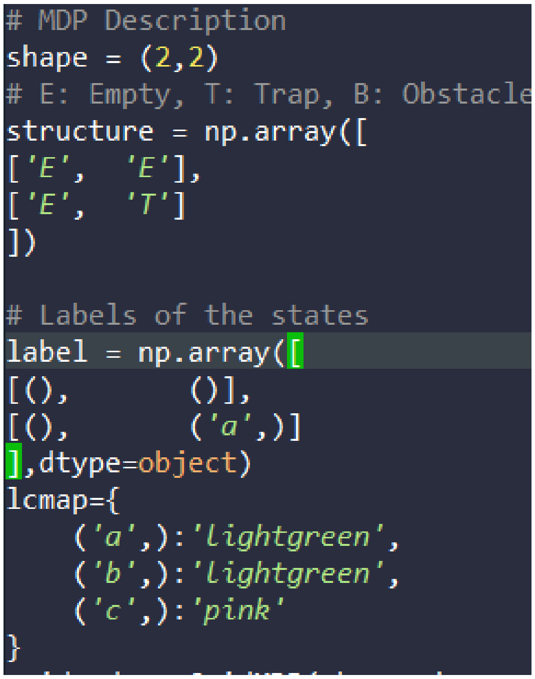
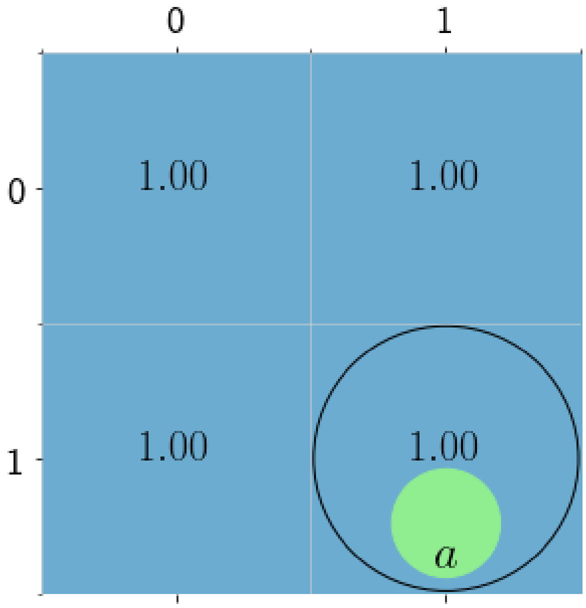
Q_learning, 2 by 2 grid, FG a



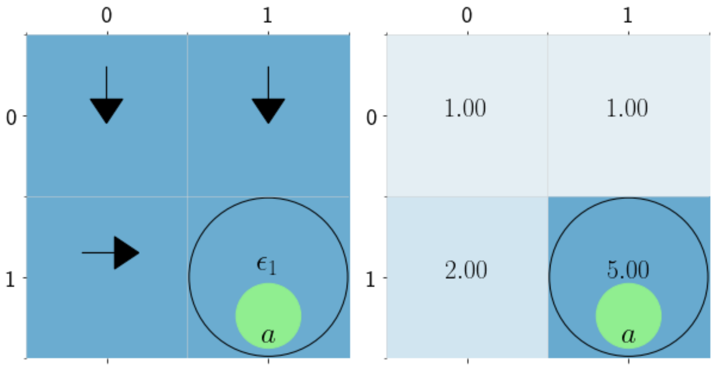
```
# MDP Description
shape = (2,2)
# E: Empty, T: Trap, B: Obstacle
structure = np.array([
['E',  'E'],
['E',  'T']
])

# Labels of the states
label = np.array([
[(),        ()],
[(),        ('a',)]
],dtype=object)
lcmap={
    ('a',):'lightgreen',
    ('b',):'lightgreen',
    ('c',):'pink'
}
```

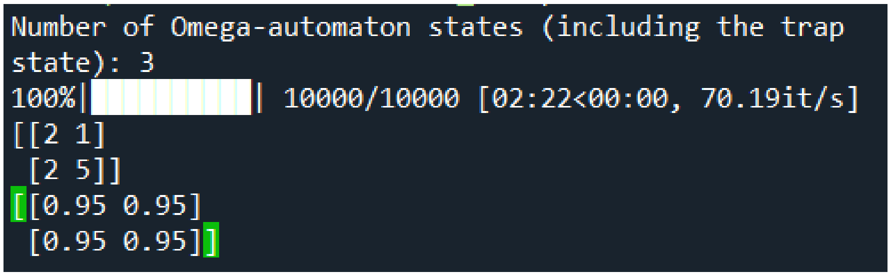
Value,



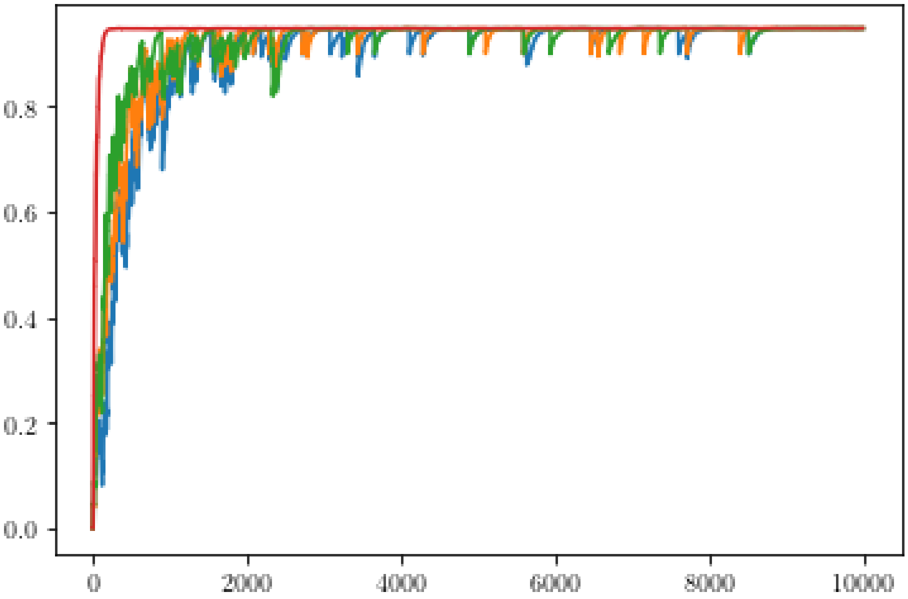
Policy,



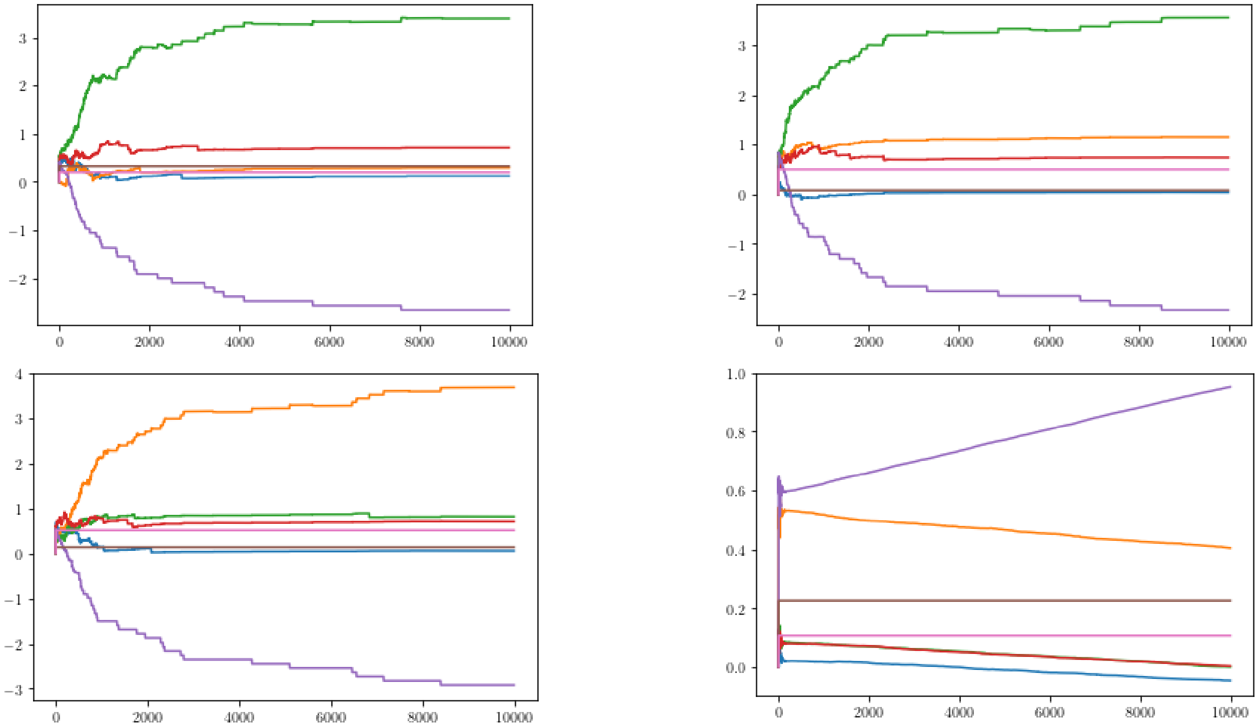
PG, 2 by 2 grid, FG a, SoftMax policy,



```
Number of Omega-automaton states (including the trap
state): 3
100%|██████████| 10000/10000 [02:22<00:00, 70.19it/s]
[[2 1]
 [2 5]]
[[0.95 0.95]
 [0.95 0.95]]
```

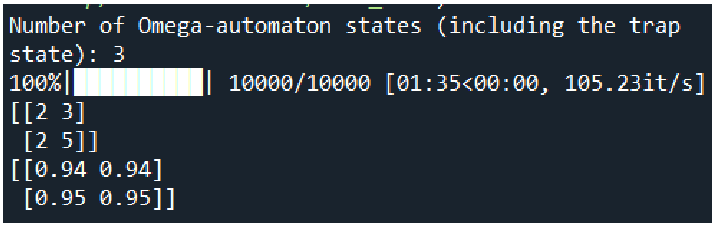


Value function estimation v.s. episodes

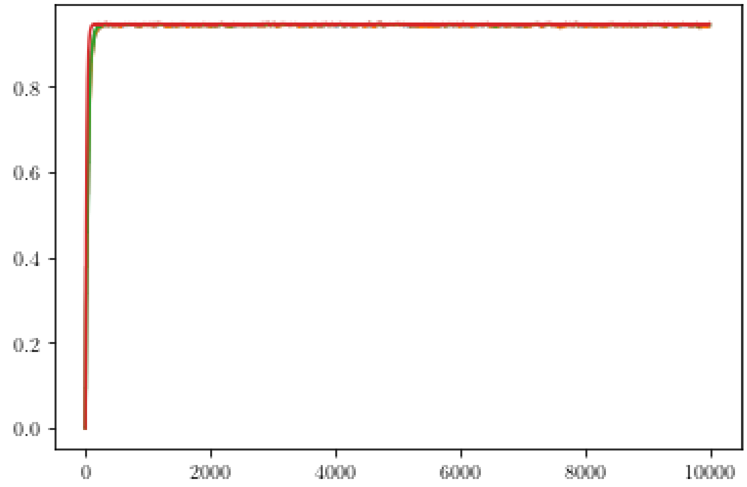


Policy parameter changes v.s. episodes

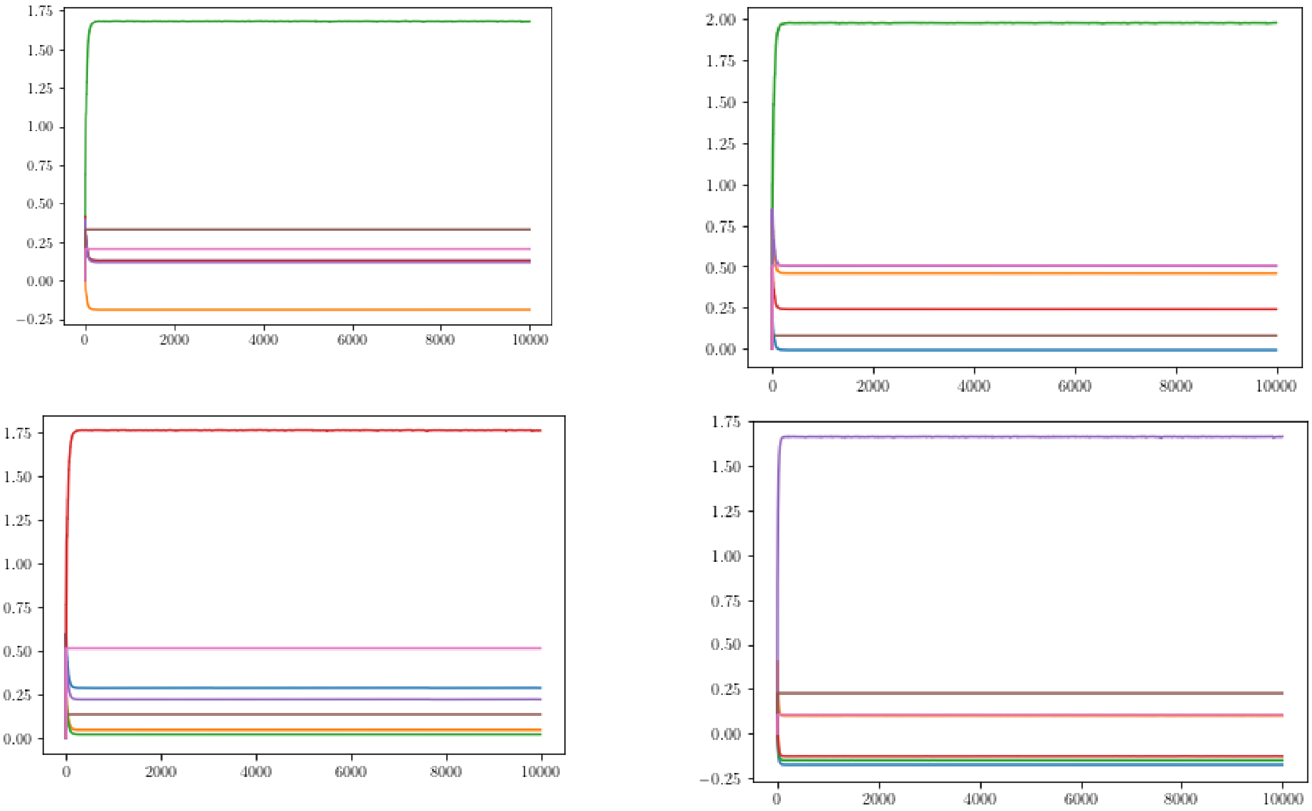
PG, 2 by 2 grid, FG a, MAX(SoftMax) policy,



```
Number of Omega-automaton states (including the trap
state): 3
100%|██████████| 10000/10000 [01:35<00:00, 105.23it/s]
[[2 3]
 [2 5]]
[[0.94 0.94]
 [0.95 0.95]]
```



Value function estimation v.s. episodes



Policy parameter changes v.s. episodes