16-782
Planning & Decision-making in Robotics

Planning under Uncertainty: Partially Observable Markov Decision Processes (POMDP)

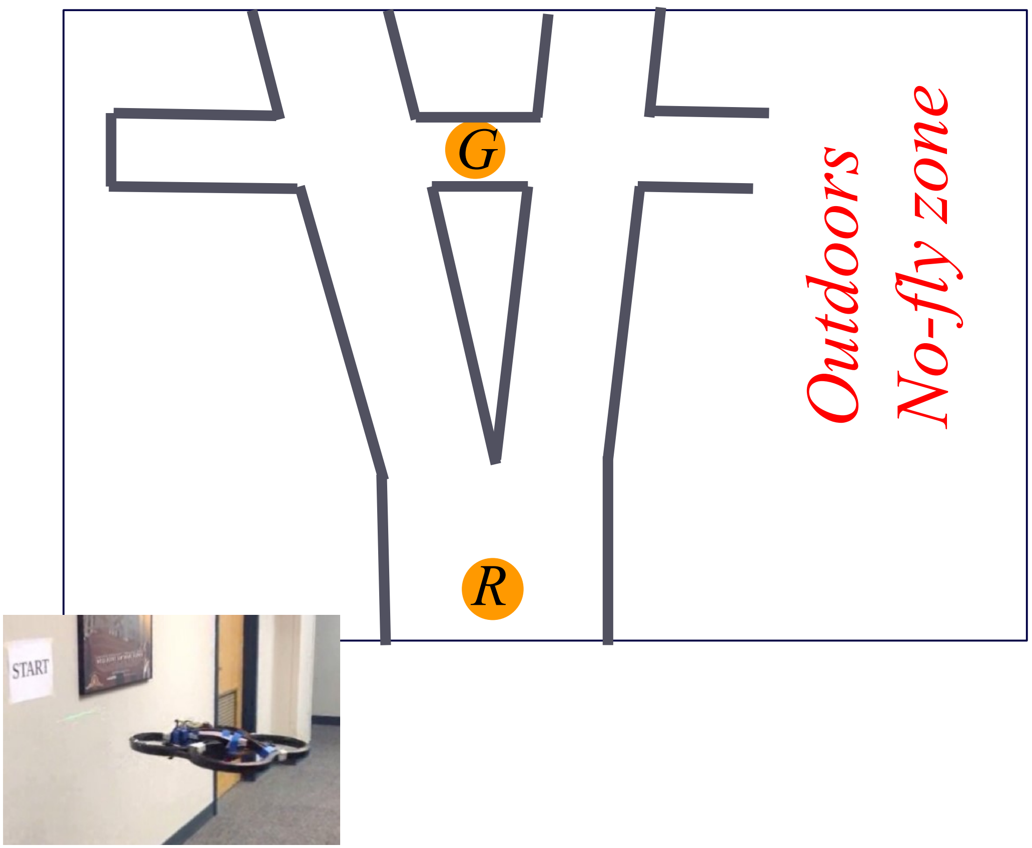
Maxim Likhachev

Robotics Institute

Carnegie Mellon University

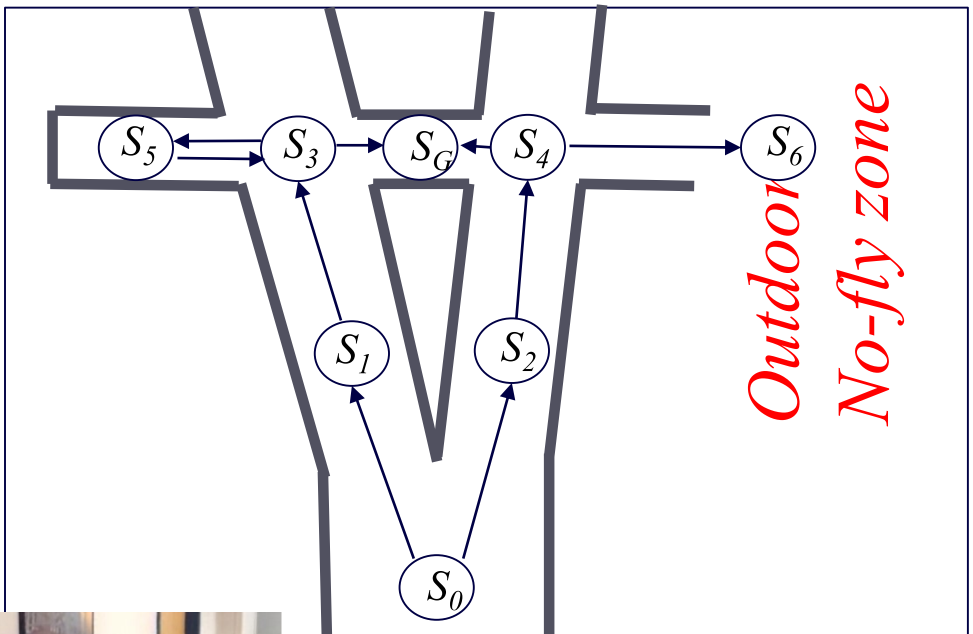
Graph vs. MDP vs. POMDP

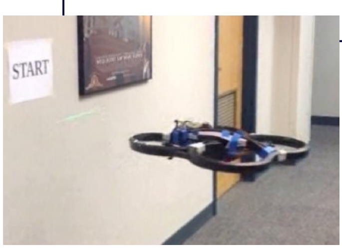
- Consider a path planning example



Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume perfect action execution and full knowledge of the state (i.e., perfect localization)

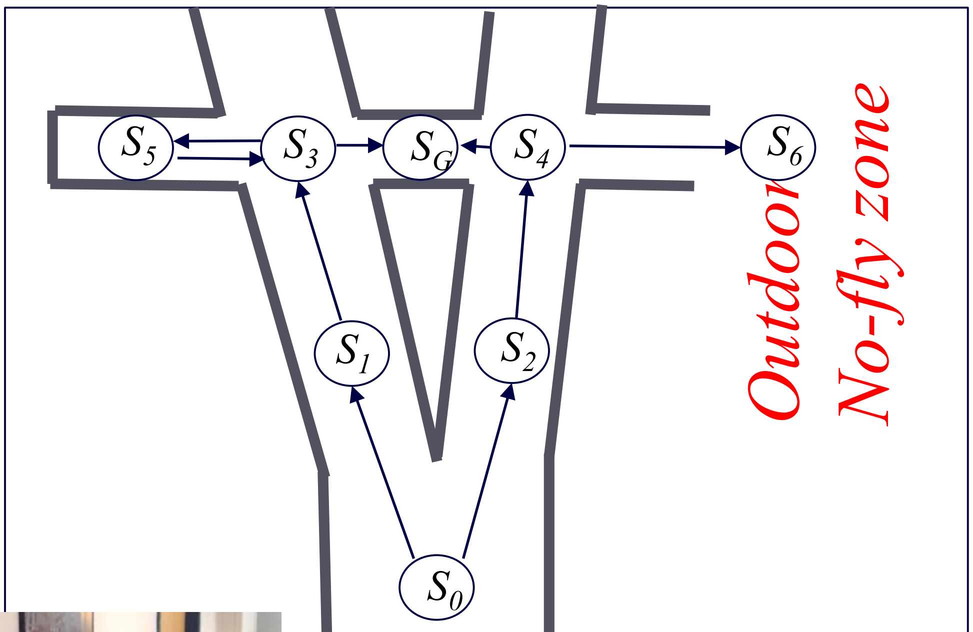


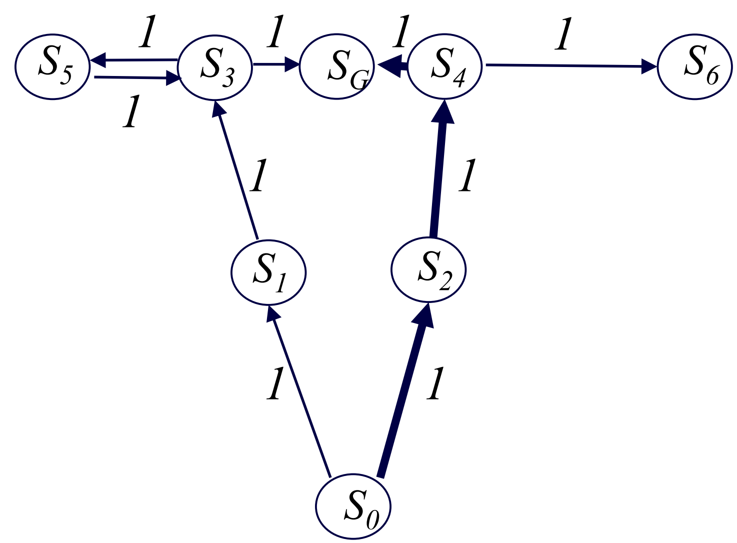


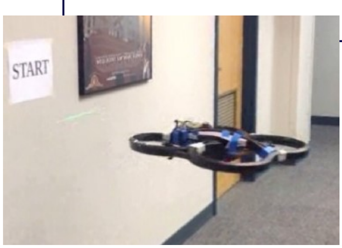
Graph

Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume perfect action execution and full knowledge of the state (i.e., perfect localization)







Graph:

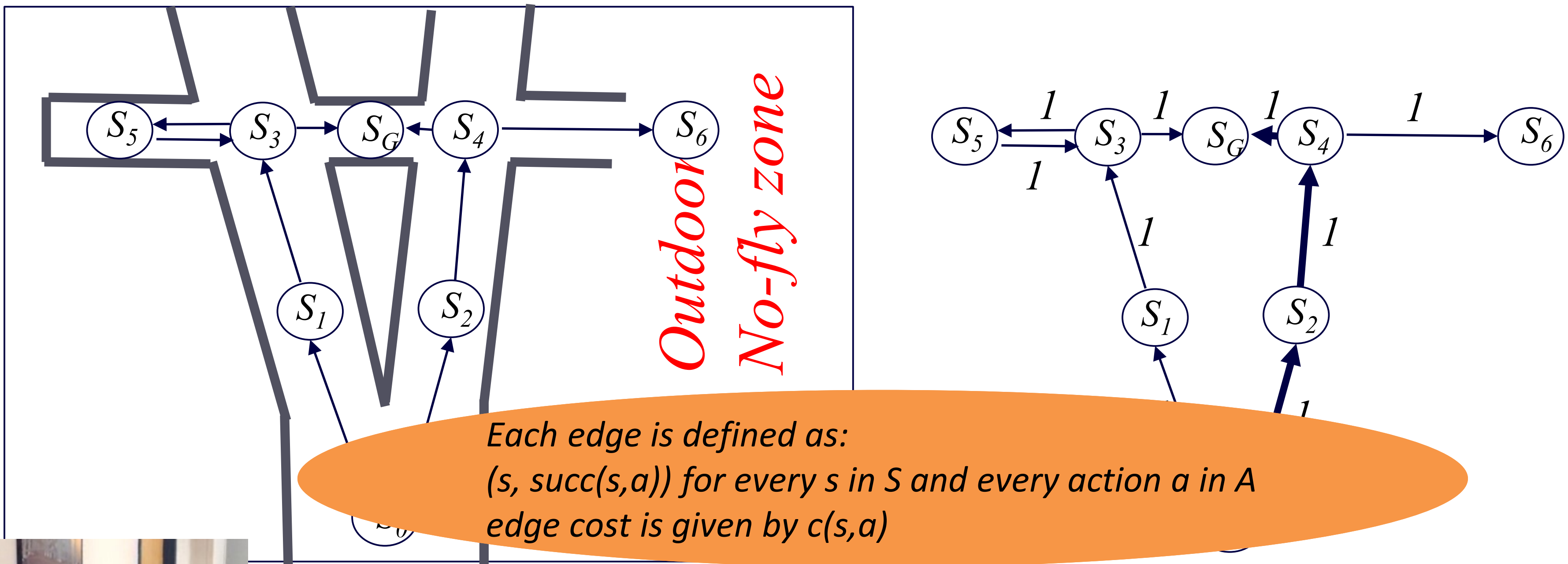
Defined as {S, A, C},

where S – set of states, A – set of actions, C – costs of all (s,a) pairs.

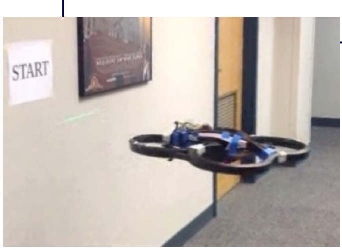
Graph vs. MDP vs. POMDP

- Consider a path planning example

- Assume perfect action execution and full knowledge of the state (i.e., perfect localization)



Each edge is defined as:
(s, succ(s,a)) for every s in S and every action a in A
edge cost is given by c(s,a)

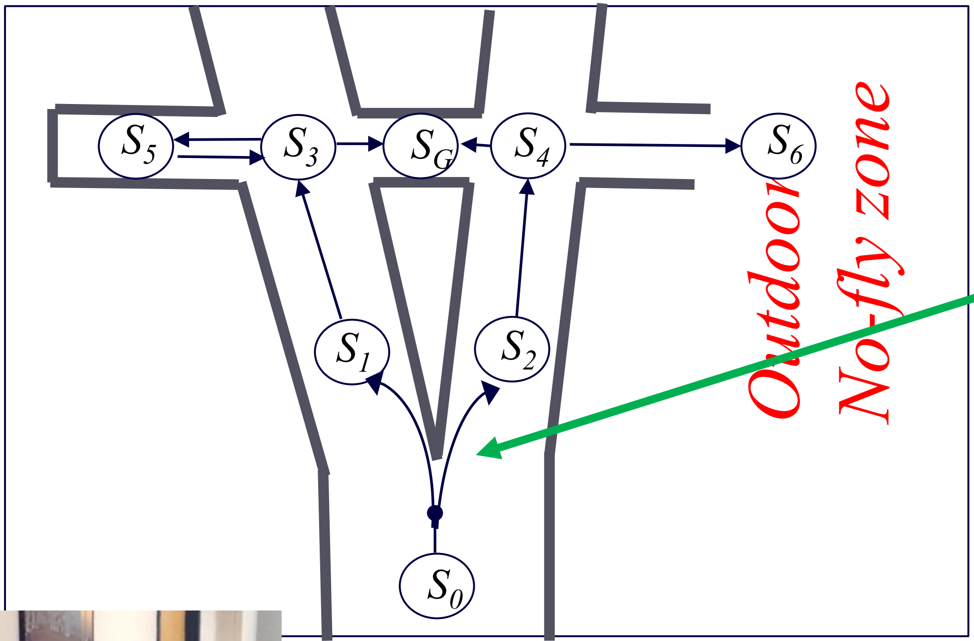


Graph:

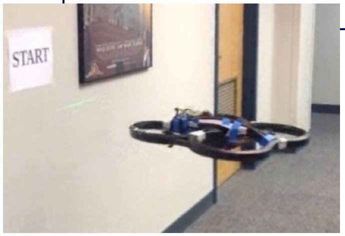
Defined as {S, A, C},
where S – set of states, A – set of actions, C – costs of all (s,a) pairs.

Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume **imperfect action execution** and full knowledge of the state (i.e., perfect localization)



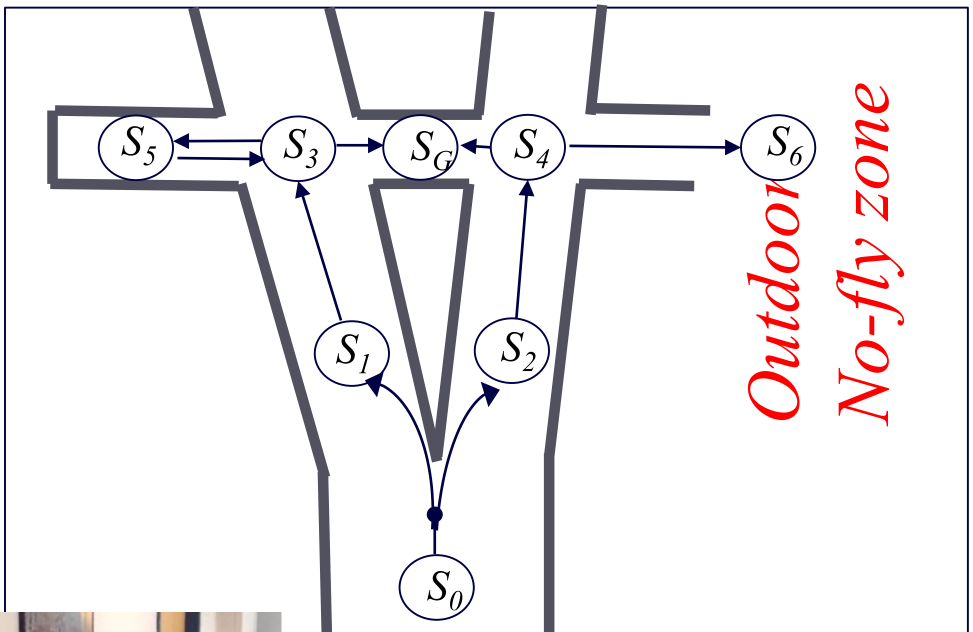
Let's assume
50% chance of ending up on the left and
50% ending up on the right

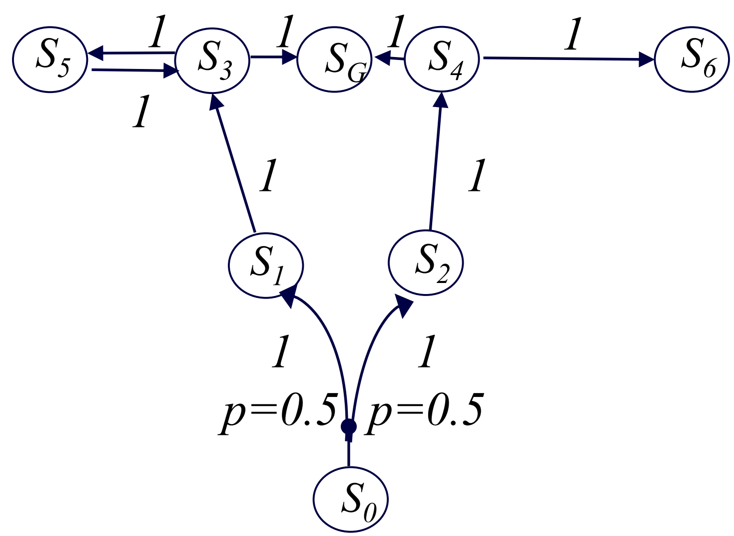


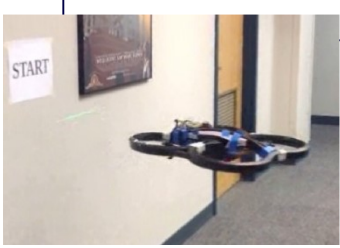
MDP:

Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume **imperfect action execution** and full knowledge of the state (i.e., perfect localization)







MDP:

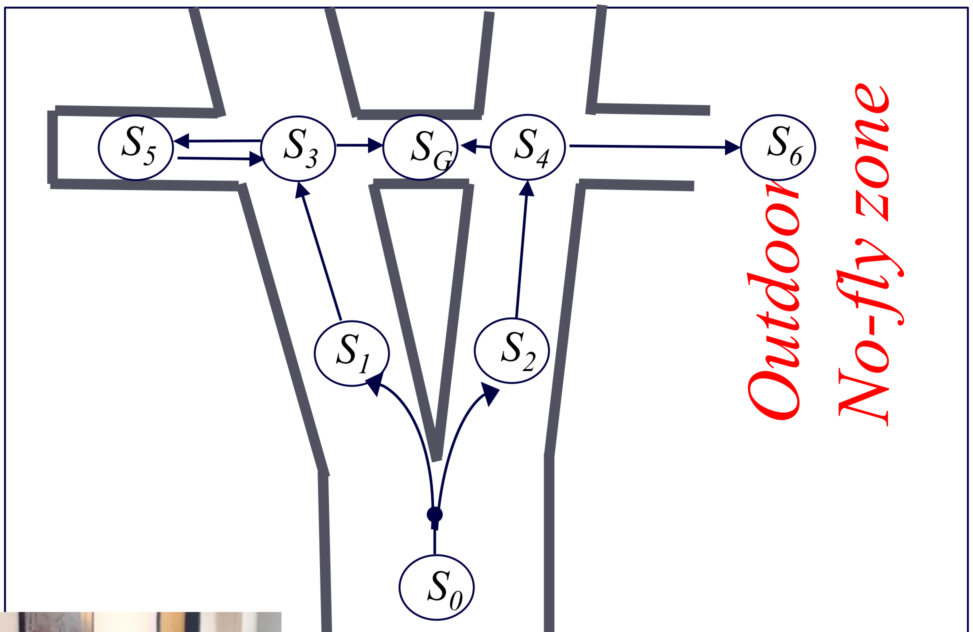
Defined as {S, A, T, C}, where S – set of states, A – set of actions, ***T(s,a,s') - Prob(s'|s, a)****, C – costs of all (s,a) pairs*

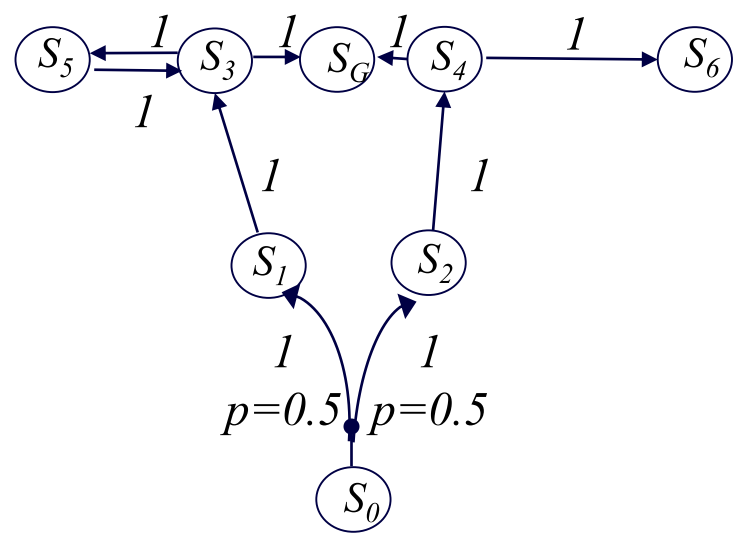
Graph vs. MDP vs. POMDP

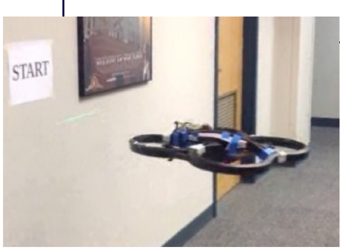
- Consider a path pl

What is an optimal policy here?

- Assume **imperfect action execution** and full knowledge of the state (i.e., perfect localization)







MDP:

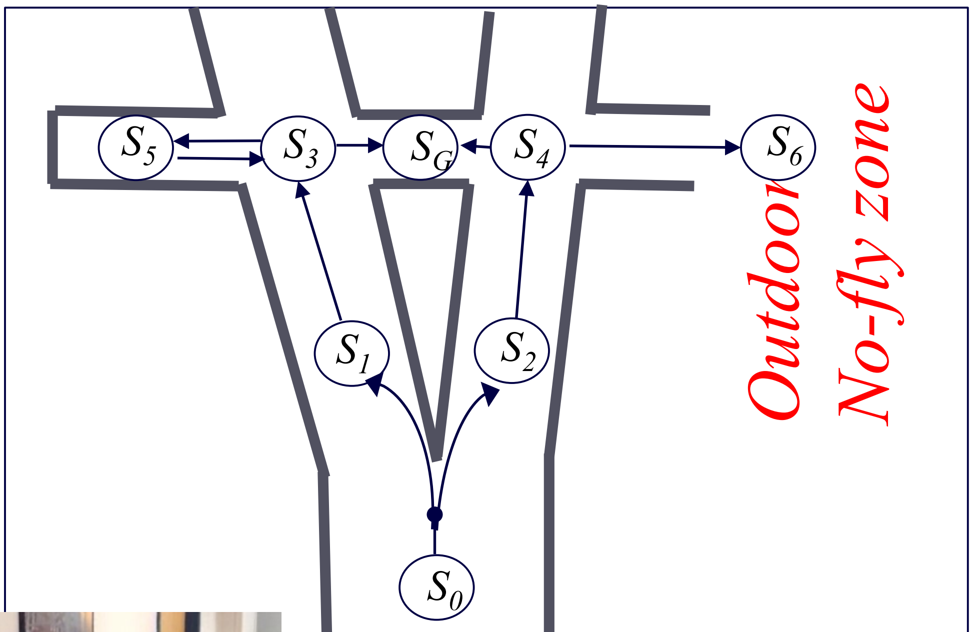
Defined as {S, A, T, C}, where S – set of states, A – set of actions, T(s,a,s') - Prob(s'|s, a), C – costs of all (s,a) pairs

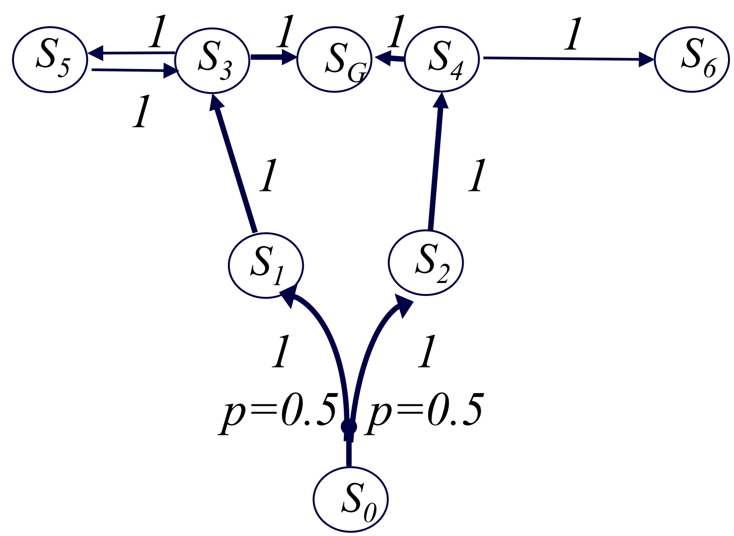
Graph vs. MDP vs. POMDP

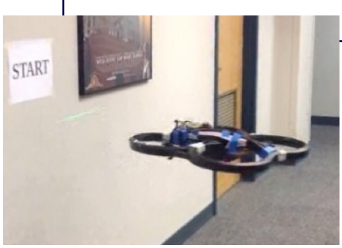
- Consider a path pl

What is an optimal policy here?

- Assume **imperfect action execution** and full knowledge of the state (i.e., perfect localization)





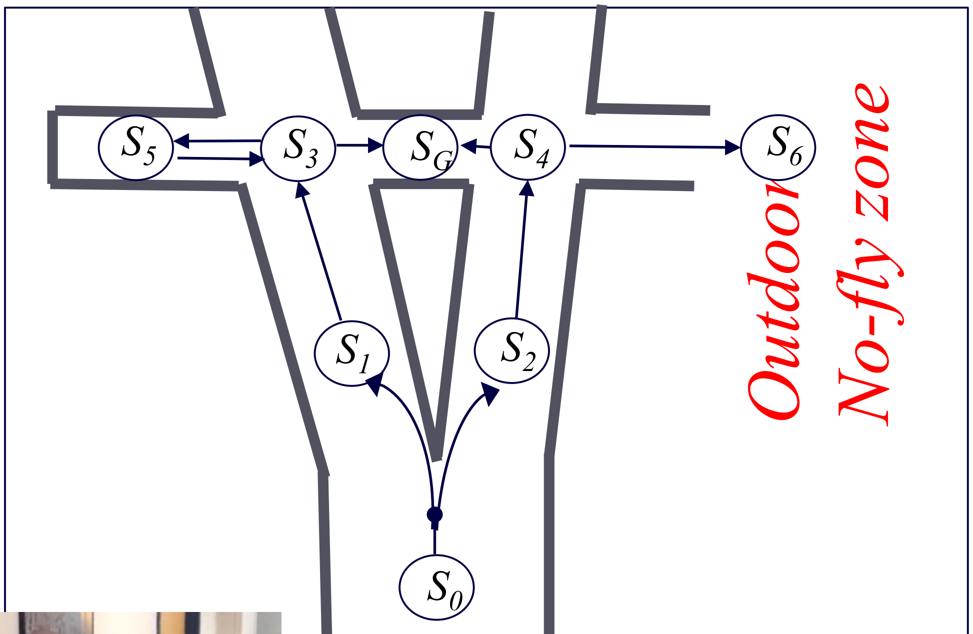


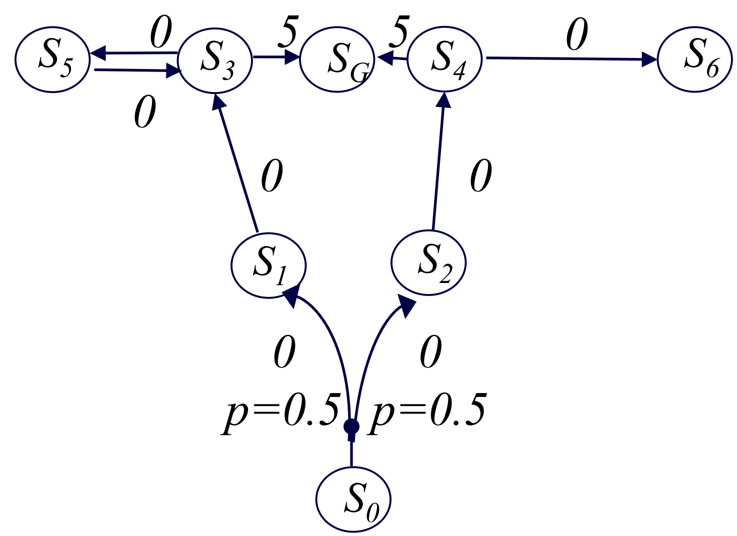
MDP:

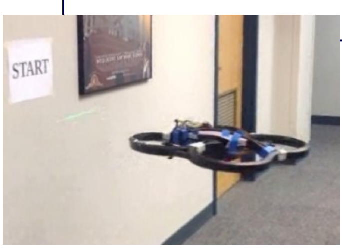
Defined as {S, A, T, C}, where S – set of states, A – set of actions, T(s,a,s') - Prob(s'|s, a), C – costs of all (s,a) pairs

Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume **imperfect action execution** and full knowledge of the state (i.e., perfect localization)





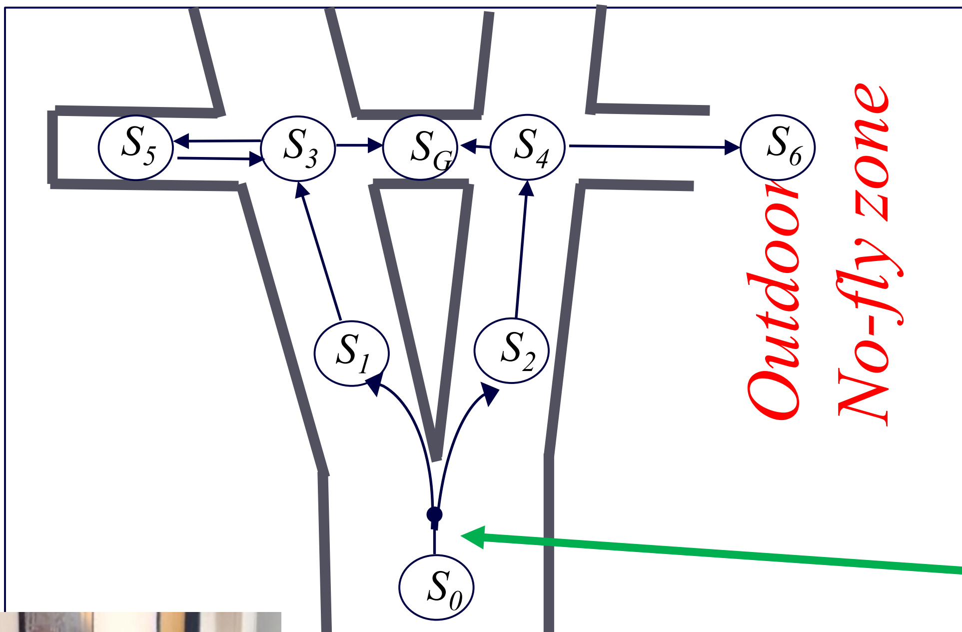


MDP (rewards version):

Defined as {S, A, T, R}, where S – set of states, A – set of actions, T(s,a,s') - Prob(s'|s, a), R – rewards for all (s,a) pairs

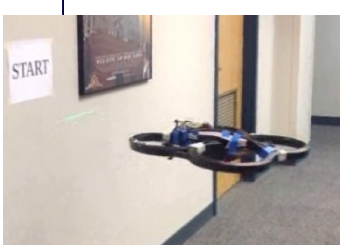
Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume imperfect action execution and **partial observability of the state** (i.e., **imperfect localization**)



Let's assume
UAV initially knows it is at S_0
During execution: it can only sense adjacent obstacles and being at goal

After taking this action, UAV doesn't know whether it is at state S_1 or S_2



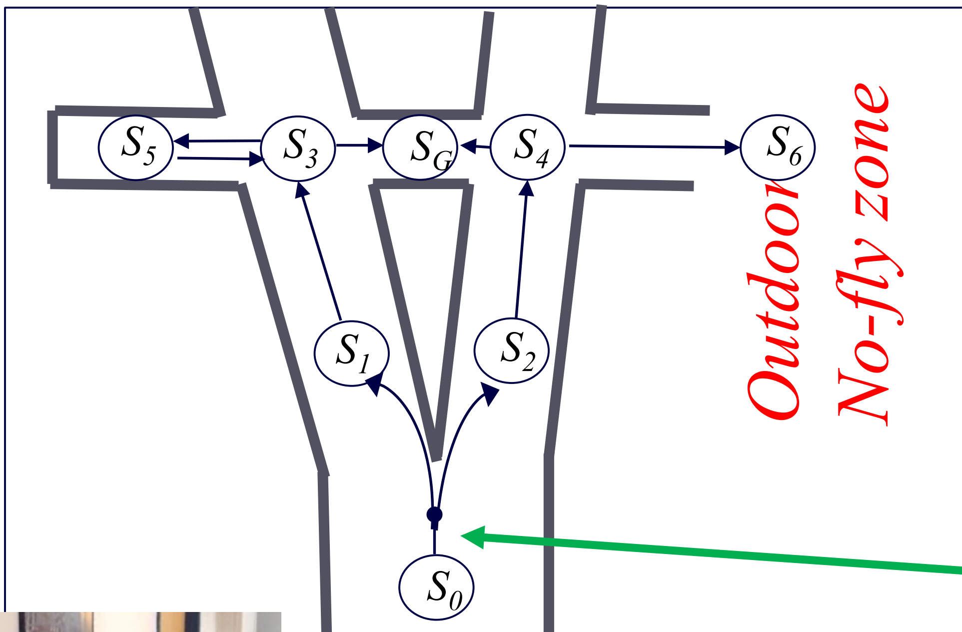
POMDP:

Graph vs. MDP vs. POMDP

- Consider a path pl

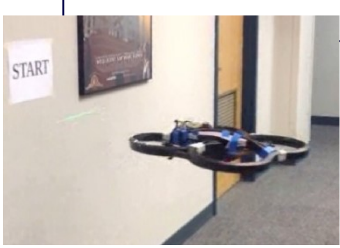
What is an optimal policy here?

- Assume imperfect action execution and **partial observability of the state** (i.e., **imperfect localization**)



Let's assume
UAV initially knows it is at S_0
During execution: it can only sense adjacent obstacles and being at goal

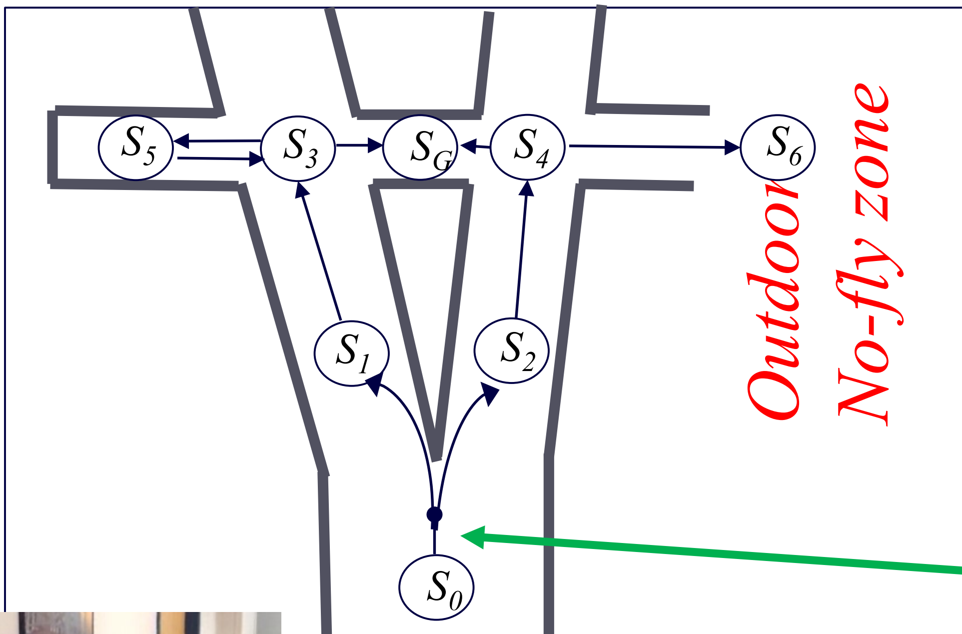
After taking this action, UAV doesn't know whether it is at state S_1 or S_2



POMDP:

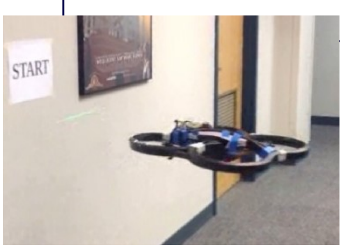
Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume imperfect action execution and **partial observability of the state** (i.e., **imperfect localization**)



Let's assume
UAV initially knows it is at S_0
During execution: it can only sense adjacent obstacles and being at goal

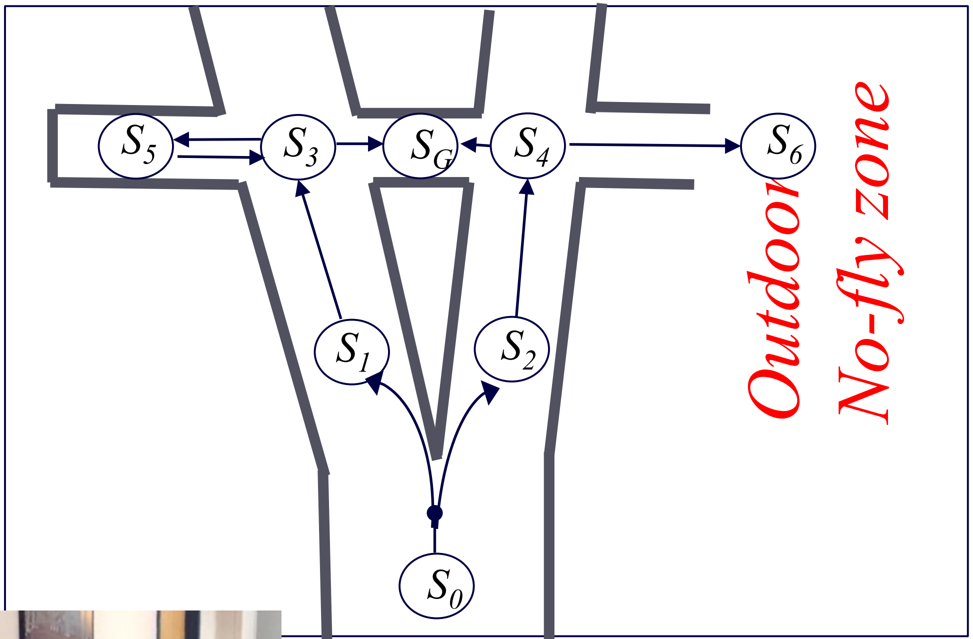
After taking this action, UAV doesn't know whether it is at state S_1 or S_2

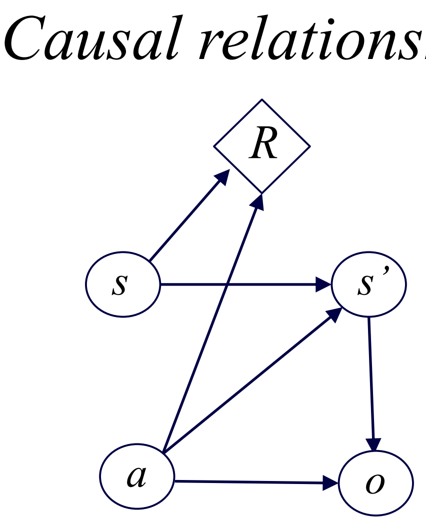


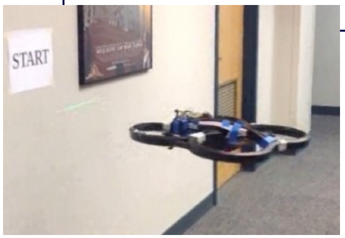
POMDP: *{S, A, T, R, Ω, O}, where S, A, T, R (or C) – same as in MDP, Ω – set of all possible observation vectors o,* ***O(s',a,o) – Prob(o|s',a) probability of seeing o after executing action a and ending up at state s'***

Graph vs. MDP vs. POMDP

- Consider a path planning example
- Assume imperfect action execution and **partial observability of the state** (i.e., **imperfect localization**)







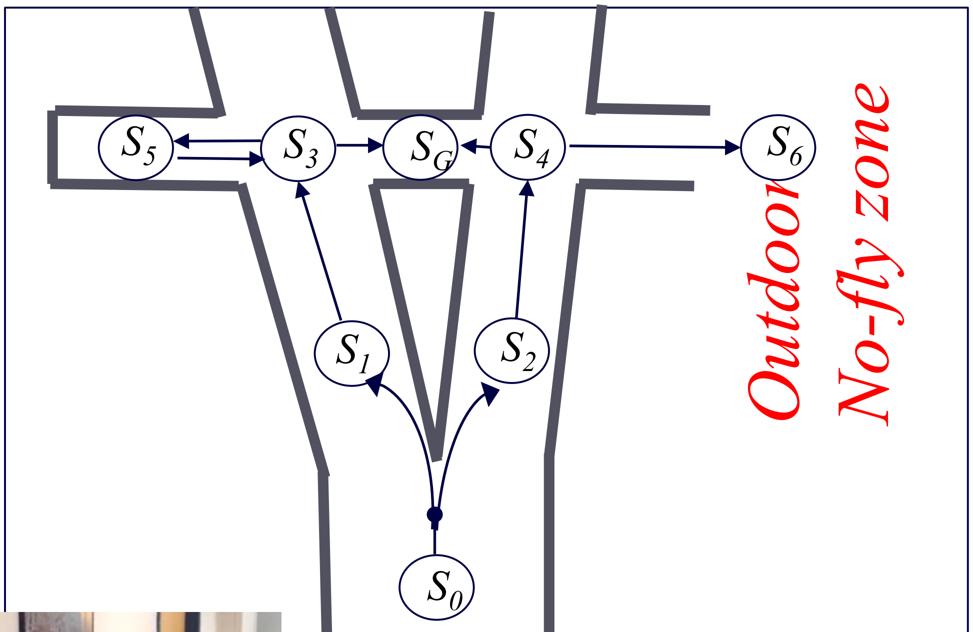
POMDP: *{S, A, T, R, Ω, O}, where S, A, T, R (or C) – same as in MDP, Ω – set of all possible observation vectors o,* ***O(s',a,o) – Prob(o|s',a) probability of seeing o after executing action a and ending up at state s'***

Graph vs. MDP vs. POMDP

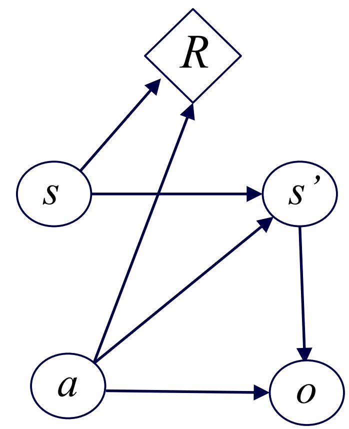
Example of POMDP problems
where the robot knows its own pose perfectly
(perfect localization)?

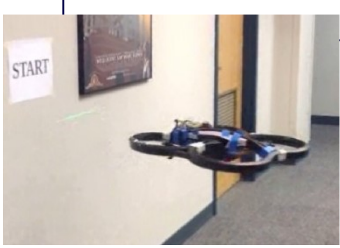
-

- Assume imperfect action execution and **partial observability of the state** (i.e., **imperfect localization**)



Causal relationship

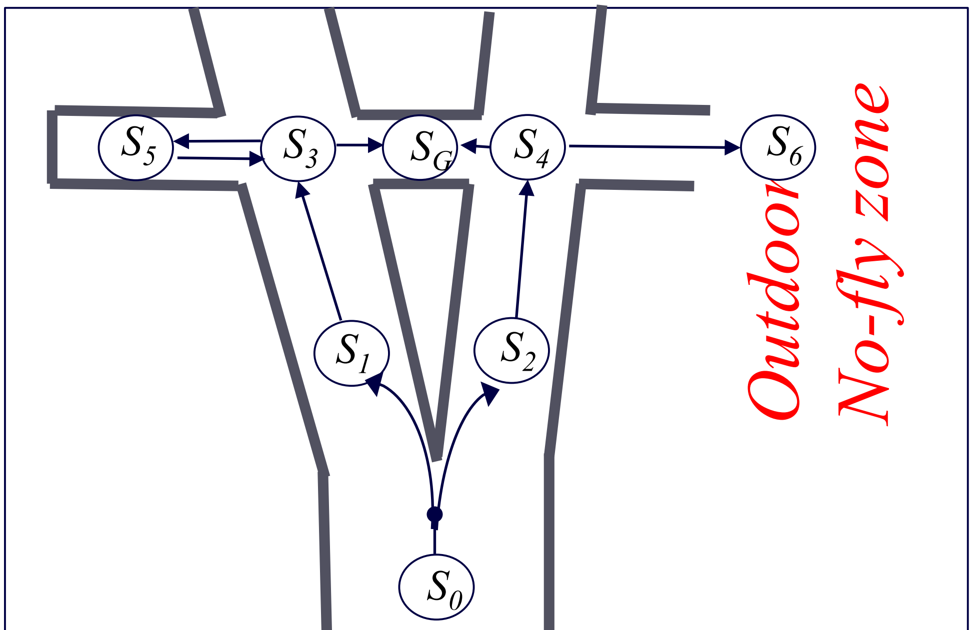




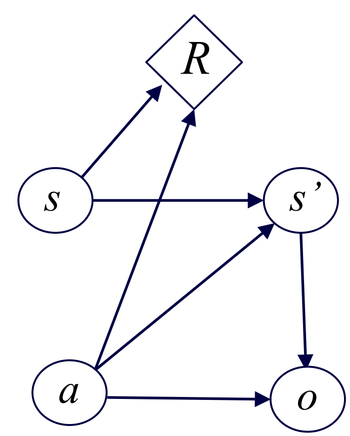
POMDP: *{S, A, T, R, Ω, O}, where S, A, T, R (or C) – same as in MDP, Ω – set of all possible observation vectors o,* ***O(s',a,o) – Prob(o|s',a) probability of seeing o after executing action a and ending up at state s'***

Belief State Space

- **Belief state *b***: Probability distribution over the states the robot believes it is currently in



Causal relationship



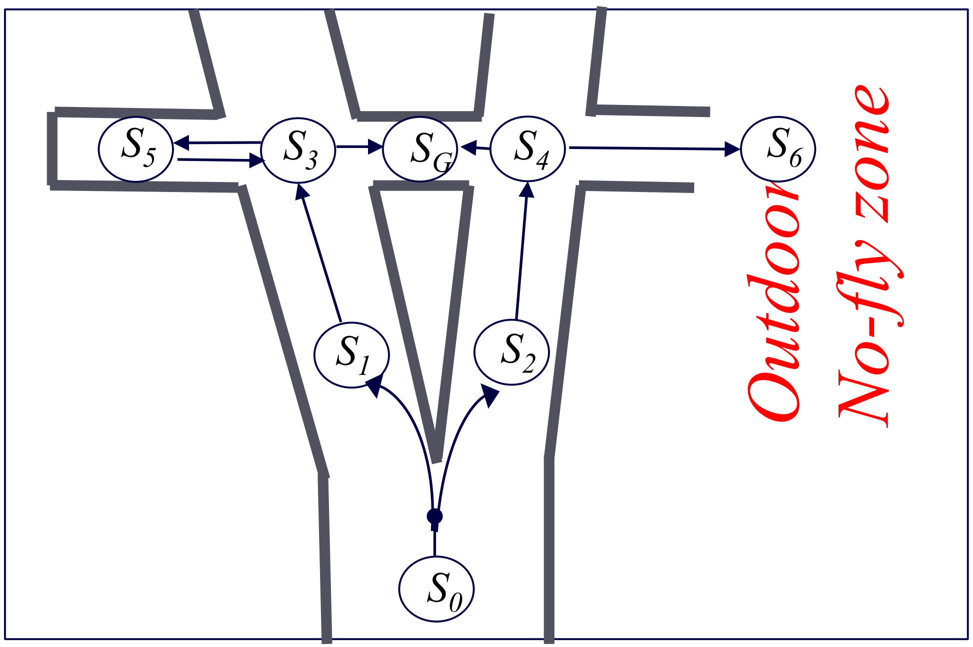
POMDP: *{S, A, T, R, Ω, O}, where* $T(s,a,s') = P(s'|s,a)$, $R(s,a)$, $O(s',a,o) = Prob(o|s',a)$

Belief State Space

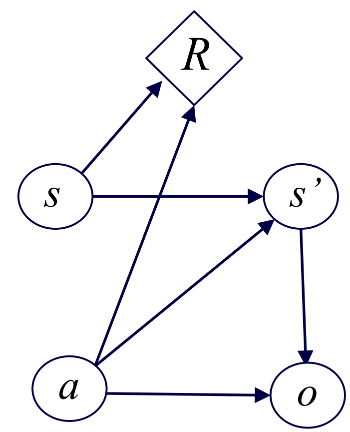
- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

b – a vector of size N (# of states in S)
$\Sigma^N b_i = 1$, and $b_i \geq 0$ for all i

Suppose the robot knows it is initially in s_0.
Then initial $b = [1,0,0,0,0,0,0,0]^T$. That is, $P(s_0) = 1$



Causal relationship



POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s') = P(s'|s,a), R(s,a), O(s',a,o) = Prob(o|s',a)*

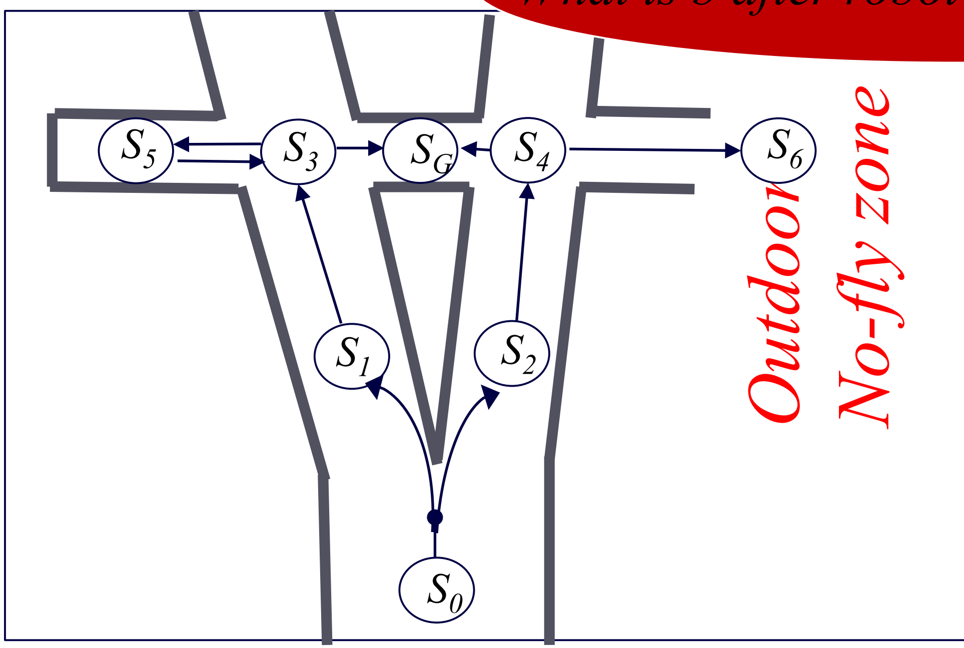
Belief State Space

- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

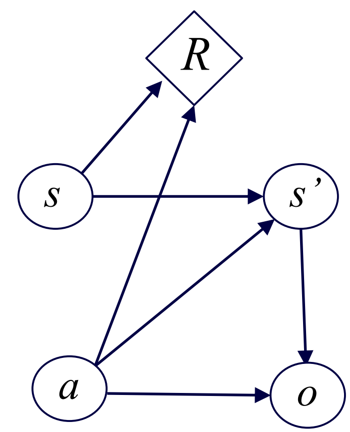
b – a vector of size N (# of states in S)
$\Sigma^N b_i = 1$, and $b_i \geq 0$ for all i

Suppose the robot knows it is initially in s_0.
Then initial $b = [1,0,0,0,0,0,0,0]^T$. That is, $P(s_0) = 1$

What is b after robot takes the 1st action?



Causal relationship

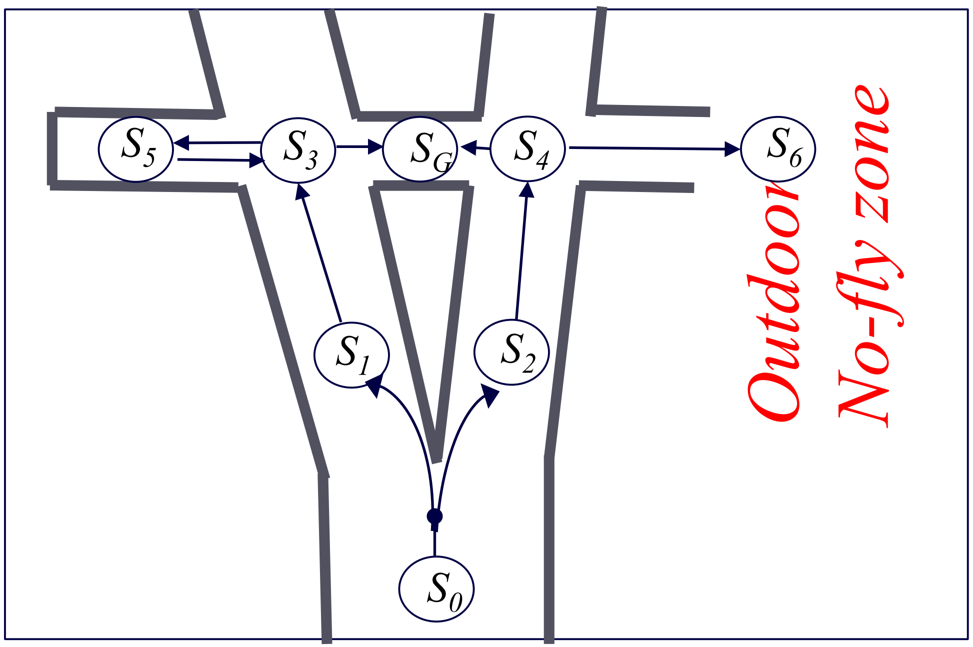


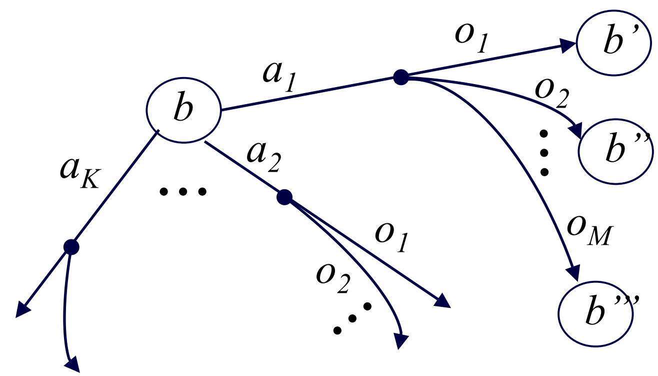
POMDP: *{S, A, T, R, Ω, O}, where $T(s,a,s') = P(s'|s,a)$, $R(s,a)$, $O(s',a,o) = Prob(o|s',a)$*

Belief State Space

- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

Belief State Space
(for K actions, M possible observations)





POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s') = P(s'|s,a), R(s,a), O(s',a,o) = Prob(o|s',a)*

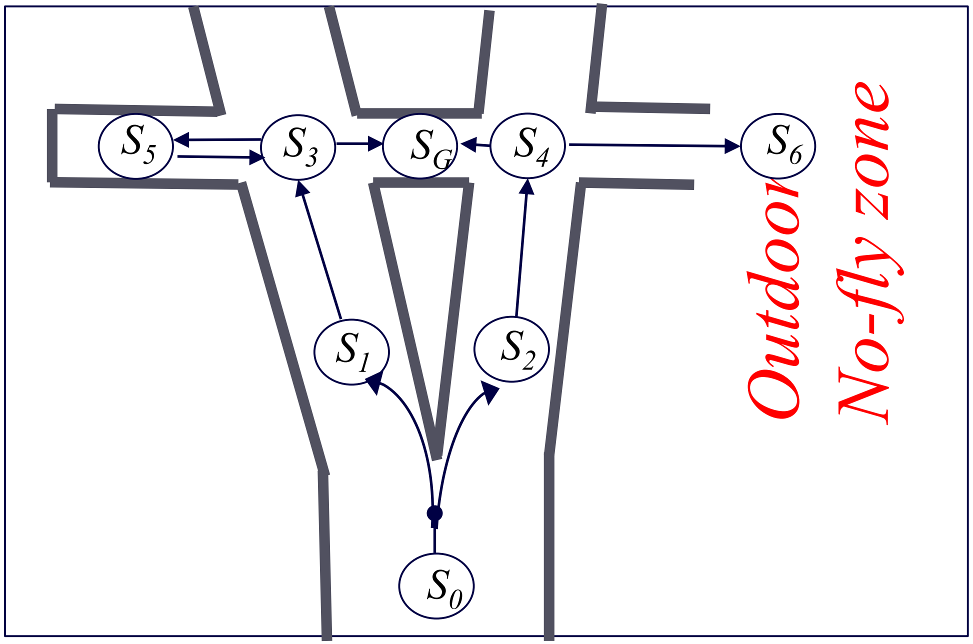
Belief State Space

- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

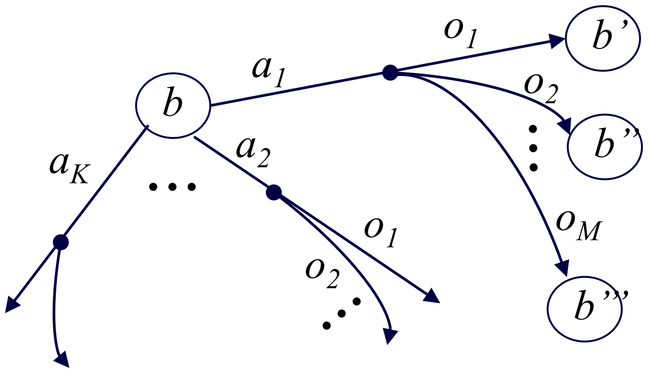
b': P(s'|b,a,o) for every s' in S;

$$b'(s') = P(s'|b,a,o) = \frac{O(s',a,o)\sum_s\{T(s,a,s')*b(s)\}}{P(o|b,a)}$$

Here how outcome beliefs are computed



Belief State Space
(for K actions, M possible observations)



POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s') = P(s'|s,a), R(s,a), O(s',a,o) = Prob(o|s',a)*

Belief State Space

- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

b’: P(s’|b,a,o) for every s’ in S;

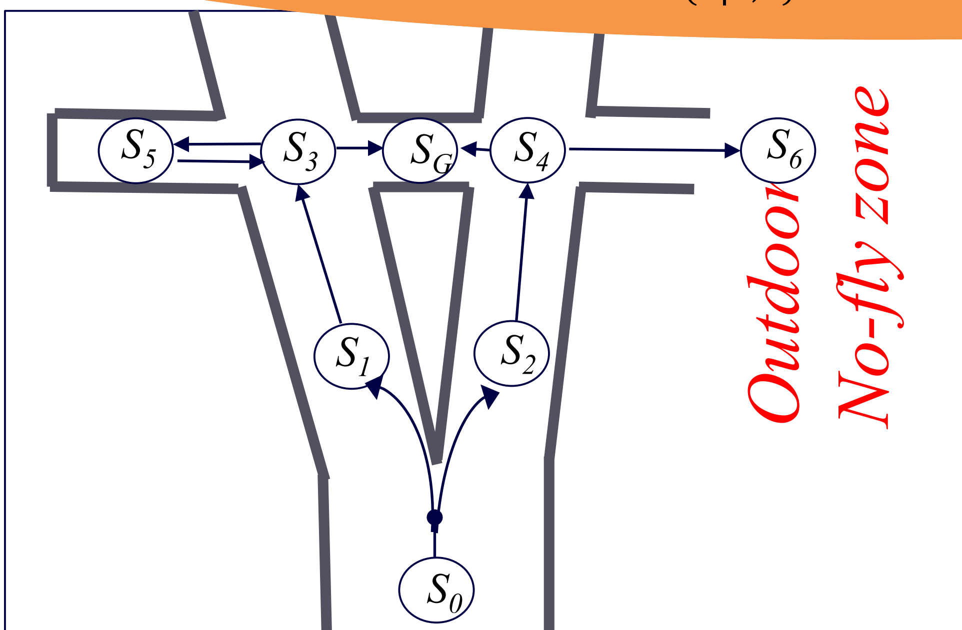
$$b'(s') = P(s'|b,a,o) = \frac{O(s',a,o)\sum_s\{T(s,a,s')*b(s)\}}{P(o|b,a)}$$

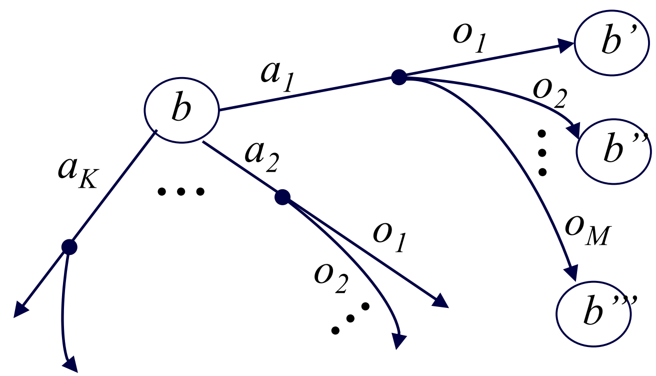
Here how outcome beliefs are computed

Derivation:

$$P(s'|b,a,o) = \frac{P(o|b,a,s')P(s'|b,a)}{P(o|b,a)} = \frac{P(o|s',a)\sum_s\{P(s'|s,a)*P(s)\}}{P(o|b,a)}$$

…ations)





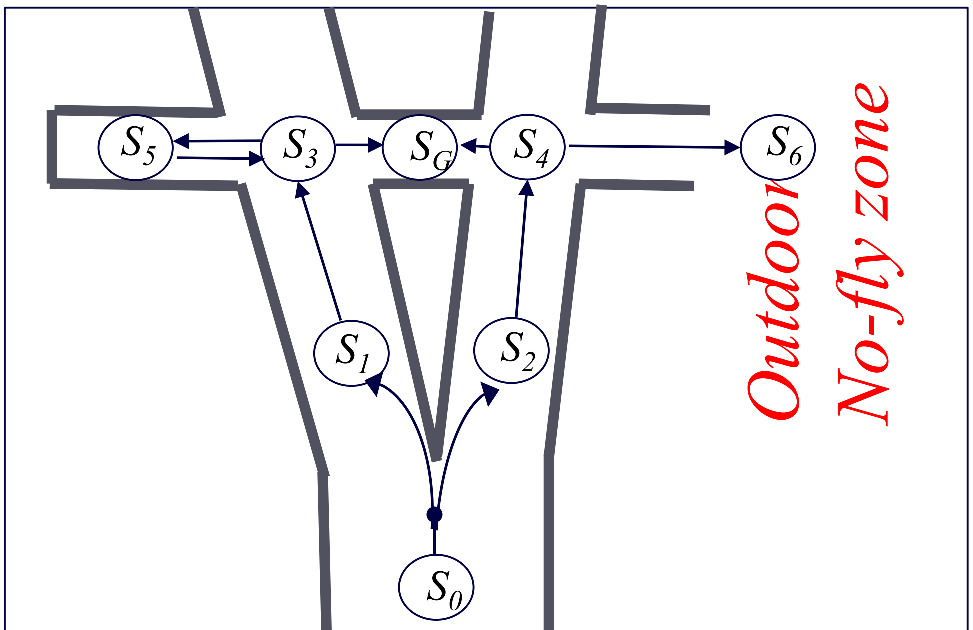
POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s’) = P(s’|s,a), R(s,a), O(s’,a,o) = Prob(o|s’,a)*

Belief State Space

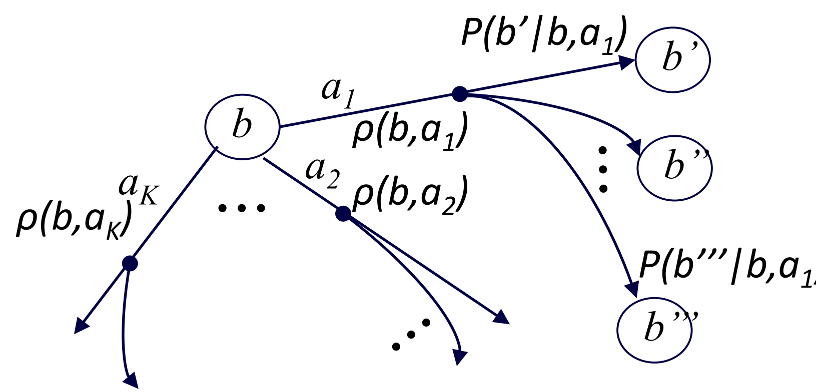
- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

What is Belief State Space?

It is MDP!
We just need to compute transition probabilities $\tau(b,a,b') = P(b'|b,a)$ and reward function $\rho(b,a)$



Belief State Space
(for K actions, M possible observations)

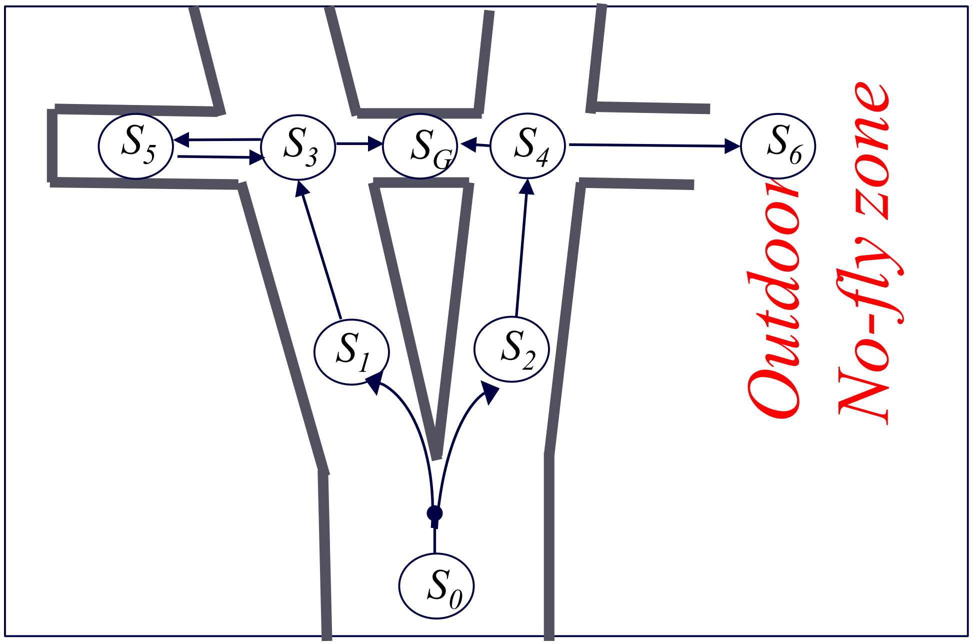


POMDP: *{S, A, T, R, Ω, O}, where $T(s,a,s') = P(s'|s,a)$, $R(s,a)$, $O(s',a,o) = Prob(o|s',a)$*

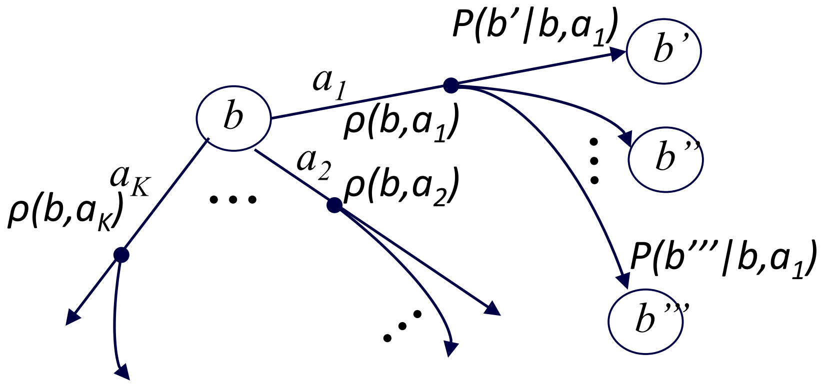
Belief State Space

- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

$\tau(b,a,b') = P(b'|b,a) = \sum_{o \text{ leading to } b'} P(o|b,a) = \sum_{o \text{ leading to } b'} \sum_{s'} P(o|s',a) \sum_{s} P(s'|s,a) b(s)$



Belief State Space
(for K actions, M possible observations)



POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s') = P(s'|s,a), R(s,a), O(s',a,o) = Prob(o|s',a)*

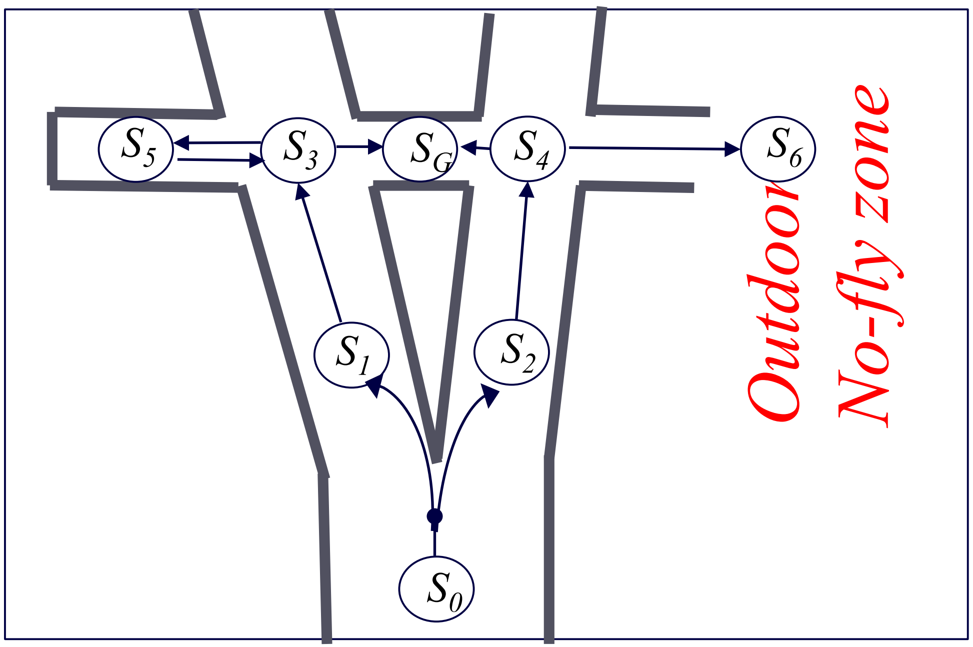
Belief State Space

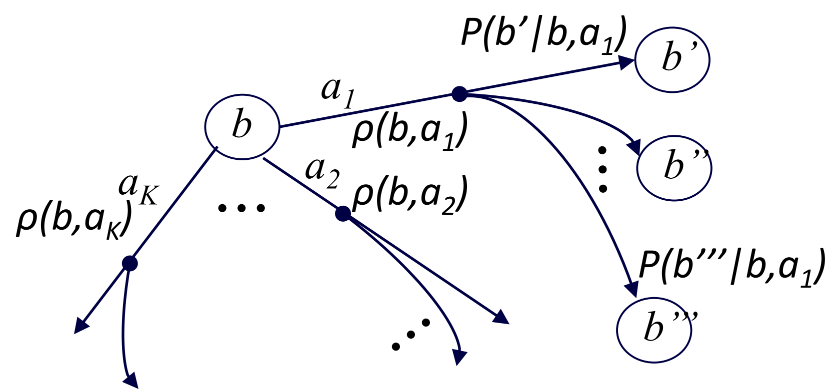
- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

$\tau(b,a,b') = P(b'|b,a) = \sum_{o \text{ leading to } b'} P(o|b,a) = \sum_{o \text{ leading to } b'} \sum_{s'} P(o|s',a) \sum_s P(s'|s,a) b(s)$

$\rho(b,a) = \sum_s R(s,a) b(s)$

Belief State Space
(for K actions, M possible observations)





POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s') = P(s'|s,a), R(s,a), O(s',a,o) = Prob(o|s',a)*

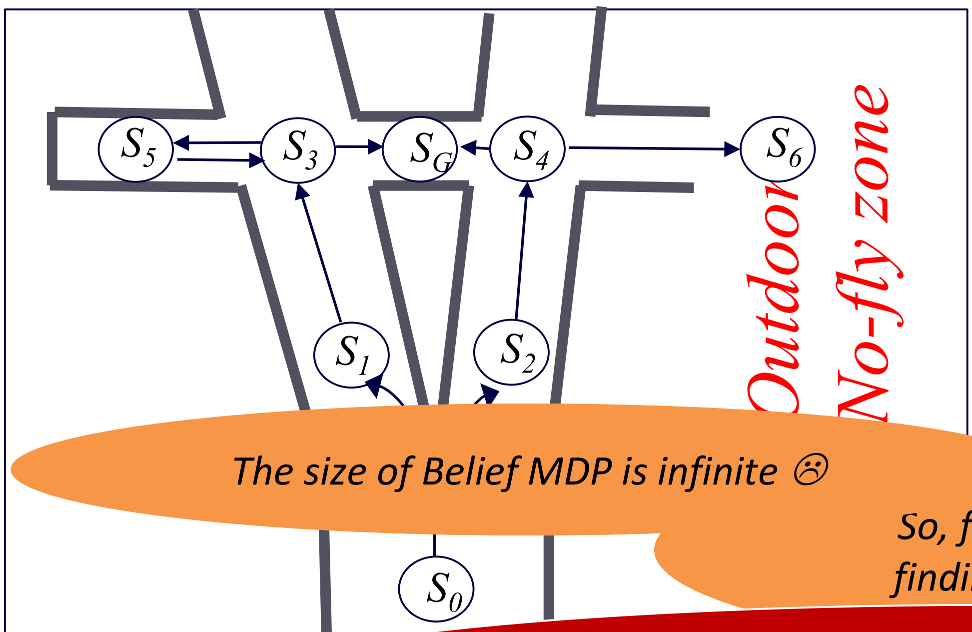
Belief State Space

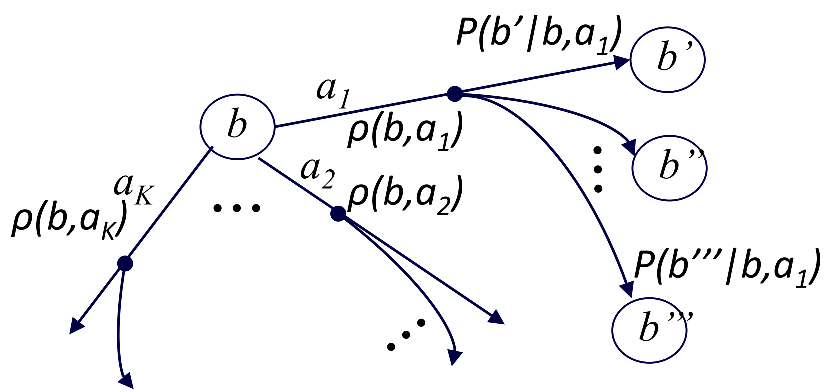
- **Belief state *b***: Probability distribution over the states the robot believes it is currently in

$\tau(b,a,b') = P(b'|b,a) = \sum_{o \text{ leading to } b'} P(o|b,a) = \sum_{o \text{ leading to } b'} \sum_{s'} P(o|s',a) \sum_s P(s'|s,a) b(s)$

$\rho(b,a) = \sum_s R(s,a) b(s)$

Belief State Space
(for K actions, M possible observations)





The size of Belief MDP is infinite ☹

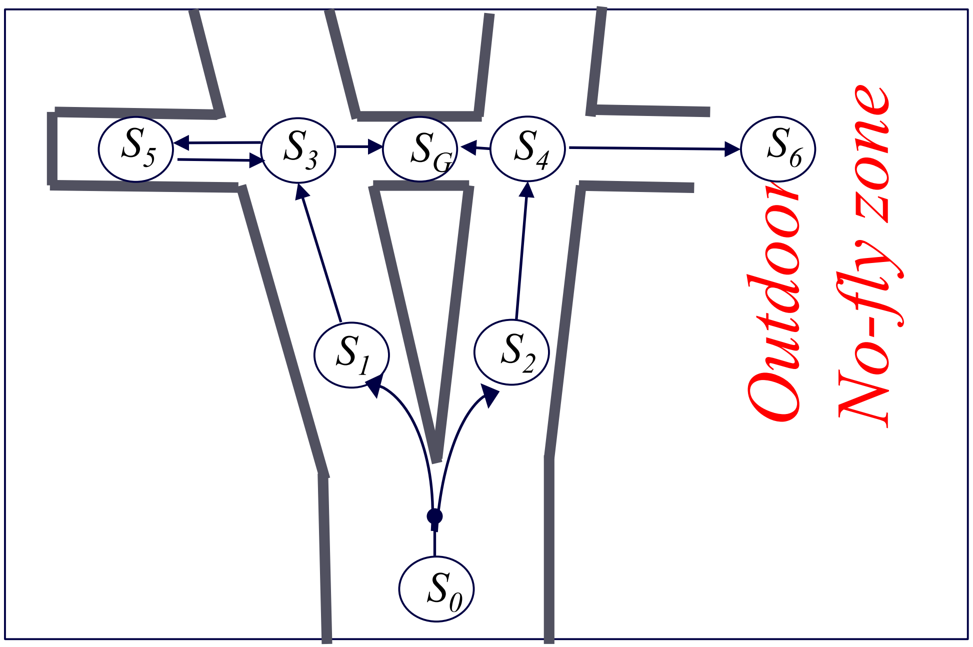
So, finding an optimal policy for POMDP = finding an optimal policy for Belief MDP ☺

POMDP

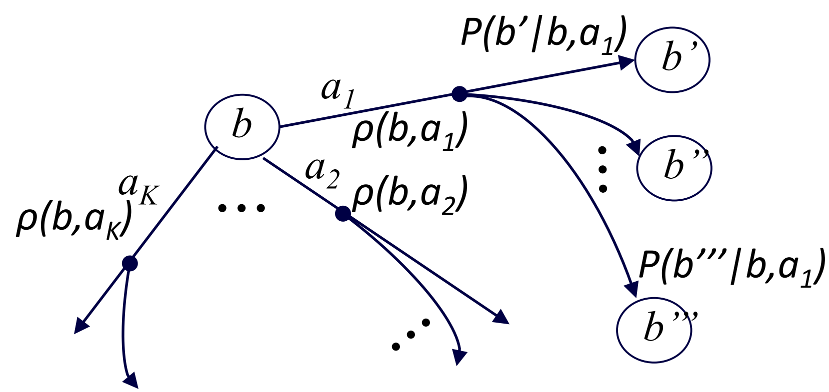
We can even use Value Iteration you studied, can't we?

Belief State Space

- **Belief state *b***: Probability distribution over the states the robot believes it is currently in
- Popular techniques for solving POMDPs
  - by discretizing belief statespace into a finite # of states [Lovejoy, ‘91]
  - by taking advantage of the geometric nature of value function [Kaelbing, Littman & Cassandra, ‘98]
  - by sampling-based approximations [Pineau, Gordon & Thrun, ‘03]



Belief State Space
(for K actions, M possible observations)

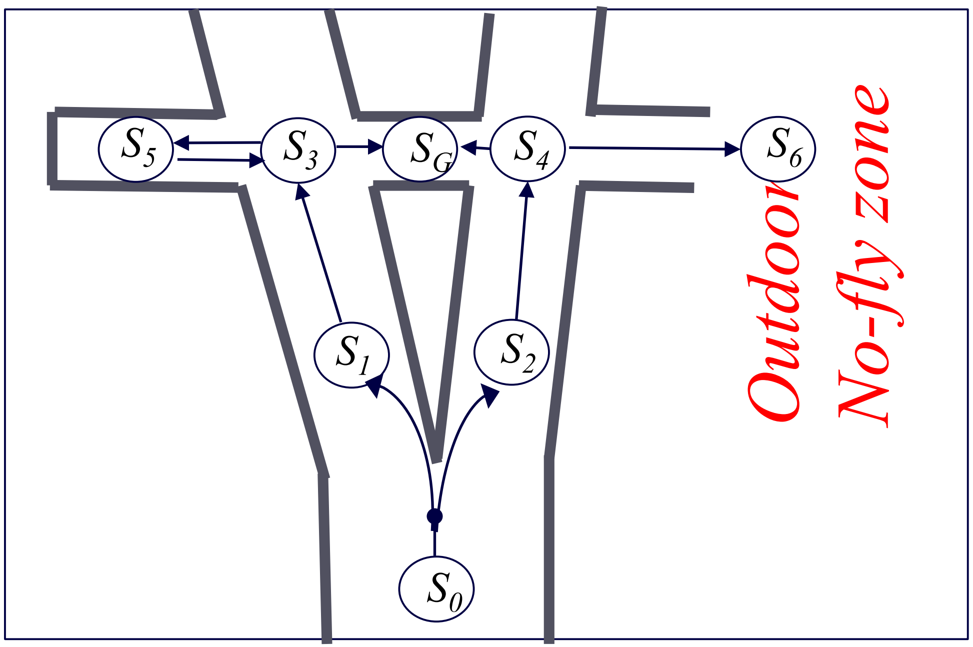


POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s’) = P(s’|s,a), R(s,a), O(s’,a,o) = Prob(o|s’,a)*

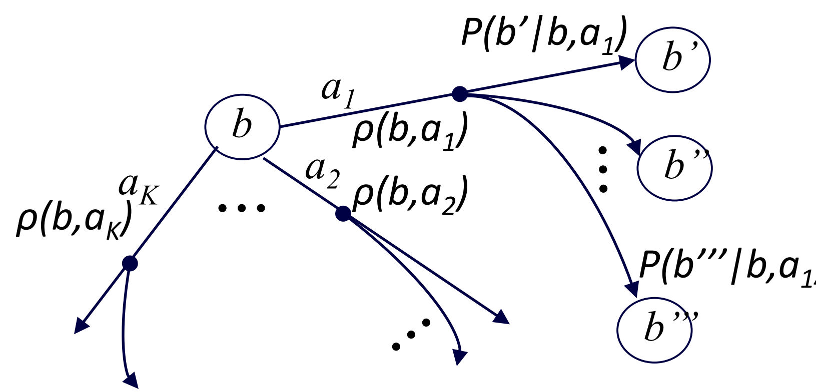
Value Function is piecewise linear and convex

Value function of horizon 1: $V(b, a) = \sum_s R(s,a)b(s)$; $V(b) = max_a \sum_s R(s,a)b(s)$;

How does this look geometrically?



Belief State Space
(for K actions, M possible observations)



POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s') = P(s'|s,a), R(s,a), O(s',a,o) = Prob(o|s',a)*

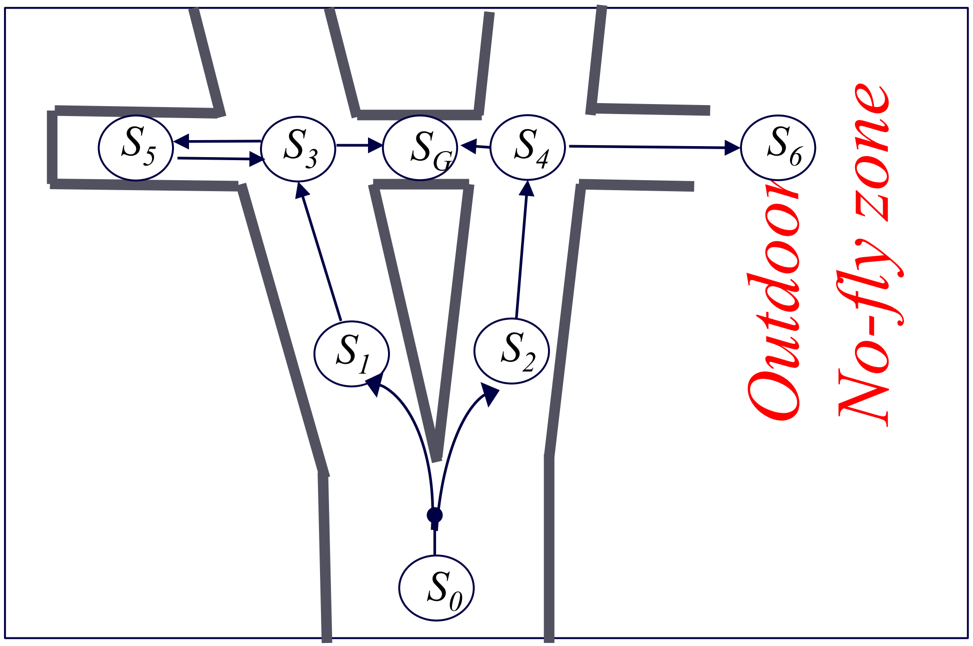
Value Function is piecewise linear and convex

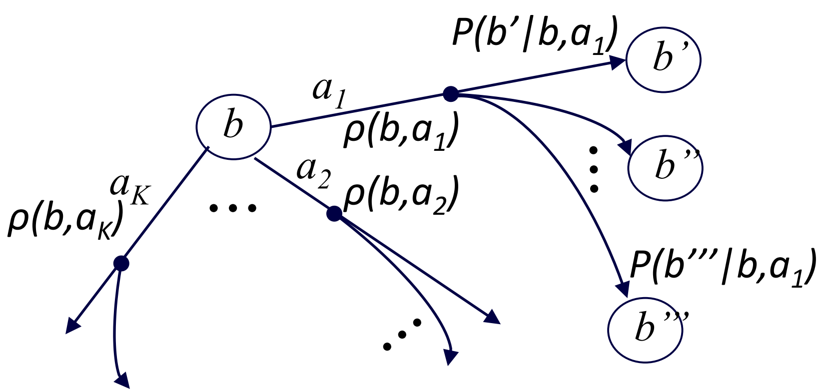
Value function of horizon 1: $V(b, a) = \sum_s R(s,a)b(s)$; $V(b) = max_a \sum_s R(s,a)b(s)$;

Value function of horizon 2: $V(b, a_{t=1}) = \sum_s R(s, a_{t=1})b(s) + \gamma E_{b'}\{V(b')\}b(s)$;

Value Iteration can also be done in the space of these vectors, increasing horizon by 1 at each iteration:
Compute $V(b,a_{t=i})$ = function $(V'(b,a_{t=i-1}))$ done on a set of hyperplanes

(for K actions, M possible observations)





POMDP: *{S, A, T, R, Ω, O}, where T(s,a,s') = P(s'|s,a), R(s,a), O(s',a,o) = Prob(o|s',a)*

What You Should Know…

- What problems should be modeled as planning on Graphs vs. MDPs vs. POMDPs
- How POMDPs can be transformed into a Belief MDP
- How to plan in Belief MDP