

In [5]:

```
import numpy as np
from frozen_lake import FrozenLakeEnv

environment = FrozenLakeEnv()
epochs = 1000
if_break = True

def Func(alpha, gamma):
    V = np.zeros(16)

    for epoch in range(epochs):
        state = 0 # stan początkowy kazdego epizodu
        if_break = True
        while if_break:
            random_action = np.random.randint(4)
            tupl = environment.P[state][random_action]
            next_state = tupl[0][1]

            if next_state == 15:
                R = 1
            else:
                R = 0

            V[state] = (V[state] + alpha * (R + gamma*V[next_state] - V[state]))
            state = next_state

            #stan terminalny oznacza koniec epizodu
            if state == 5 or state == 7 or state == 11 or state == 12 or state == 15:
                if_break = False

    return V
```

In [11]:

```
print(Func(0.1,1.0))
print(Func(0.3,0.9))
print(Func(0.5,0.5))
```

```
[0.01680923 0.01564125 0.02732064 0.01640462 0.02149848 0.
0.0639221 0. 0.03465722 0.0493267 0.19954174 0.
0. 0.1885731 0.52879748 0. ]
[0.00106957 0.00089715 0.00140605 0.00107878 0.00062791 0.
0.0044984 0. 0.00200877 0.01661671 0.00975564 0.
0. 0.10698457 0.20813015 0. ]
[3.43285981e-04 5.68175864e-05 8.32619584e-05 3.53909987e-05
1.15083017e-03 0.00000000e+00 1.04765155e-04 0.00000000e+00
4.28157684e-03 1.27098390e-02 5.34812900e-03 0.00000000e+00
0.00000000e+00 1.34975655e-01 3.95619300e-01 0.00000000e+00]
```