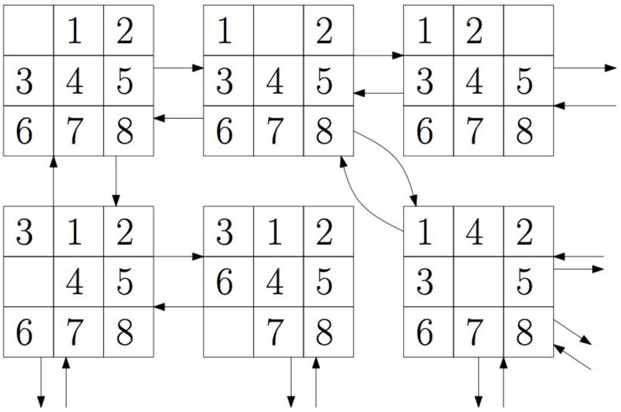
A man has to take a wolf, a goat, and some cabbage across a river. His rowboat has enough room for the man plus either the wolf or the goat or the cabbage. If he takes the cabbage with him, the wolf will eat the goat. If he takes the wolf, the goat will eat the cabbage. Only when the man is present are the goat and the cabbage safe from their enemies. All the same, the man carries wolf, goat, and cabbage across the river. How?

Exercise: [25 Marks]

Formulate the problem precisely and draw a state-transition diagram of the complete state space.

A* - Sliding Tile Puzzle [35 Marks]

The sliding tile puzzle is a simple toy that consists of a number of tiles, and an empty space that can be used to move tiles around. The sliding tile puzzle defines a graph, where each state of the puzzle is a vertex, and there are edges between states if they can be reached by a single move:



We can define the planning problem as a search problem by giving:

- An initial state.
- A goal state
- A set of operators that define how to create states from other states.

A deterministic plan is a sequence of actions to achieve a goal from a given starting state. A deterministic planner is a problem solver that can produce a plan. The Input is an initial state and description, a specification of the actions available to the system, and a goal description. The specification of the actions includes their preconditions and their effects.

A*

In this assignment , you will implement a program that uses A* search to solve the Eight Puzzle game. Your program will take an initial board position, and then try to find a sequence of moves from the initial position to the goal position. Moves are represented by moving the black space left, right, up, or down.

Manhattan distance: this heuristic is used for its simplicity and also because it underestimate (i.e., lower bound) on the number of moves required to bring a given board to the solution board. We simply compute the sum of distances of each tile from where it belongs, completely ignoring all other tiles. For example:

```
2 1 3     1 2 3        1 2 3 4 5 6 7 8
5 4 0     4 5 6        ---------------
6 7 8     7 8 0        1+1+0+1+1+3+1+1 = 9
```

This is so, because the tile "1" is 1 move away, the tile "2" is 1 move away, the tile "3" is 0 moves away, the tile "4" is 1 move away, the tile "5" is 1 move away, the "6" tile is 3 moves away, the "7" is 1 move away, and the "8" is 1 move away. Note that we do not consider the empty space

Important: to implement your search algorithm, you will need to:

- Implement an open list, that allows you to add a state /priority pair and remove the pair with the smallest priority
- Implement a closed list, that allows you to add a state, and check if a state is in the closed list
- Implement a search state, that allows you to compare two states, find the estimated cost from one state to another, and generate all children of a given search state

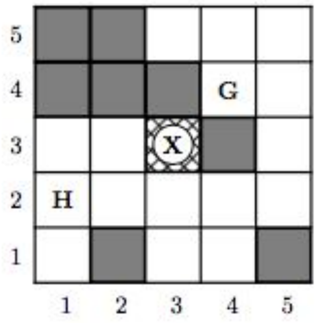
Exercise:

- Implement the A* algorithm using the number of misplaced tiles as a heuristic function.
- Replace the previous heuristic with the manhattan distance.

Blackbox planner [40 Marks]

Hansel and Gretel are lost in a rectangular forest. Even worse, they are separated from each other. Fortunately, there is the friendly neighborhood witch who is willing to help them reunite. As an appropriate place for the meeting she proposes her gingerbread house in the middle of the forest.

Hansel and Gretel can both move in four directions, namely north, east, south and west. Call the actions north$_H$, south$_H$, east$_H$, west$_H$, north$_G$, south$_G$, east$_G$, and west$_G$, respectively, with the obvious semantics of moving either Hansel or Gretel by one grid cell in the indicated direction. Some cells with extremely thick vegetation are inaccessible (shown in grey in the figure below) and cannot be moved to. The goal is to move both Hansel and Gretel to the goal position in the center of the grid, marked by X. Their current positions are indicated by H and G, respectively.



Exercise:

Solve the problem (and help Hansel and Gretel) using PDDL and the blackbox solver.

- You need to create two separate files (domain file and problem file).

Rubric:

The deliverable of this project will consists of a python code and PDDL files (and a report for the state-transition system). The following is the allocation of marks.

The state-transition system well designed, documented and explained (19 - 25 Marks)	The state-transition system is logically well designed (13-18 Marks)	The state-transition system has slight logic errors that do no significantly affect the results. (7-11 Marks)	The state-transition system has significant logic errors (3-6 Marks)	The state-transition system is incorrect (0-2 Marks)

The A* algorithm is logically well designed without inappropriate design choices (e.g., unnecessary loops) The implementation support both heuristics	The A* Algorithm always works properly and meets the specification of the algorithm The implementation supports at least one of the two heuristics	The A* Algorithm works properly in limited cases	The A* algorithm is incorrectly implemented
(27-35 Marks)	(18-26 Marks)	(8-17 Marks)	(0-7 Marks)

	The solution is logically well designed	The solution has slight logic errors that do no significantly affect the results.	The solution has significant logic errors	The solution is incorrect
Domain (PDDL)	(14-20 Marks)	(7-13 Marks)	(3-6 Marks)	(0-2 Marks)
Problem (PDDL)	(14-20 Marks)	(7-13 Marks)	(3-6 Marks)	(0-2 Marks)