Homework Problem 1

Formal Methods in Robotics (Fall 2019)

In this problem, we will construct a **Gridworld** transition system (TS) for a two-player turn-based game between players `P1` and `P2` using `IGLSynth` tool. We will do this by implementing a `generate_graph` function. The following write-up gives a top-level idea of the provided code-base.

Background

Recall a TS is defined as

$$TS = \langle S, Act, \delta, AP, L \rangle \tag{2}$$

In `IGLSynth`, the component of `TS` given by $\langle S, Act, \delta \rangle$ is represented as a graph with action-labeled edges using a data structure `iglsynth.game.tsys.TSys`, which represents a deterministic transition system. Note that `TSys` derives from class `iglsynth.util.graph.Graph` and inherits all of its methods and properties. Refer to documentation of the classes for more information.

A grid world is defined as the class `iglsynth.game.gridworld.Gridworld`, which is a sub-class of `iglsynth.game.tsys.TSys`. The gridworld defines a vertex and an edge class to represent the vertex and edge properties. See documentation of `Gridworld.Vertex` and `Gridworld.Edge` for methods and properties associated with them.

The `Gridworld.generate_graph` function generates the graph of grid world by adding all vertices and edges based on dimensions, connectivity and kind of grid world. In this problem, we assume the kind to be `TURN-BASED`. Update the given template to implement the `generate_graph` function.

Note: The **TODO** tags identify the locations where you need to update the code.

Why is Action Class in gw_graph.py?

Because the `Action` class API is still experimental.

An `Action` is a `Callable` object that acts on a `Vertex` object to return a new `Vertex` object. See definition of `N, E, …` actions to understand how to define and implement actions.

[Any suggestions or feedback on Action class API would be greatly appreciated.]

Running the code

1. If using PyCharm, just `Run` the file. PyCharm will take care of configurations for you.
2. If you want to use terminal, then

```
PC$ docker exec -it <docker-container name> /bin/bash
Docker$ cd /home/iglsynth/
Docker$ python3 -m pytest FMR_HW/gw_graph.py
```

Check if all tests are pass or not. Ignore any warnings.