QR project documentation

Rasyan Ahmed (10784063) & Michelle Appel (10170359)

1 4a

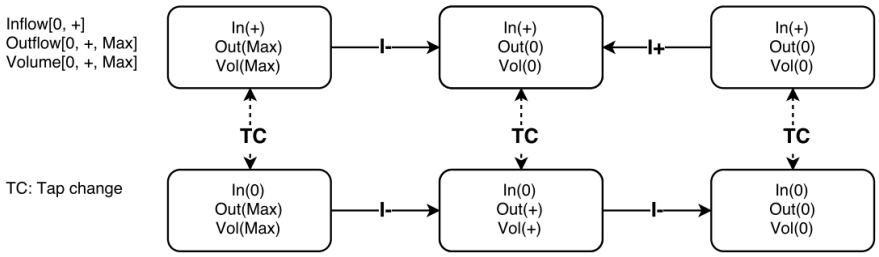

Figure 1: Graph showing all possible states and their transitions.

There are two possible values for the inflow and three possible values that are always the same for the volume and outflow. this results in 2 x 3 = 6 possible states.
Since the only form of change in this system that can happen at random is the opening and closing of the tab (inflow) we have decided to simplify the problem and not use a double notation where each possible quantity has a magnitude value as well as a derivative value as this would dramatically increase the possible number of states.
Instead we simply add the Tab Change action as a possible transition for each state. This action changes inflow from the current value to the only other value available. This inclusion should not change the behaviour of this system but only simplify its representation.

There is also some ambiguity in the system definition so we have made some further assumptions:
At maximum outflow, the outflow outpaces the inflow of a open tab, this means that in such a situation the volume (and therefor the outflow) would decrease in the next state. When the inflow is positive and the outflow is also positive but below maximum, we assume that an equilibrium would be found so that they are equal and no changes to the volume occur.

The tab change action happens at instant speed so no other changes happen from the current state to its child state. In the graph this action is represented by a dashed line.
To find all possible states by exploring from one state you have to start and one of the two states where the volume (and therefor the outflow) are at maximum since these two states can not be found from any of the other states. We assume therefor that our exploration starts from one of these states (important for 4b and 4c).

2 4b

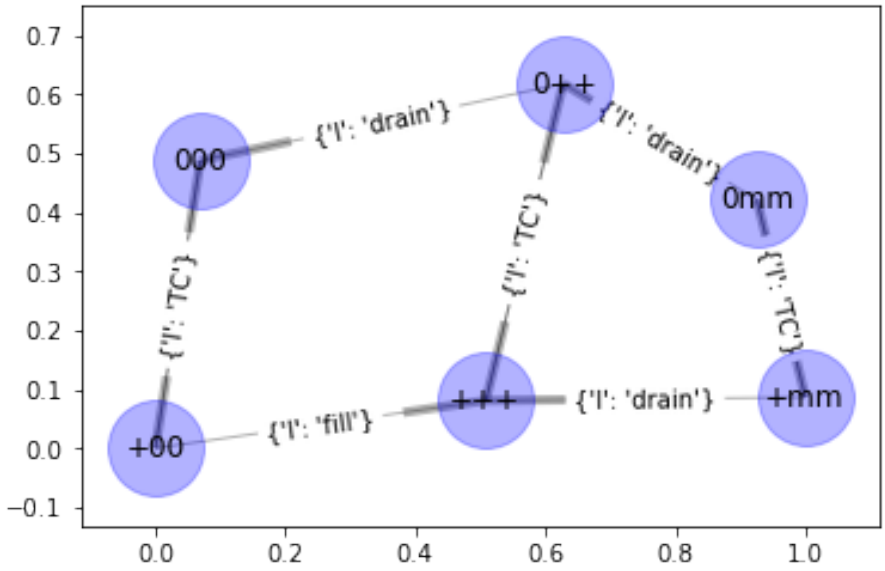

Figure 2: Graph made automatically with the code in QRproject.ipynb file.

The implementation of this algorithm was done in the python programming language in the Ipython notebook format of Jupyter notebook which is available as part of the anaconda python package.

For the implementation of this algorithm we have simplified the notation of each state into a three letter string. the first letter signifies the value of the inflow, the second the volume and the final one the outflow.
The edges of the graph are directional, the thin part represent the starting point and the larger part represents the arrowhead (ending point of the graph), the tab changes have an arrowhead on both sides as they go both ways.
The algorithm itself is a basic depth first search tree exploring algorithm which

finds the child nodes of the current node and adds each yet unseen child node into the exploration stack. The next item in the stack is then explored and this repeats until all nodes in the stack are explored and therefor all possible nodes have been found. For this algorithm to work you have to use one of the two states with maximum volume as the starting node.
For more details on the details on the implementation please see the code itself which is commented.

3 4c

The trace tells you in clear natural language what situation each node represents and what action it performs to get find its child nodes. This has been done by placing print statements throughout the code of the algorithm.
Below is a copy of the output of the trace. As the algorithm is deterministic, all possible runs will generate the same trace.

3.1 Output of the trace

current node = +mm
Node situation: Water is flowing from the tab and the kitchen sink is full which means the water is draining as fast as possible.
Two child nodes have been found: 0mm and +++
0mm is the result of simply closing the tab, this is done instantly so no other changes take place

This node has not been seen before, so it will be added to the exploration stack.
+++ is the result of there being more outflow of water then inflow, so some of the water is drained away.

This node has not been seen before, so it will be added to the exploration stack.

current node = +++
Node situation: Water is flowing from the tab and there is water in the kitchen sink which means that some of the water is draining away.
Two child nodes have been found: 0++ and +++
0++ is the result of simply closing the tab, this is done instantly so no other changes take place

This node has not been seen before, so it will be added to the exploration stack.
+++ is the result of there being no inflow nor outflow of water, so nothing changes.

current node = 0++

Node situation: The tab is shut tight and there is water in the kitchen sink which means that some of the water is draining away.
Two child nodes have been found: +++ and 000
+++ is the result of simply opening the tab, this is done instantly so no other changes take place
000 is the result of there being more outflow of water then inflow, so some of the water is drained away.

This node has not been seen before, so it will be added to the exploration stack.

current node = 000
Node situation: The tab is shut tight and The kitchen sink is empty therefor there is no outflow of water.
Two child nodes have been found: +00 and 000
+00 is the result of simply opening the tab, this is done instantly so no other changes take place

This node has not been seen before, so it will be added to the exploration stack.
000 is the result of there being no inflow nor outflow of water, so nothing changes.

current node = +00
Node situation: Water is flowing from the tab and The kitchen sink is empty therefor there is no outflow of water.
Two child nodes have been found: 000 and +++
000 is the result of simply closing the tab, this is done instantly so no other changes take place
+++ is the result of the empty sink filling up due to the tab being open without there being any outflow.

current node = 0mm Node situation: The tab is shut tight and the kitchen sink is full which means the water is draining as fast as possible.
Two child nodes have been found: +mm and 0++
+mm is the result of simply opening the tab, this is done instantly so no other changes take place
0++ is the result of there being more outflow of water then inflow, so some of the water is drained away.