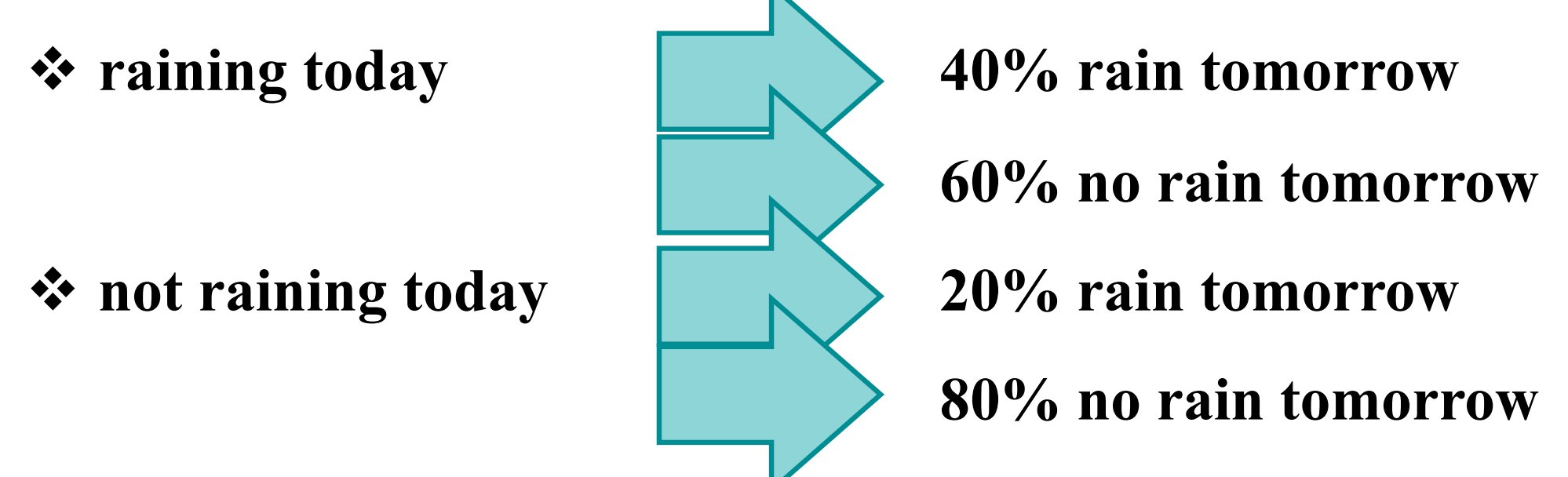
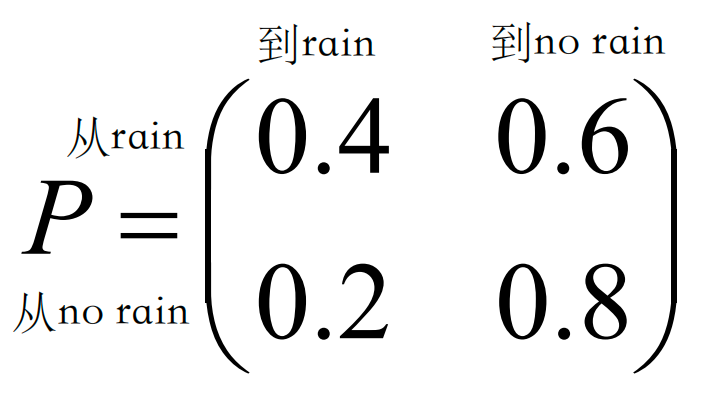
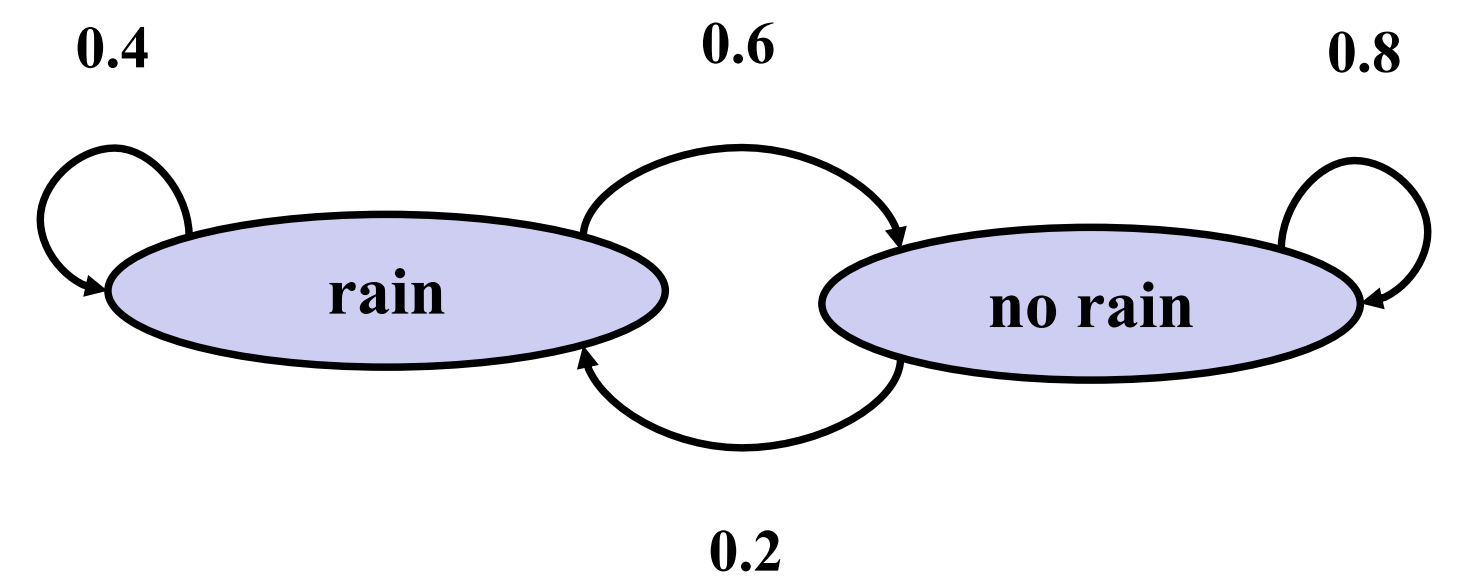
**Markov Process**

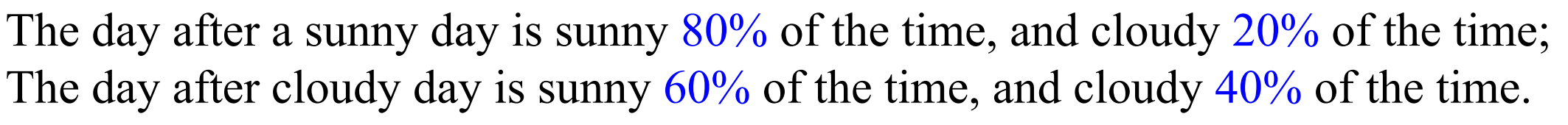
**例题一**



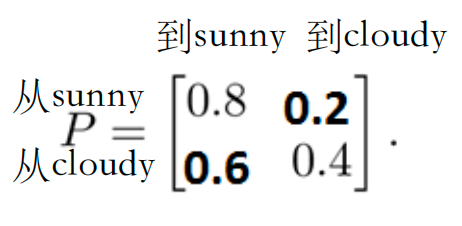
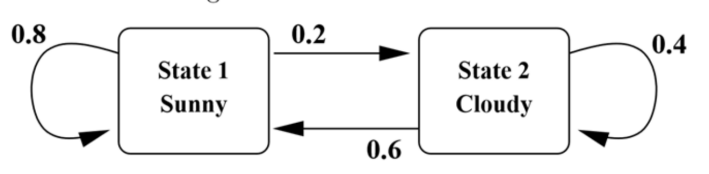
**transition matrix: transition diagram:**



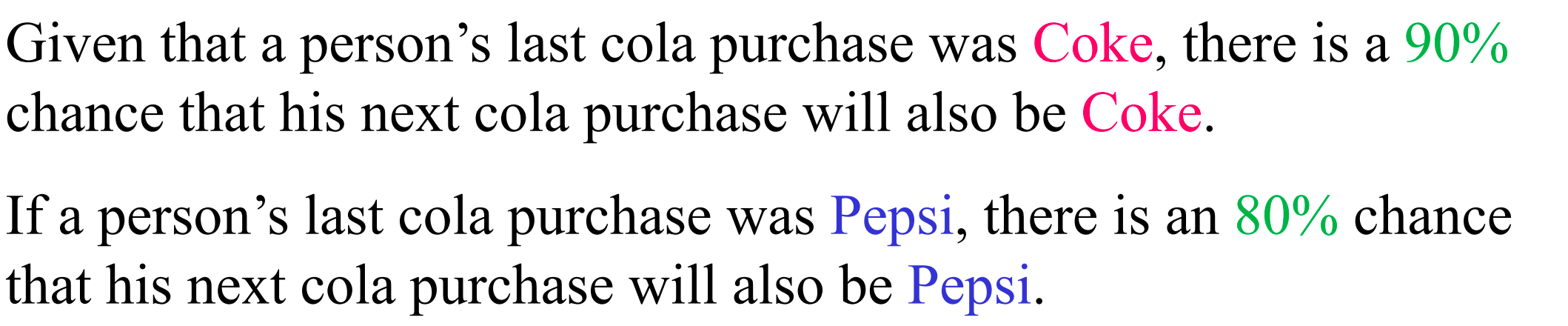
**例题二**

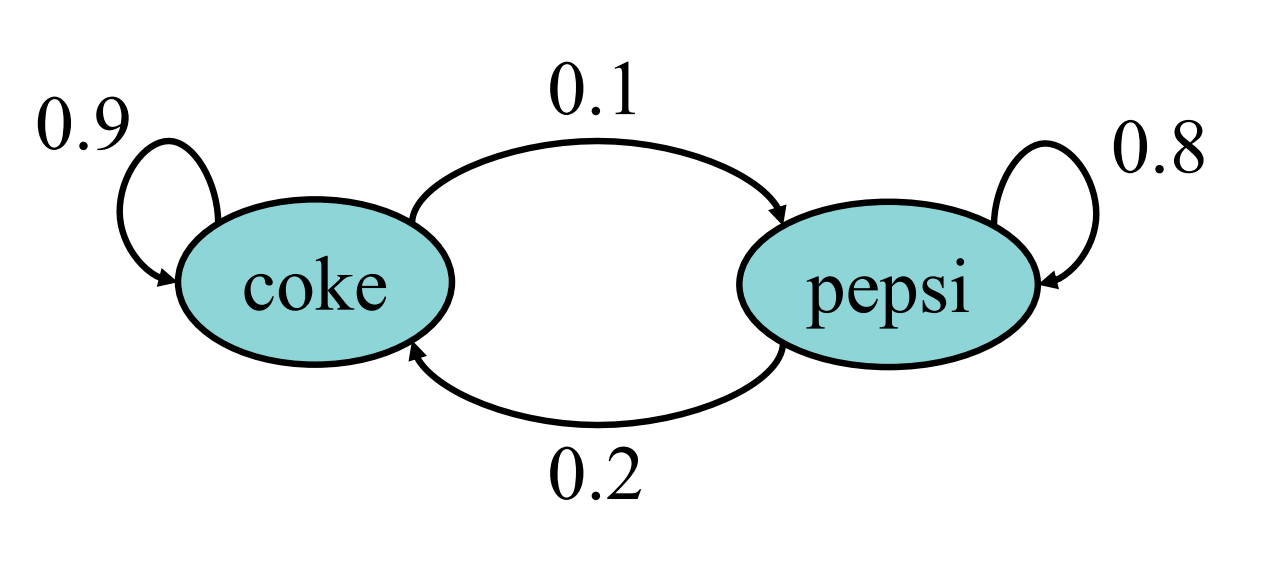


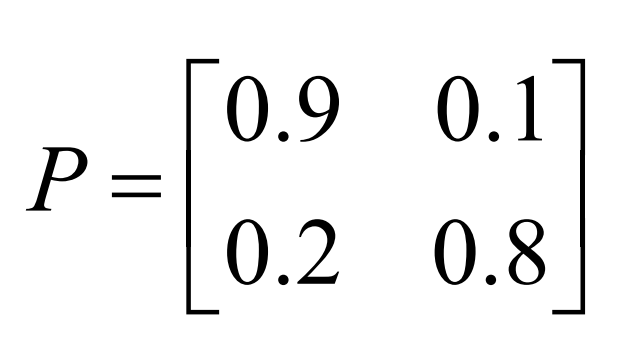
**transition matrix: transition diagram:**

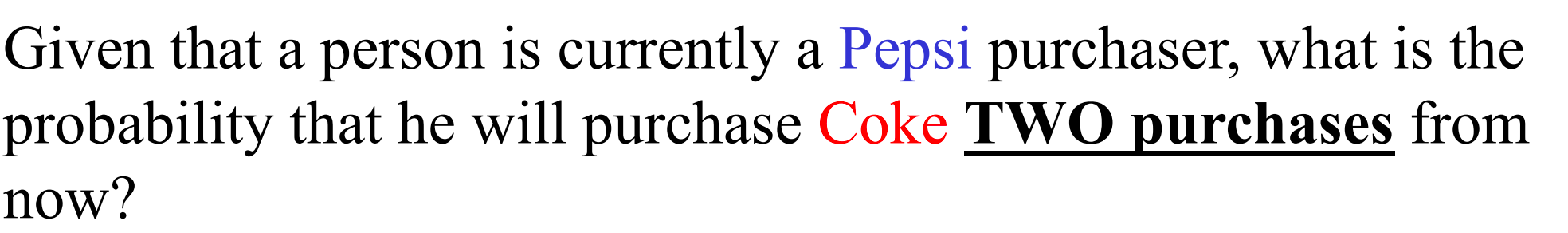


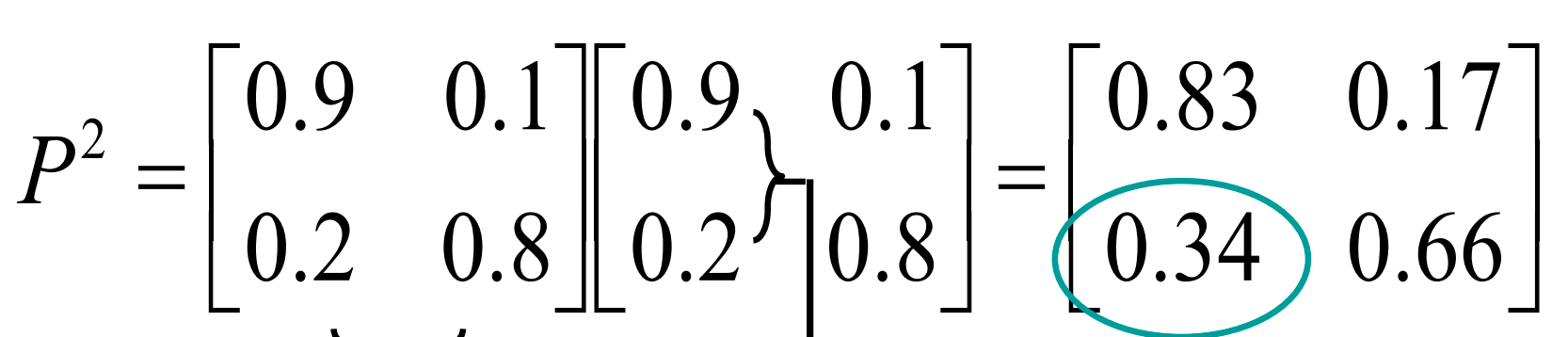
**例题三**

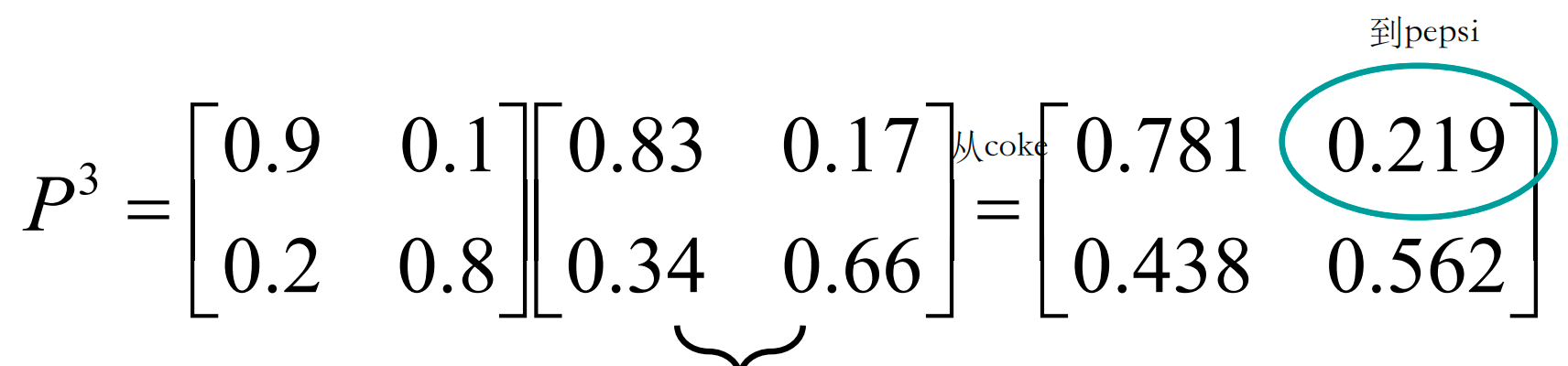
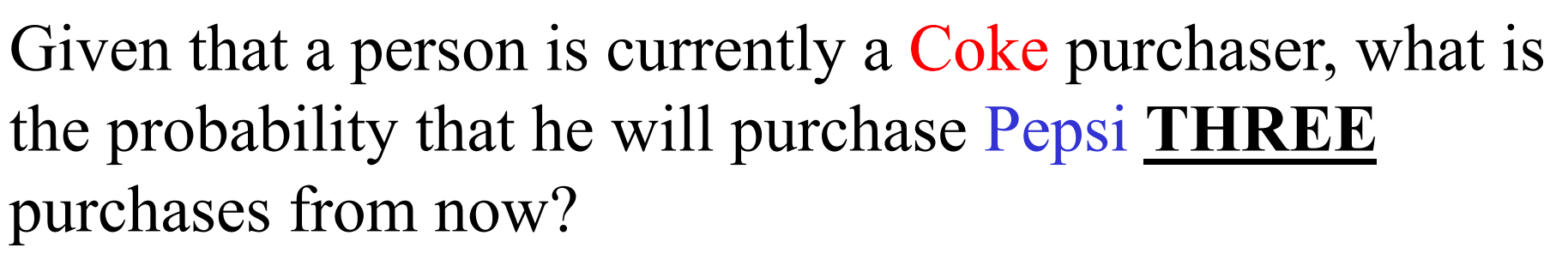


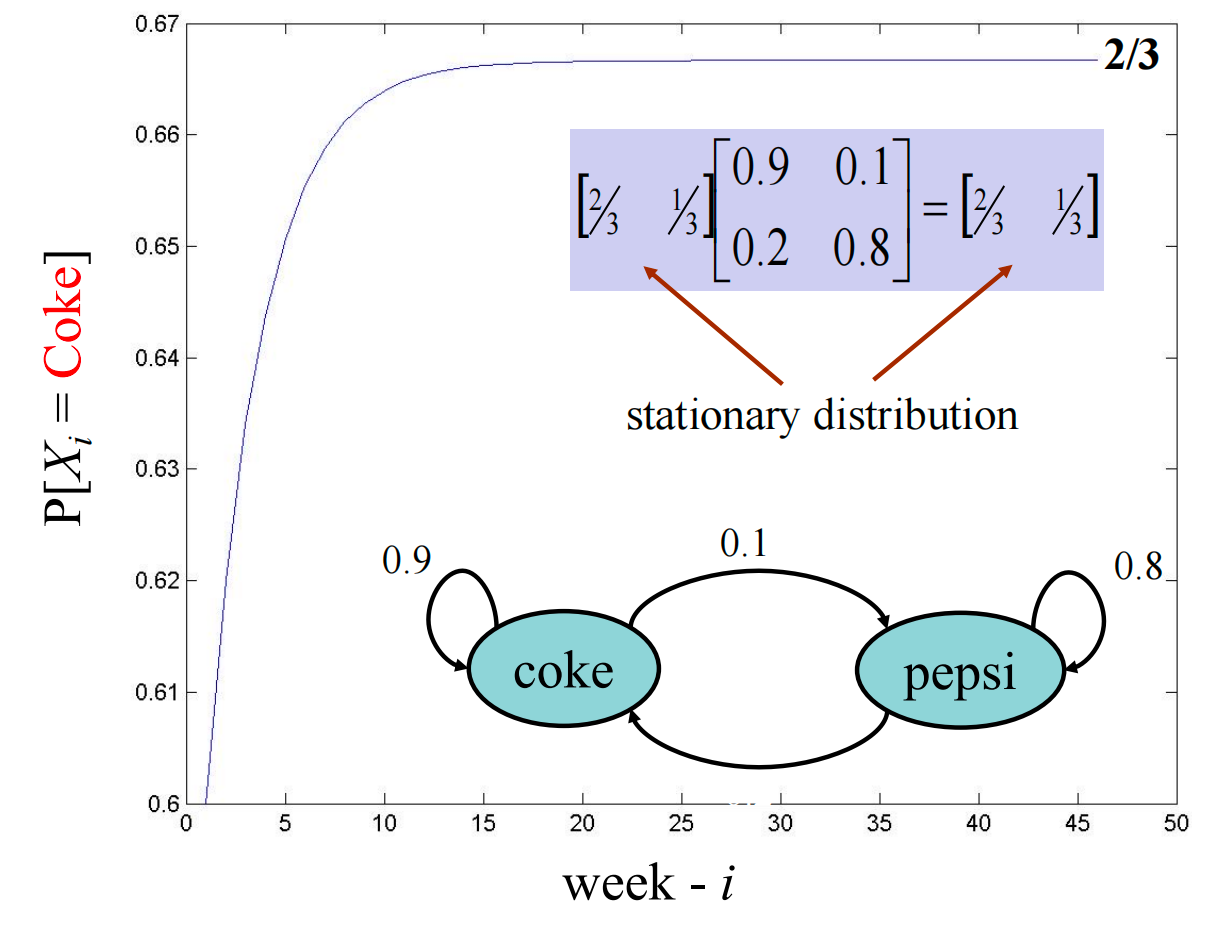
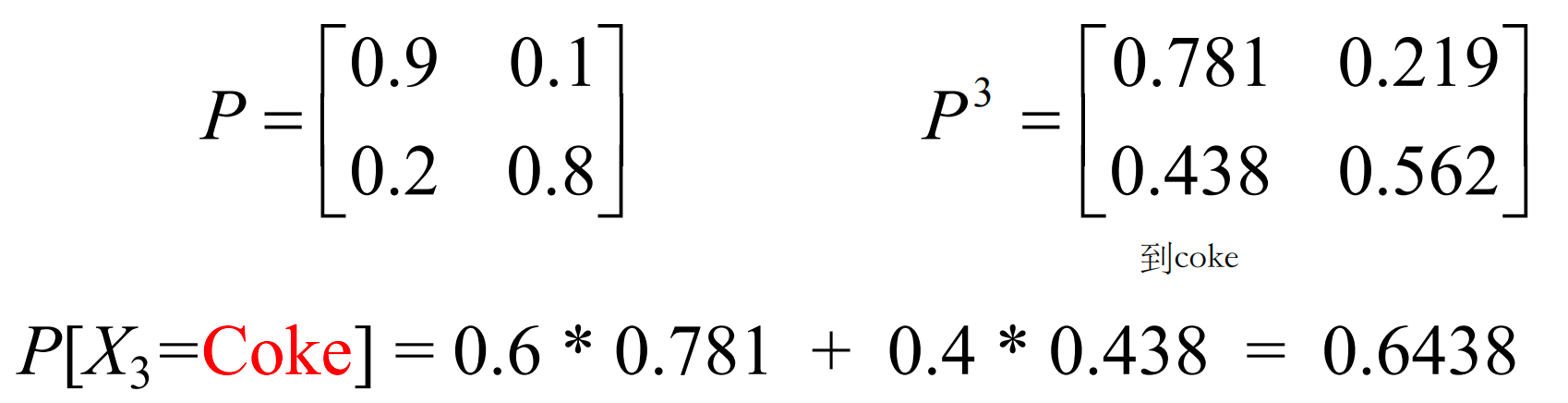
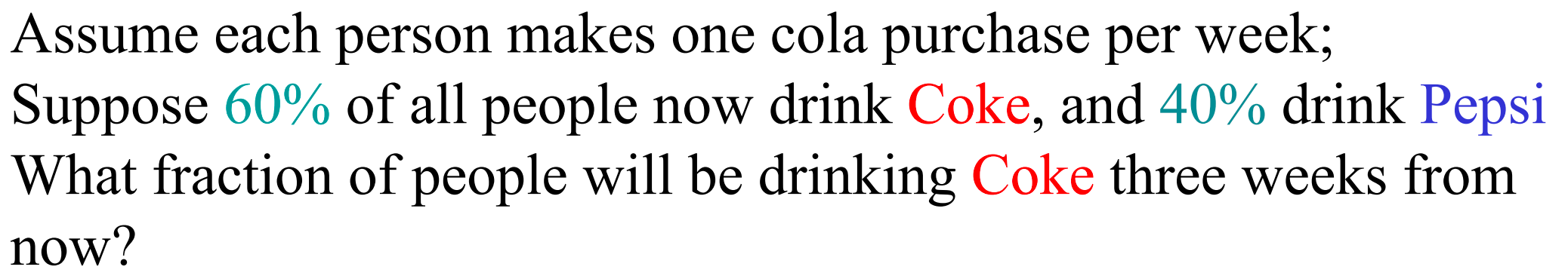












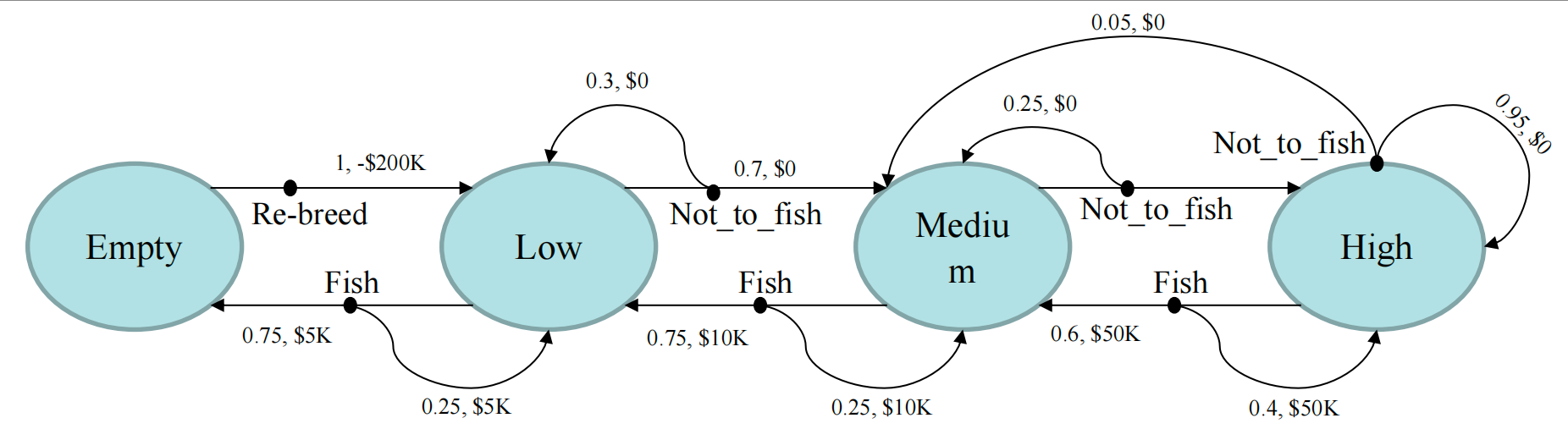


State: 鱼的多少，empty low medium high

Action: 捕猎or不捕

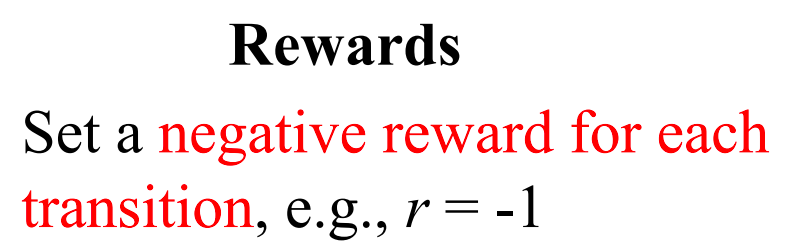
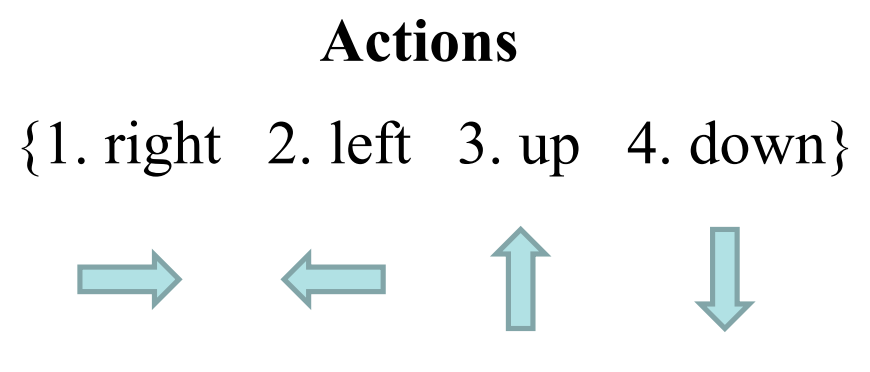
Reward: -200k 5k 10k 50k

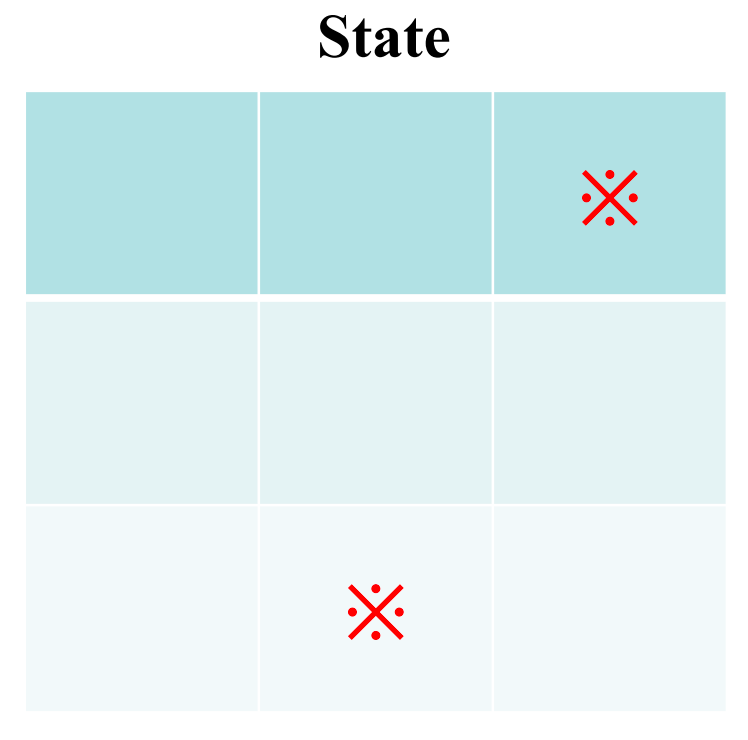
Possibility: 捕猎的话75%概率降级，不捕猎的话30%概率降级



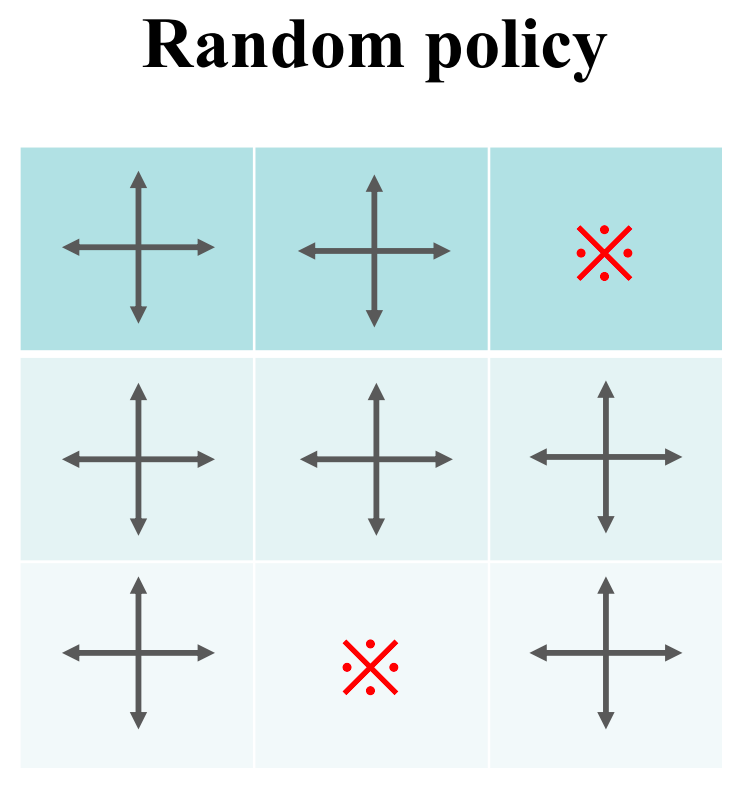
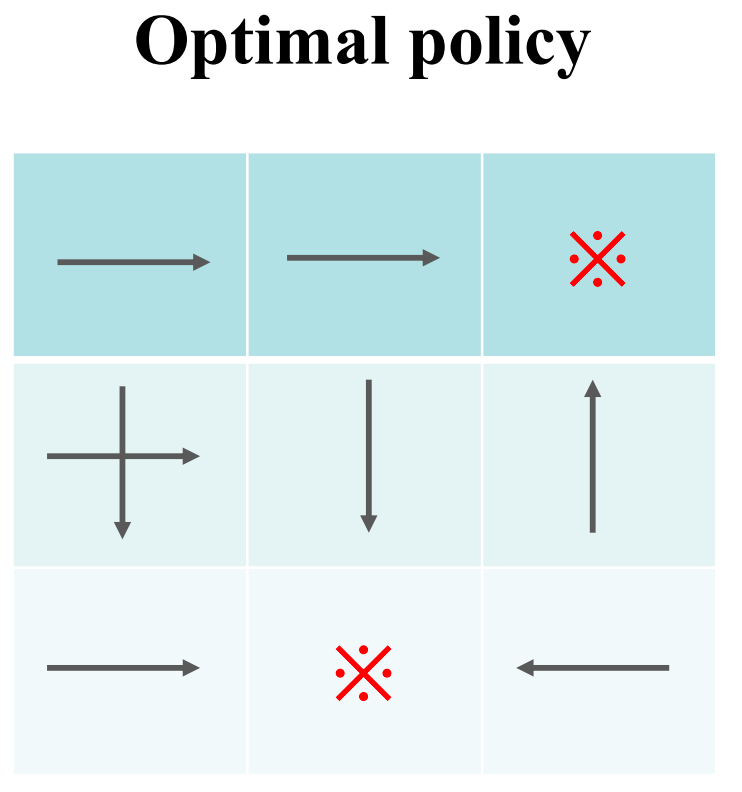
比如：从medium到low捕猎的概率是0.75，回报是10k



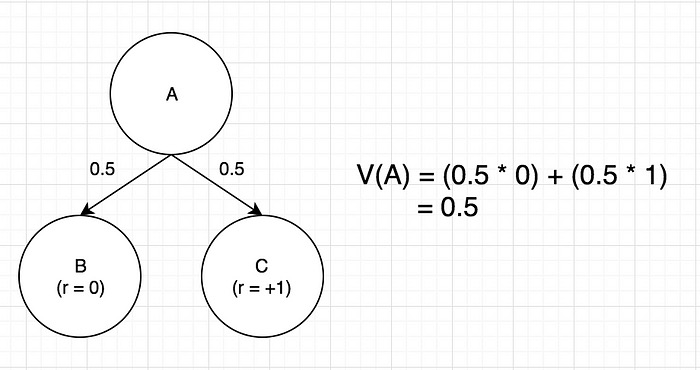




然后会有两种policy，一个是随机的，一个是最优的



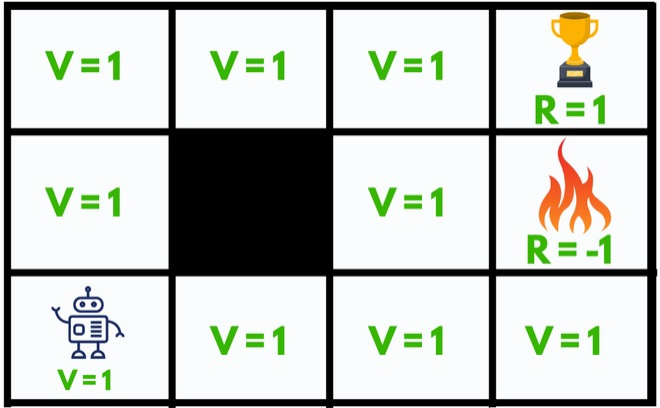
**Value function**



A的value等于所有下一步的reward乘上过去的概率的和

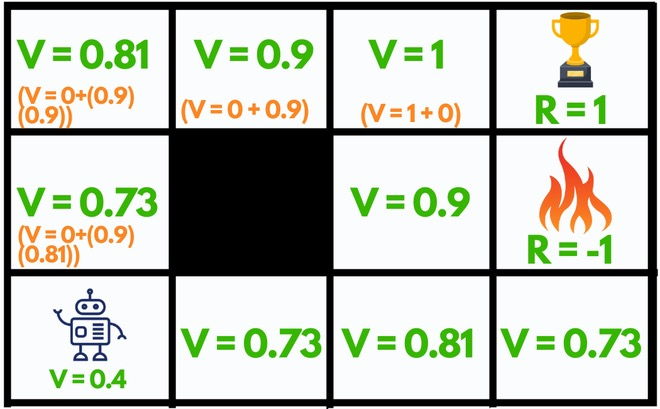
**Bellman equation**

**例题一**



目标：到达trophy state (R=1)，躲开火 (R=-1)

用了Bellman equation之后

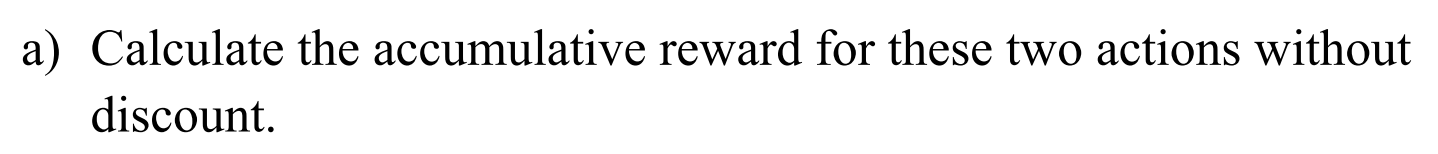
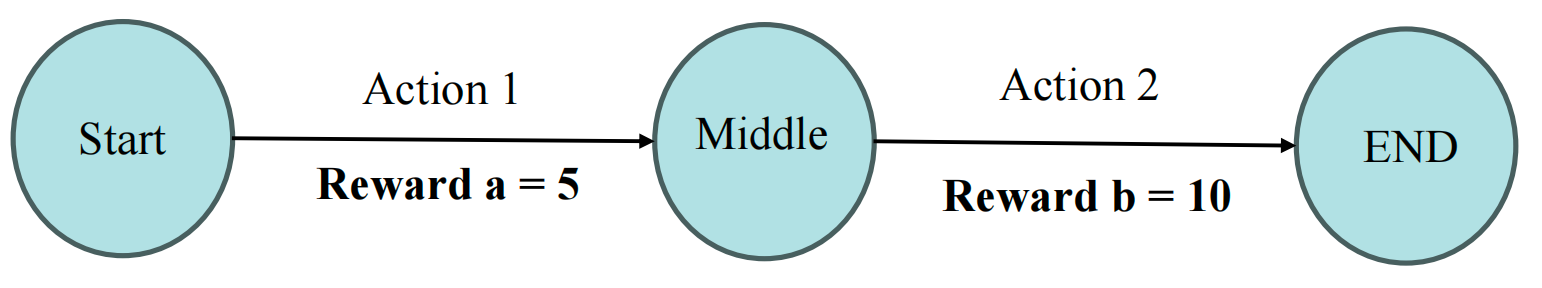


比如火左边那个格子，可以上下右，但它会往上走，因为是maximum value

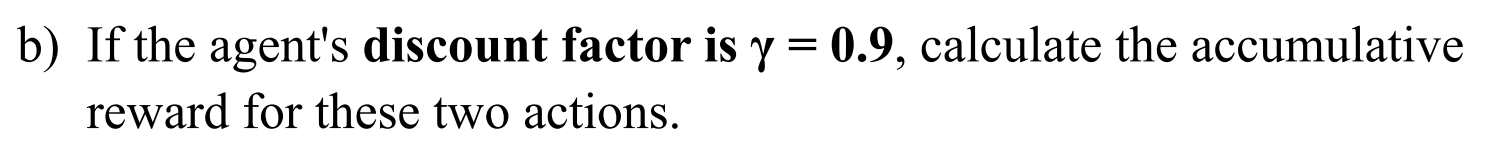
对于机器人在的格子，它会往右或者上走，因为value是一样的

**例题二**

**计算reward**



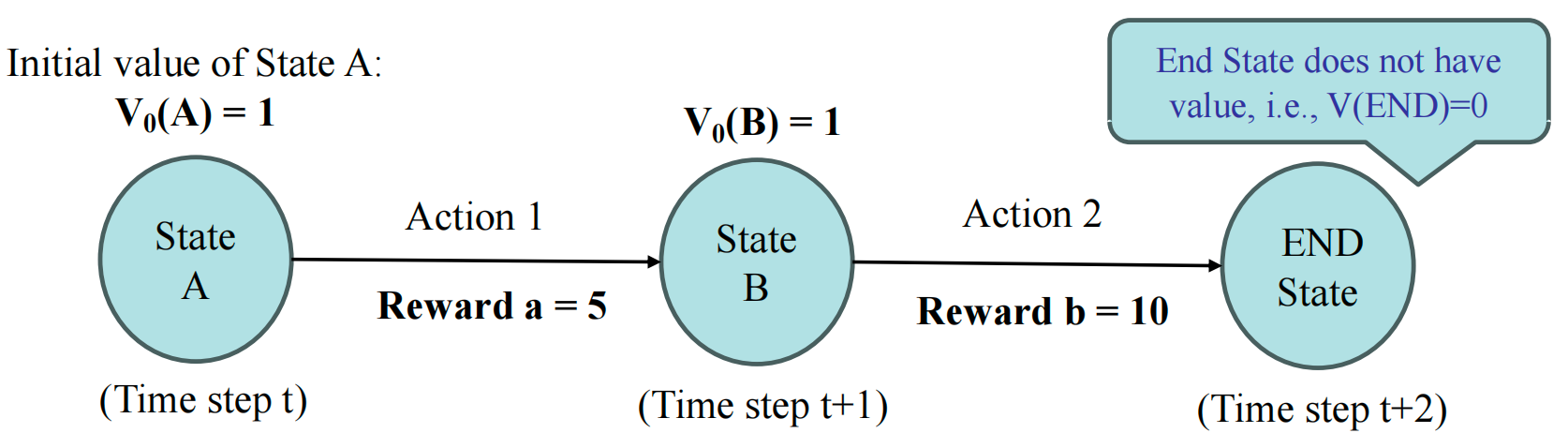
没有discount的话就直接相加呗，5+10=15

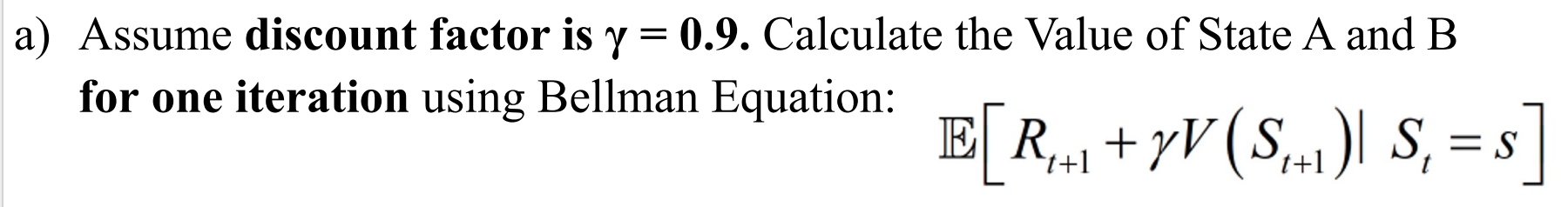


第一个reward加上第二个reward乘以discount factor

5+10\*0.9=14

**计算value**





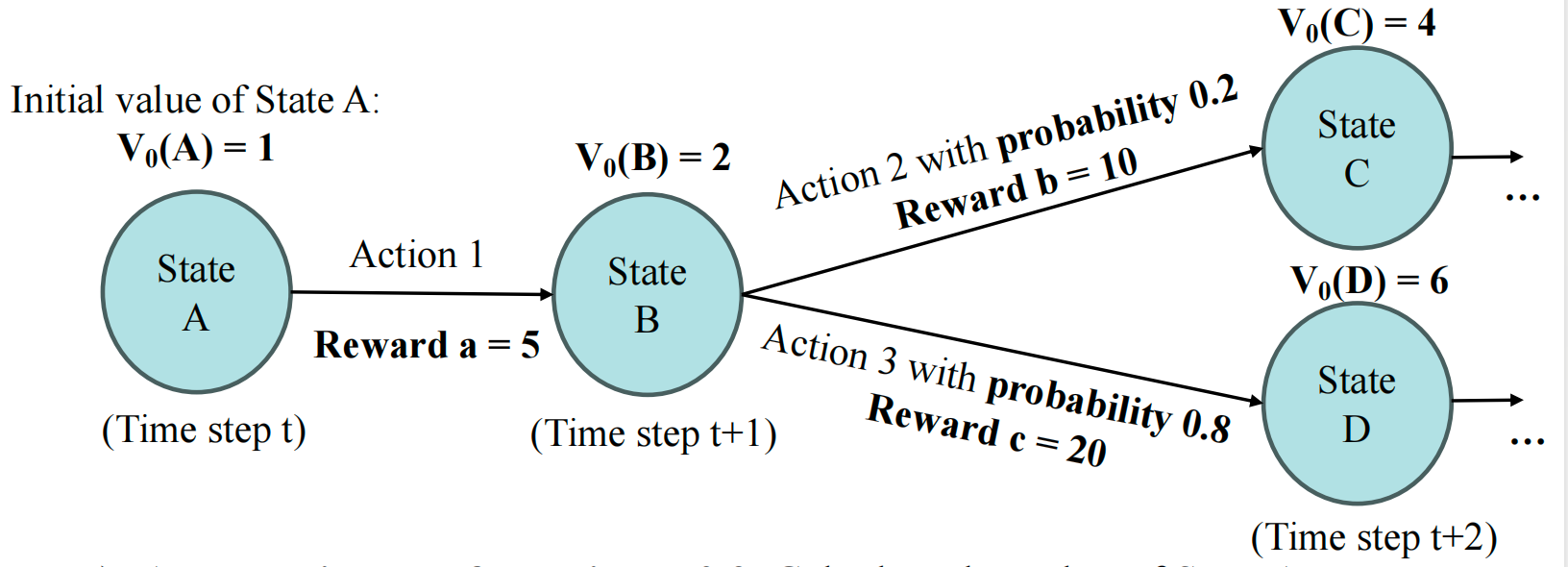
A的value等于到下一步的reward加上下一个状态的初始value乘以discount factor

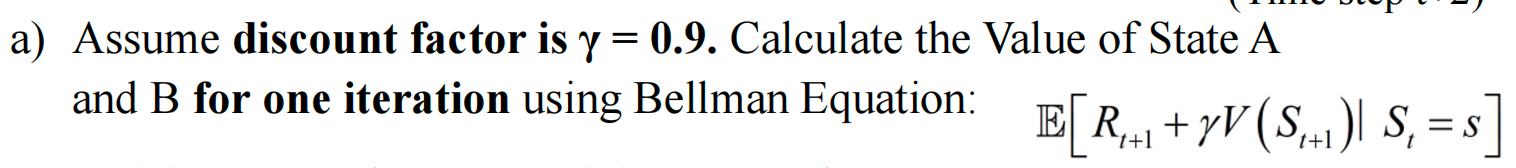
5+1\*0.9=5.9

B的value等于到下一步的reward加上下一个状态的初始value乘以discount factor

10+0\*0.9=10

**带上probability算value**





A的value等于到下一步的reward加上下一个状态的初始value乘以discount factor

5+2\*0.9=6.8 （和上面那道题方法一样）

B的value等于（到下一步的reward加上下一个状态的初始value乘以discount factor）再乘上probability然后再相加

（10+4\*0.9）\*0.2 + （20+6\*0.9）\*0.8 =2.72+20.32=23.04

**总结**

**如果算的是reward， 那么就把reward 乘上discount factor**

**如果算的是 value， 那么就把 value 乘上discount factor**

**Q-learning**



Goal is to **eat** the big pile of cheese and **avoid** the poison.

The learning rate **α** is 0.1, the discount factor ***γ*** is 0.99

Reward: **+0** when going to a state with **no cheese or taking more than five**

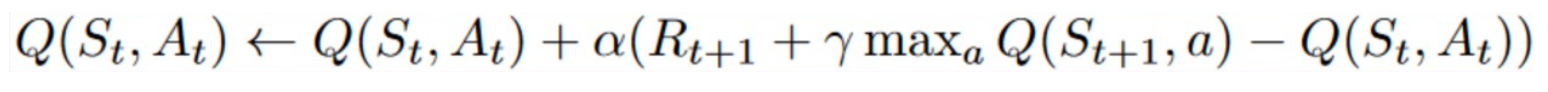
**steps**; **+1** when going to a state with **a small cheese**; **+10** when going to the

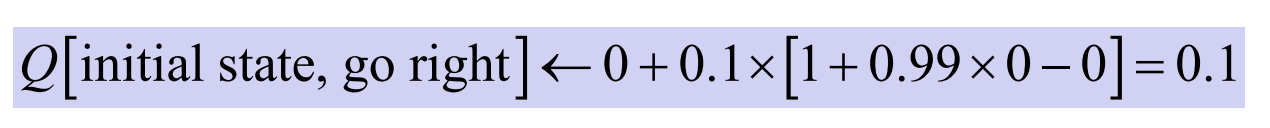
state with **the big pile of cheese**; **-10** when going to the state with **the poison**

**初始化Q-table**



随机选到了向右走，





0.1是learning rate

0.99是discount factor

更新Q-table

