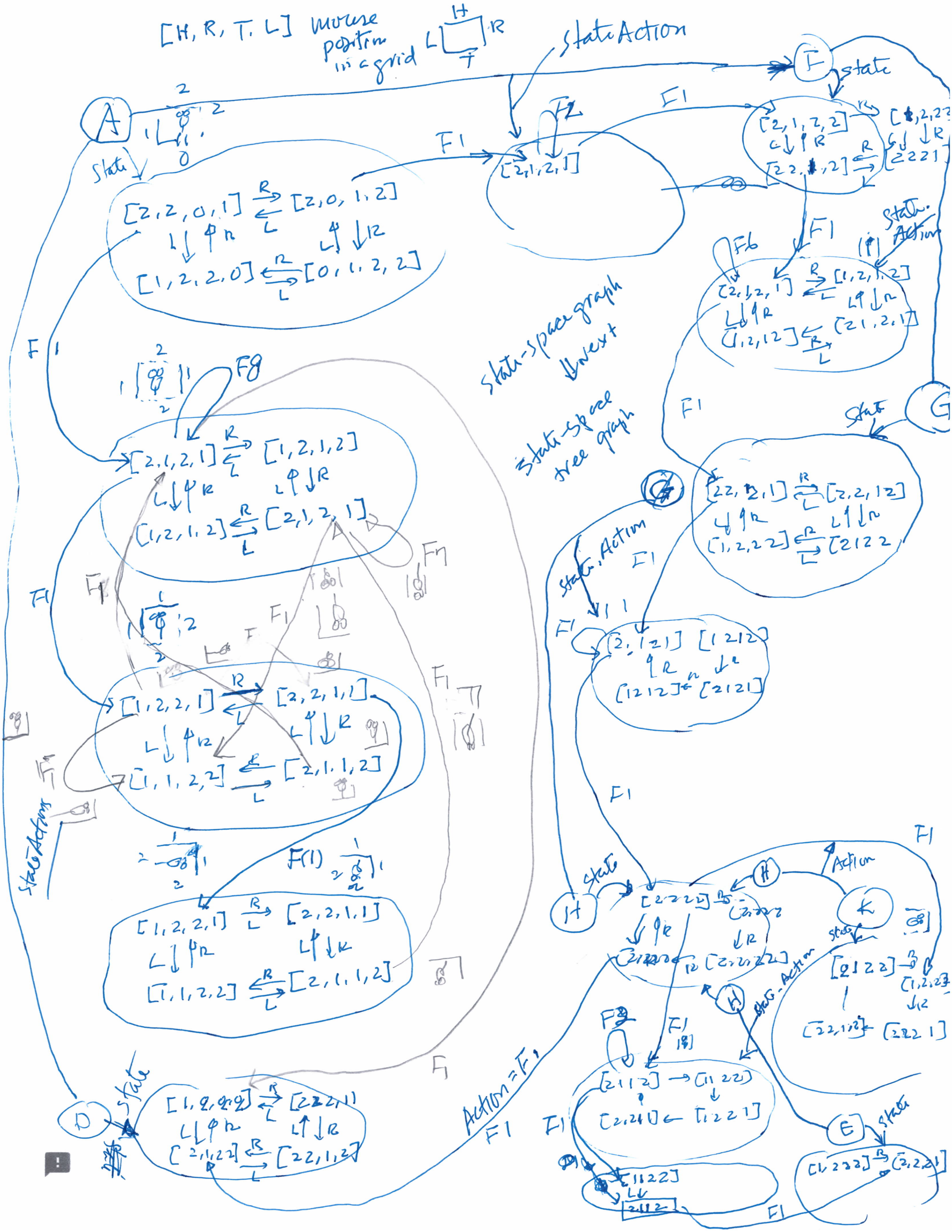
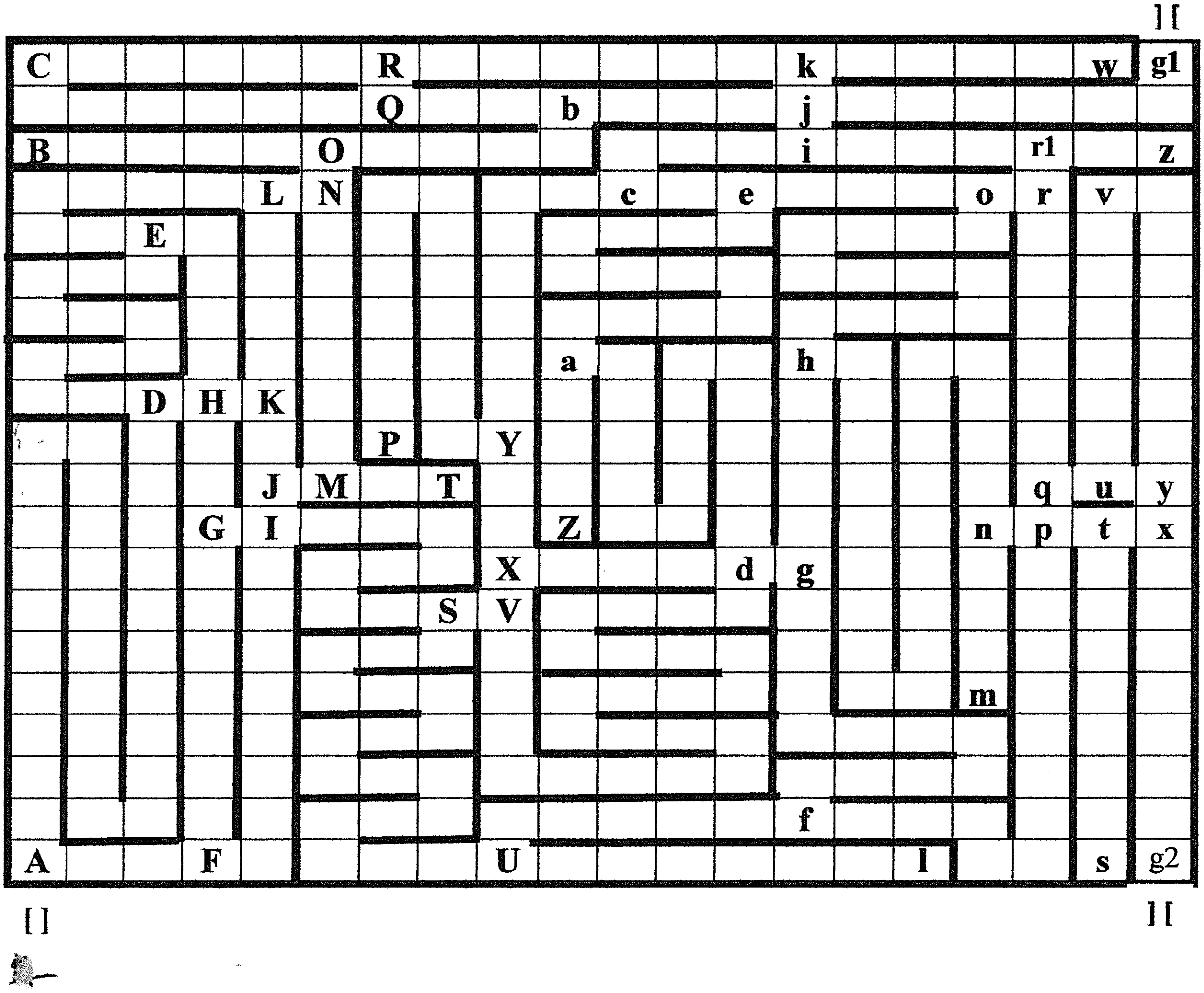

[H, R, T, L] mouse position in a grid
StateAction
state
F1
F2
F6
F7
F8
F1
[2, 2, 0, 1] [2, 0, 1, 2]
[1, 2, 2, 0] [0, 1, 2, 2]
[2, 1, 2, 1] [1, 2, 1, 2]
[1, 2, 1, 2] [2, 1, 2, 1]
[1, 2, 2, 1] [2, 2, 1, 1]
[1, 1, 2, 2] [2, 1, 1, 2]
state-space graph
next
state-space tree graph
State Action
Action = F1
A
D
E
F
G
H
K

The following graph is a maze example: Assume A is the entrance [ ]. Both g1 and g2 are exits] [.



CS 380 As 01 01162023

Future Research (I will discuss the following in class and will take time to develop its description):

Develop a theory for claiming annotations, A, B, …, Z, a, b, …, z, which are necessary and sufficient. Obtain a connected undirected weighted graph from a given maze. Will this obtained graph uniquely correspond to the given maze?

Design and implement the agent (the maze problem-solver, a mouse, a program, or a simulator), such that for given any maze, the agent can find an exit efficiently, once it enters the maze.

Input to the agent is any maze. (the agent can take different mazes as its input).

The agent consisting algorithms that traverse a given maze to find an exit.