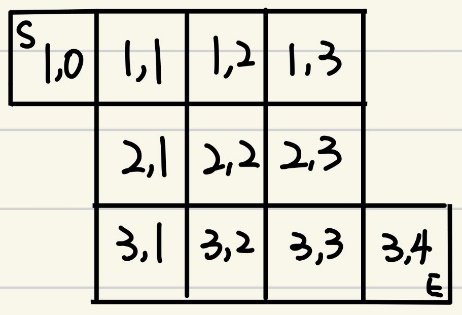
# Group 6: Miuki, Ming, and Joshua

# Project Summary

*This game is about connecting pipes in a 3x3 grid, where the start is in the top left corner and the end is in the bottom right corner. This game aims to connect the starting pipe, the ending pipe, and every pipe in between them (there are 5 types of pipes) by rotating the pipes on the grid. The water can only flow forward and downward. We made a simple diagram to get a better idea of ​​what the entire grid looks like. We will simulate how pipes are placed for the grid to have a solution.*

# Propositions

* Location (p, loc): There is a pipe with a certain pipe orientation at location loc. There are 11 locations available for a 3x3 grid and 3 pipes with a total of 12 pipe orientations.
* NeighborLR (l1, l2): location l1 and l2 is neighbor and they are beside each other
* NeighborUD (l1, l2): location l1 and l2 is neighbour and one is on top of the other
* Connected (l1, l2): location l1 and l2 is connected
* Have\_to\_east (loc): the pipe on loc has an opening facing east
* Have\_to\_south (loc): the pipe on loc has an opening facing south
* Have\_from\_west (loc): the pipe on loc has an opening facing west
* Have\_from\_north (loc): the pipe on loc has an opening facing north

# Constraints

One location:

* There is only one pipe orientation of one pipe at location 10 and location 34(the start and end piece)

the opening of the start piece at location 10 can only face east

Location (E, 10)

the opening of the end piece at location 34 can only face west

Location (W, 34)

* All other locations (locations that are not 10 and 34) will not have opening only facing east or only facing west
* If there is a certain pipe at a location, then that location can have a different pipe orientation at the same location but not a different pipe

Location (straight pipe,11)>>add\_exactly\_one (Location (EW,11), Location(NS,11))

(If there is a straight pipe (orated in EW or NS) at location 11, then location 11 can have exactly one pipe orientation from (EW, NS). But the other 2 pipes in a total of 8 pipe orientations can not be at location 11)

* For all locations except for 22, 10, and 34, there is one pipe orientation from one pipe depending on their location and pipe on them and all other orientation will not be allowed in this grid cell

Ex. Location (3-opening-pipe orientation,11)>> Location(['S','E','W'], 11) or

Location (3-opening-pipe orientation,11)>> (~Location (['N', 'S', 'E'], 11) & ~Location (['N', 'S', 'W'], 11) & ~Location (['N', 'E', 'W'],11))

* If a pipe is orientated in a way at a location, then that location will not have the openings that the pipe orientation does not have

(ex. Location (NS,11)>> (~Have\_to\_east (11) **&**~Have\_from\_west (11)))

* If location loc has no opening toward east, then location loc can’t be connected to the grid cell on its right

~ Have\_to\_east (loc)>>~Connected (loc, loc+1)

* If location loc has no opening toward south, then location loc can’t be connected to the grid cell below it

~ Have\_to\_south(loc)>> ~Connected (loc, loc+10)

* Same constraint for opening west and north ^

Two locations:

* If locations differ in exactly 1 or 10, they will be neighbor

(ex. 10 and 11 will be NeighborLR, and 11 and 12 will be NeighborUD)

* Every location does not differ in 1 or 10 will not be neighbor

(ex. ~NeighborLR (10, 34) & NeighborUD (10, 34))

* If two locations are not neighbors, they are not connected (ex. 10 and 34 is not connected)
* No connection upward or left.

(Connected (l1, l2) **~** Connected(l2, l1) where l2>l1)

Whole grid

-there is a solution if there are a connection from start to end

(Connected (10, 11) & Connected (11, 12) & … & Connected (33, 34)) | … | (Connected (10, 11) & Connected (11, 21) & … & Connected (33, 34))

# Model Exploration

* For the model exploration, we first tried to set first row all to straight pipe with the function no\_sol\_with\_row\_strai (). According to our model, this should have no solution

A screenshot of a computer

Description automatically generated

A picture of the no\_sol\_with\_row\_strai in code

A black background with white text

Description automatically generated

Output when execute no\_sol\_with\_row\_strai ()

This setup does not have a solution because even though element in each row have connection between them, but there is no connection between rows which is needed for the constraint for the whole grid.

#may need detail

* Remove a pipe in solution route when it only has one solution

A screen shot of a computer

Description automatically generated A screenshot of a computer screen

Description automatically generated

On the left is a test case we used for our model and on the right, it shows that this test case has solutions and display all the connections this test case has. And there are one solution routes for this one is: location 10 to 11 to 21 to 31 to 32 to 33 to 34.

For the model exploration, we want to remove a pipe in solution route when it only has one solution. Then, according to our model, this should not have a solution

A screen shot of a computer program

Description automatically generated A screen shot of a computer

Description automatically generated

Code for the function and output of executing the function

But instead of saying no solution for this grid, solver did give a solution that is not right (it has connection for grid cell 21). Since the condition for two grid cells to be connected is if they are neighbours and the pipes on them have opening towards each other, and 21 has no pipe on it means that there should be no connection to it and from it. We soon realized that we are missing a constraint and a proposition that checks if there are empty grid cells. Before this function, we assumed that all gird cells have a pipe on it.

A black screen with white text

Description automatically generated

Add constraint to check if any grid cell is empty

A screen shot of a computer code

Description automatically generated

Add constraint: check all location that is neighbour, if any location in a neighbour pair empty, there should be no connection to that cell

After adding the codes in the pictures, now executing the function empty\_grid\_cell () give us an output saying there are no solutions.

# Jape Proof Ideas

1. When there are left and right connected grids, it may not be necessary to look at the up and down of the left grid to see whether they are connected or not.
2. No need to consider optimal solutions.First-Order Extension

*Describe how you might extend your model to a predicate logic setting, including how both the propositions and constraints would be updated.* ***There is no need to implement this extension!***

# Useful Notation

*Feel free to copy/paste the symbols here and remove this section before submitting.*