

Problem Wk.13.3.7: Two robots on a grid [Optional]

You are given a state machine class `RobotMoves` that encodes the legal actions and successor functions for a single robot moving on a grid. The state is represented as a tuple of (x,y) grid indices and the actions are: 'up', 'down', 'right', 'left'. The machine has the initial state as its single `init` argument.

We want to write a program that uses search to plan paths for a pair of robot on an infinite grid. For example we want to find paths for the robots to meet in the same grid squares in the minimum number of steps.

We will use the `search.smSearch` function to plan the paths.

1. Assume the `sm1` and `sm2` are two instances of `RobotMoves`. Use state-machine combinators (read about them in Chapter 4 of the notes) to construct a state machine instance that represents the motions of the two robots. Set the composite state machine to be the value of the variable `m`.
2. Construct a goal function for the search; it should returns `True` when both robots are in the same grid square and `False` otherwise.

```
m = None

def goalFn(s):
    pass
```

MIT OpenCourseWare
<http://ocw.mit.edu>

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.