

Lecture 40: Exam 3 Review

CC-BY 4.0: share at will, but cite the source

Mark Hasegawa-Johnson, 5/2022

Exam 3 Mechanics

- If you want to take the exam online, fill out the “Exam 3 Online” poll by **May 6**
 - <https://www.gradescope.com/courses/348086/assignments/2021464>
 - Do this regardless of whether you’re in the online or in-person section
 - If you don’t sign up for an online exam, take the exam in person
- If you need a conflict exam, tell us why, and your schedule, by **May 6**
 - <https://www.gradescope.com/courses/348086/assignments/2021473>
 - The date and time of the conflict exam will be chosen May 7, based on the schedules submitted by people who sign up for it, and we will communicate about it by e-mail
 - The exam will happen during the week of May 8 – May 12

Exam 3 Mechanics

- Permitted: two pages of handwritten notes, front & back
- Not permitted: calculators, computers, textbook

Exam 3 Content

- One or two questions based on Exam 1 material
- One or two questions based on Exam 2 material
- Ten to twelve questions based on the last third of the course:
 - Lecture 29: Game theory
 - Lectures 30-31: Two-player games
 - Lecture 32: Markov Decision Process
 - Lectures 33-36: Reinforcement Learning
 - Lectures 37-38: Robots




Game Theory

- Dominant strategy
 - a strategy that's optimal for one player, regardless of what the other player does
 - Not all games have dominant strategies
- Nash equilibrium
 - an outcome (one action by each player) such that, knowing the other player's action, each player has no reason to change their own action
 - Every game with a finite set of actions has at least one Nash equilibrium, though it might be a mixed-strategy equilibrium.
- Pareto optimal
 - an outcome such that neither player would be able to win more without simultaneously forcing the other player to lose more
 - Every game has at least one Pareto optimal outcome. Usually there are many, representing different tradeoffs between the two players.
- Mixed strategies
 - A mixed strategy is optimal only if there's no reason to prefer one action over the other, i.e., if $0 \leq p \leq 1$ and $0 \leq q \leq 1$ such that:

$$\begin{aligned}(1 - p)w + px &= (1 - p)y + pz \\ (1 - q)a + qc &= (1 - q)b + qd\end{aligned}$$

Two-Player Games

- Alternating two-player zero-sum games

-  = max node,  = min node,  = chance node

- Expectiminimax search

$$U(s) = \max_a \sum_{s'} P(s'|s, a)U(s') \text{ or } U(s') = \min_{a'} \sum_{s''} P(s''|s', a')U(s'')$$

- Limited-horizon computation and heuristic evaluation functions

$$U(s) = w_1 f_1(s) + w_2 f_2(s) + \dots$$

- Alpha-beta search

- Min node can update beta, Max node can update alpha
- If beta ever falls below alpha, prune the rest of the children

- Computational complexity of minimax and alpha-beta

- Minimax is $O\{b^d\}$. With optimal move ordering, alpha-beta is $O\{b^{d/2}\}$.

- Stochastic search: $U(s) \approx \frac{1}{n} \sum_{i=1}^n U(i^{th} \text{ random game starting from } s)$

Markov Decision Process

- The exact solution is given by Bellman's equation:

$$U(s) = R(s) + \gamma \max_a \sum_{s'} P(s'|s, a)U(s') \quad \forall s, s'$$

- Value iteration starts with length-0 paths, and iteratively extends them:

$$U_t(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s, a)U_{t-1}(s')$$

- Policy iteration alternates two steps:

- Policy evaluation: find out the value of each state under current policy:

$$U^\pi(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi(s))U^\pi(s')$$

- Policy improvement: change the action, in each state, to improve value:

$$\pi(s) = \operatorname{argmax}_{a \in A(s)} \sum_{s'} P(s'|s, a)U^\pi(s')$$

Reinforcement Learning

- Model-Based: learn $P(s'|s, a)$ and $R(s)$, then solve the MDP
- Q-learning: $Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha(Q_{local}(s_t, a_t) - Q_t(s_t, a_t))$
 - TD: $Q_{local}(s_t, a_t) = R_t(s_t) + \gamma \max_{a' \in A(s_{t+1})} Q_t(s_{t+1}, a')$
 - SARSA: $Q_{local}(s_t, a_t) = R_t(s_t) + \gamma Q_t(s_{t+1}, a_{t+1})$
- Deep Q-learning: $\mathcal{L} = \frac{1}{2} E[(Q_t(\vec{s}_t, \vec{a}_t) - Q_{local}(\vec{s}_t, \vec{a}_t))^2]$
- Imitation Learning: $\mathcal{L} = -\log \pi_{a_t}(\vec{s}_t)$
- Actor-Critic: $\mathcal{L}_{actor} = -\sum_a \pi_a(s) Q_t(s, a)$

How to solve the Robot Arm problem

1. Create a configuration space (a space whose coordinates are the set of all configuration parameters for the robot). Typically,

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

2. Label the START
3. Label the GOAL (there might be more than one set of configuration parameters that is an acceptable way to reach the GOAL).
4. Label the OBSTACLES (convert them from (x, y) to (θ_1, θ_2)).
5. Use BFS or A* (or value iteration if the environment is stochastic, or RL if the environment is unknown) to find the shortest path from START to GOAL, avoiding all OBSTACLES.

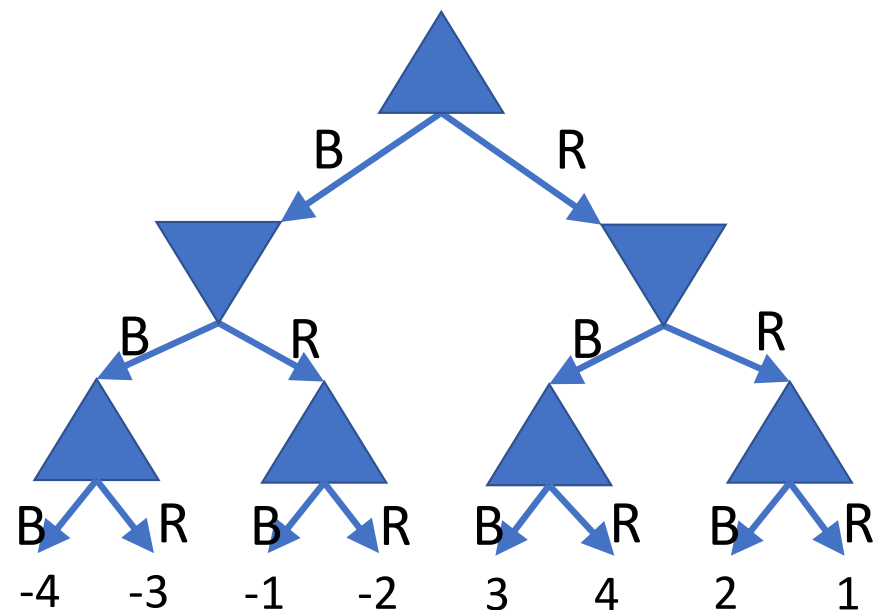
Sample problem: Game theory

	Alice R1	Alice R2
Bob R1	A:5,B:5	A:10,B:10
Bob R2	A:10,B:10	A:5,B:5

- Is this a zero-sum game?
 - No
- Find Pareto-optimal outcomes
 - (R1,R2) and (R2,R1)
- Find dominant strategy, if any
 - There are none
- Find fixed-strategy Nash equilibrium, if any
 - (R1,R2) and (R2,R1)
- Find mixed-strategy Nash equilibrium, if any
$$5p + 10(1 - p) = 10p + 5(1 - p)$$
 - Solution: $p=0.5$. Game is symmetric, so $q=0.5$ also.

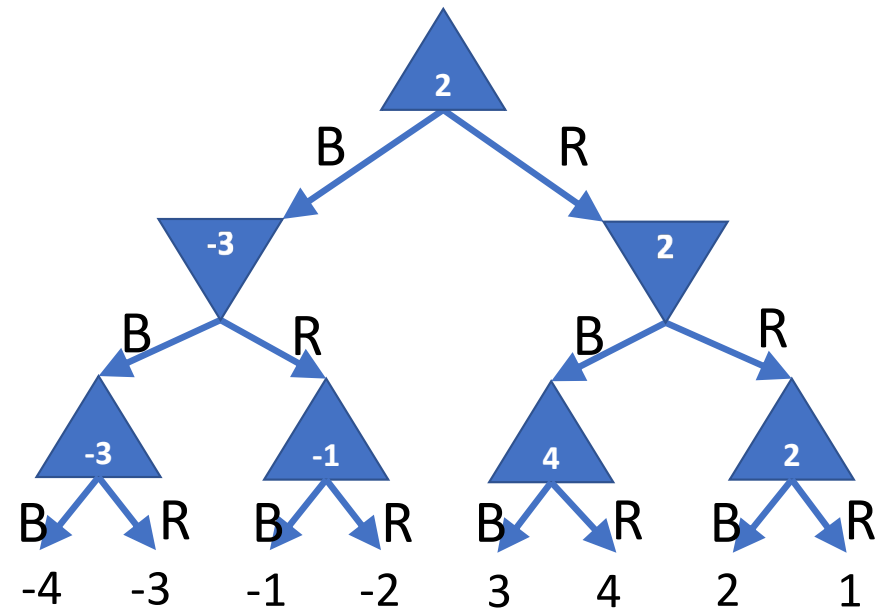
Sample problem: Minimax and Alpha-beta

- What is the value of each node?
- Re-arrange the tree so that, if moves are evaluated in order from right to left, alpha-beta will only evaluate 5 of the 8 terminal nodes
- After your re-arrangement, which edges are pruned



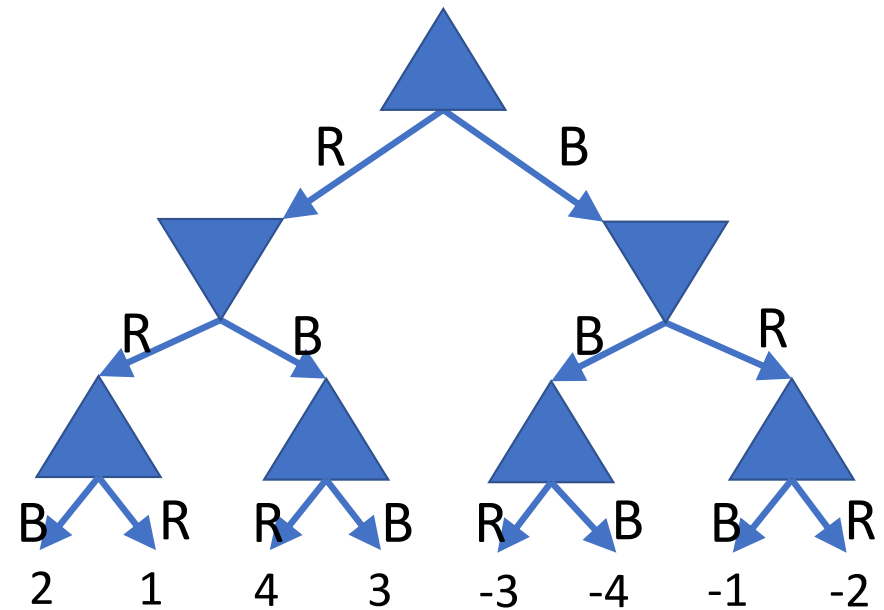
Sample problem: Minimax and Alpha-beta

- What is the value of each node?
- Re-arrange the tree so that, if moves are evaluated in order from right to left, alpha-beta will only evaluate 5 of the 8 terminal nodes
- After your re-arrangement, which edges are pruned



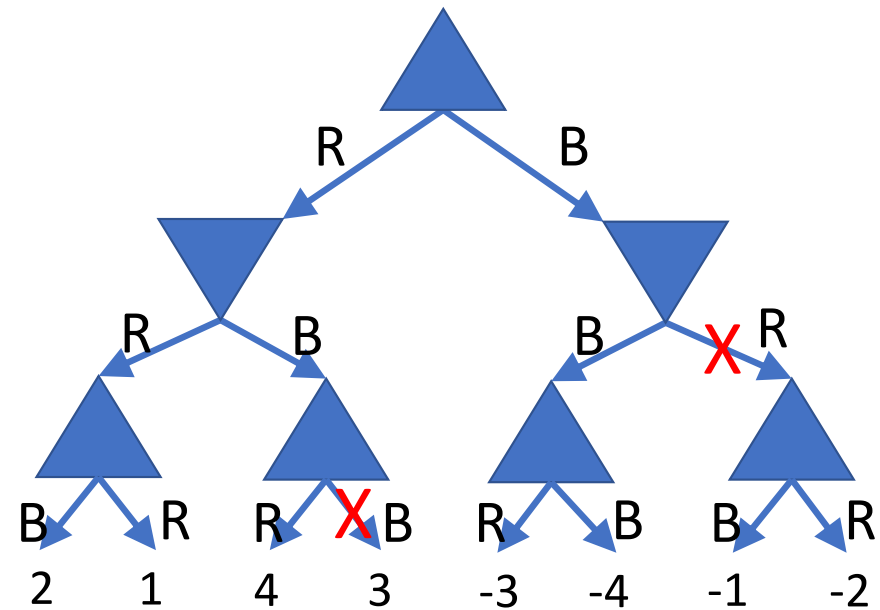
Sample problem: Minimax and Alpha-beta

- What is the value of each node?
- Re-arrange the tree so that, if moves are evaluated in order from right to left, alpha-beta will only evaluate 5 of the 8 terminal nodes
- After your re-arrangement, which edges are pruned



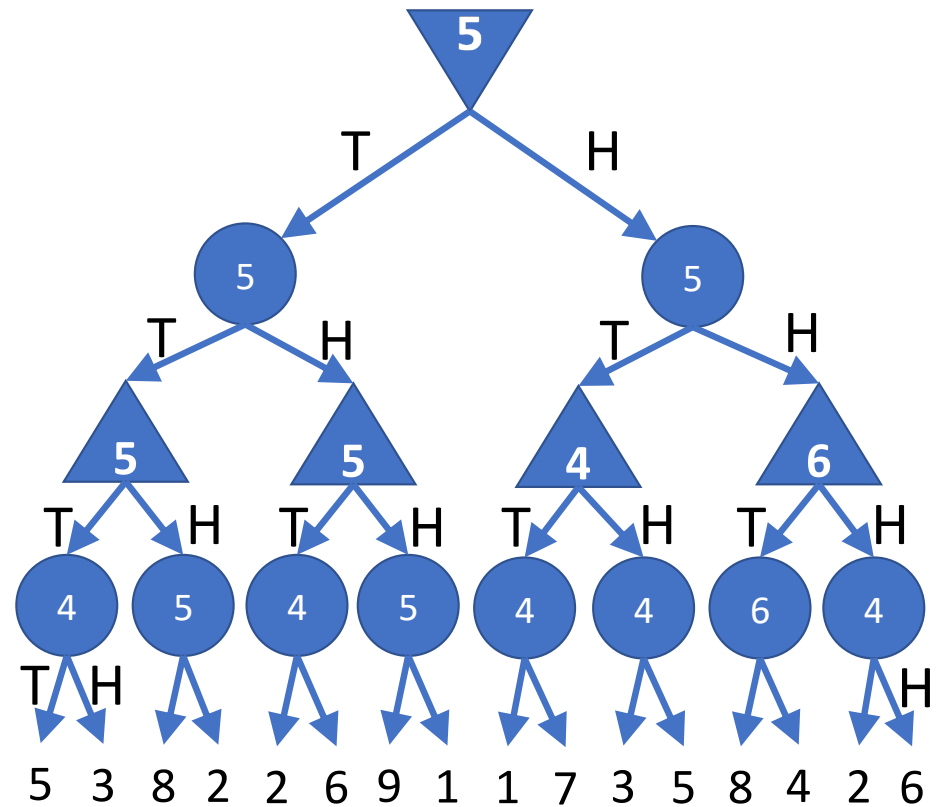
Sample problem: Minimax and Alpha-beta

- What is the value of each node?
- Re-arrange the tree so that, if moves are evaluated in order from right to left, alpha-beta will only evaluate 5 of the 8 terminal nodes
- After your re-arrangement, which edges are pruned



Expectiminimax

- Find values of all nodes



Markov Decision Process

Consider a reduced-size Gridworld, with rewards in each state as shown. Agent starts in the state with $R(s)=0$. Actions include moving to a neighboring state or staying in the same state. Suppose

$$P(s'|s, a) = \begin{cases} 0.8 & s' = a, a \in \text{Neighbors}(s) \\ 0.2 & s' = s \\ 0 & \text{otherwise} \end{cases}$$

- Consider a value iteration with $U_1(s) = R(s)$, $\gamma = 1$. Find $U_2(s)$ for all states.
- What's the smallest t for which $U_t(s) > 0$?

$R(s)$	Column 1	Column 2
Row 1	-0.04	-0.04
Row 2	-1	1
Row 3	0	-0.04

Markov Decision Process

Consider a reduced-size Gridworld, with rewards in each state as shown. Agent starts in the state with $R(s)=0$; game ends when agent reaches a state with $R(s)=-1$ or $R(s)=1$. Actions include moving to a neighboring state or staying in the same state. Suppose

$$P(s'|s, a) = \begin{cases} 0.8 & s' = a, a \in \text{Neighbors}(s) \\ 0.2 & s' = s \\ 0 & \text{otherwise} \end{cases}$$

- a) Consider a value iteration with $U_1(s) = R(s), \gamma = 1$. Find $U_2(s)$ for all states.

Answer: $U_2(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s, a) U_1(s')$

- b) What's the smallest t for which $U_t(s) > 0$?

Answer: $t = \text{length of the shortest path with a positive total reward, i.e., 3.}$

$R(s)$	Column 1	Column 2
Row 1	-0.04	-0.04
Row 2	-1	1
Row 3	0	-0.04

$U_2(s)$	Column 1	Column 2
Row 1	-0.08	0.752
Row 2	-1	1
Row 3	0	0.752

Reinforcement Learning

Consider a reduced-size Gridworld, with rewards in each state as shown. Agent starts in the state with $R(s)=0$. Suppose that $P(s'|s, a)$ is unknown. Actions include moving to a neighboring state or staying in the same state. The agent performs the following action:

- Starting state: $(3,1)$, the state with $R(s)=0$
 - Action: tries to move Right
 - Ending state: $s'=(3,2)$, the move succeeds
- a) Use Laplace smoothing, with $k=1$, to re-estimate $P(s'|s, a)$.
 - b) Assume that $Q(s, a)$ was 0 for all states initially. Using one step of TD learning, what is $Q(s, a)$, now, for $s=(3,1)$ and action=Right?

$R(s)$	Column 1	Column 2
Row 1	-0.04	-0.04
Row 2	-1	1
Row 3	0	-0.04

Reinforcement Learning

Consider a reduced-size Gridworld, with rewards in each state as shown. Agent starts in the state with $R(s)=0$. Suppose that $P(s'|s, a)$ is unknown, but we know that, for any action, the possible outcomes are only $s' = s$ or $s' \in \text{Neighbors}(s)$. Actions include moving to a neighboring state or staying in the same state. The agent performs the following action:

- Starting state: $(3,1)$, the state with $R(s)=0$
 - Action: tries to move Right
 - Ending state: $s'=(3,2)$, the move succeeds
- a) Use Laplace smoothing, with $k=1$, to re-estimate $P(s'|s, a)$ for the given (s, a, s') .
- b) Assume that $Q(s, a)$ was 0 for all states initially. Using one step of TD learning, what is $Q(s, a)$, now, for $s=(3,1)$ and action=Right?

$R(s)$	Column 1	Column 2
Row 1	-0.04	-0.04
Row 2	-1	1
Row 3	0	-0.04

$$P(s' = (3,2)|s = (3,1), a = \text{Right}) = \frac{1 + k}{1 + 3k} = \frac{2}{4}$$

The $3k$ in the denominator is because there are three possible outcomes: $s' = s$ or $s' \in \text{Neighbors}(s)$.

Reinforcement Learning

Consider a reduced-size Gridworld, with rewards in each state as shown. Agent starts in the state with $R(s)=0$. Suppose that $P(s'|s, a)$ is unknown, but we know that, for any action, the possible outcomes are only $s' = s$ or $s' \in Neighbors(s)$. The agent performs the following action:

- Starting state: (3,1), the state with $R(s)=0$
 - Action: tries to move Right
 - Ending state: $s'=(3,2)$, the move succeeds
- a) Use Laplace smoothing, with $k=1$, to re-estimate $P(s'|s, a)$ for the given (s, a, s') .
- b) Assume that $Q_0(s, a)$ was 0 for all states initially. Using one step of TD learning, with $\gamma = 1$ and $\alpha=1$, what is $Q_1(s, a)$, now, for $s=(3,1)$ and action=Right?

$R(s)$	Column 1	Column 2
Row 1	-0.04	-0.04
Row 2	-1	1
Row 3	0	-0.04

$$Q_{local}(s_t, a_t) = R_t(s_t) + \gamma \max_{a' \in A(s_{t+1})} Q_t(s_{t+1}, a')$$
$$= 0 + 0 = 0$$

$$Q_1(s, a) = Q_0(s, a) + \alpha(Q_{local}(s, a) - Q_0(s, a))$$
$$= 0 + (0 - 0) = 0$$

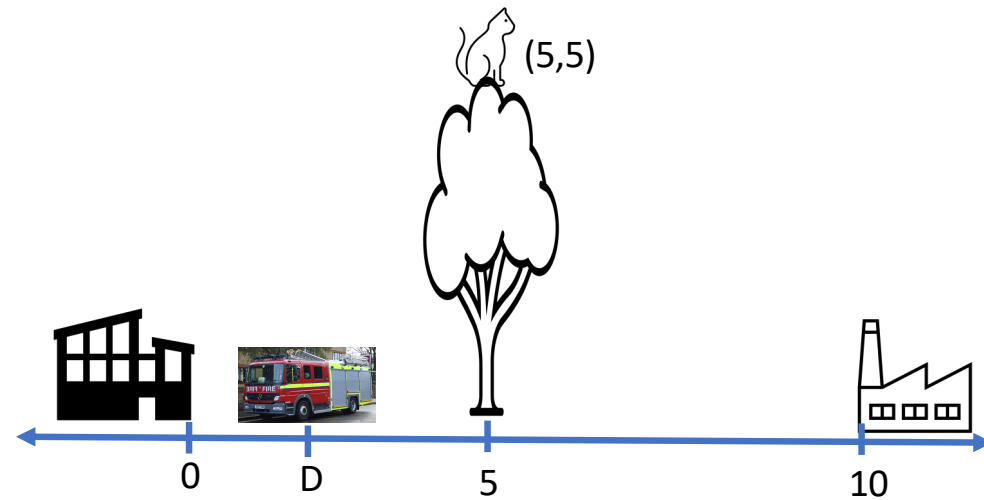
Robotics

A robot firetruck wants to rescue a cat. In order to do so, it must get the tip of its ladder up to the cat's location, $(x, z) = (5, 5)$. It needs to do so by manipulating its own horizontal location (D), the length of its ladder (L), and the angle of the ladder (θ):

$$(x, z) = (D + L \cos \theta, L \sin \theta)$$

There are buildings at locations $x = 0$ and $x = 10$. If either the base of the firetruck or the tip of its ladder comes within 1 unit of a building, it crashes.

- What is the dimension of the configuration space?
- Write one or more inequalities describing regions in configuration space that the robot cannot enter, because it would be running into a building.
- Write one or more equations describing the location of the goal in configuration space. How many configurations would reach the goal?



Robotics

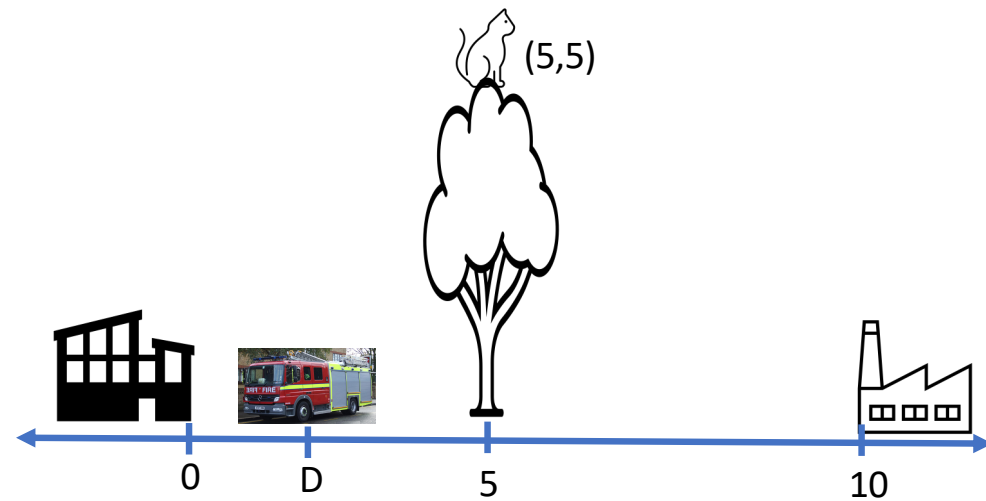
A robot firetruck wants to rescue a cat. In order to do so, it must get the tip of its ladder up to the cat's location, $(x, z) = (5, 5)$. It needs to do so by manipulating its own horizontal location (D), the length of its ladder (L), and the angle of the ladder (θ):

$$(x, z) = (D + L \cos \theta, L \sin \theta)$$

There are buildings at locations $x = 0$ and $x = 10$. If either the base of the firetruck or the tip of its ladder comes within 1 unit of a building, it crashes.

- a) What is the dimension of the configuration space?

Answer: three. (D, L, θ)



Robotics

A robot firetruck wants to rescue a cat. In order to do so, it must get the tip of its ladder up to the cat's location, $(x, z) = (5, 5)$. It needs to do so by manipulating its own horizontal location (D), the length of its ladder (L), and the angle of the ladder (θ):

$$(x, z) = (D + L \cos \theta, L \sin \theta)$$

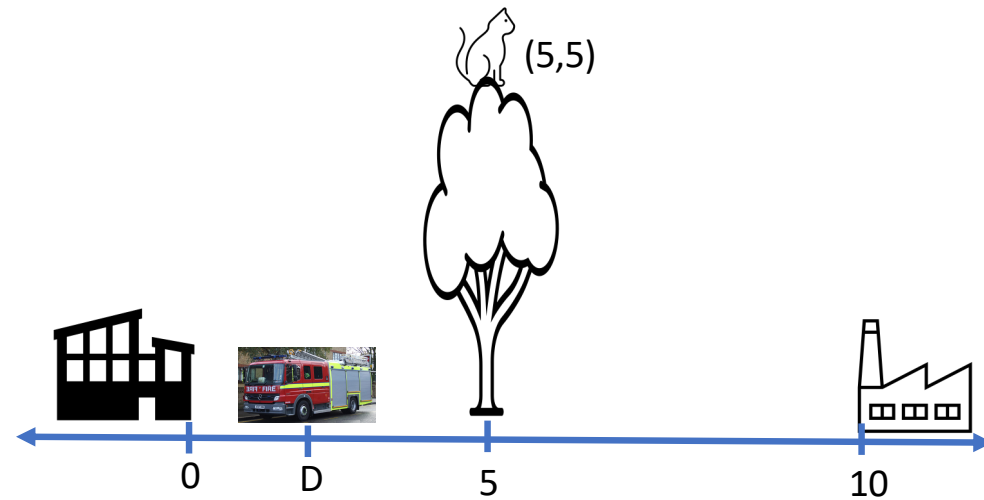
There are buildings at locations $x = 0$ and $x = 10$. If either the base of the firetruck or the tip of its ladder comes within 1 unit of a building, it crashes.

- b) Write one or more inequalities describing regions in configuration space that the robot cannot enter, because it would be running into a building.

Answer: The following regions are off limits:

Base: $D < 1, D > 9$

Ladder: $D + L \cos \theta < 1, D + L \cos \theta > 9$



Robotics

A robot firetruck wants to rescue a cat. In order to do so, it must get the tip of its ladder up to the cat's location, $(x, z) = (5, 5)$. It needs to do so by manipulating its own horizontal location (D), the length of its ladder (L), and the angle of the ladder (θ):

$$(x, z) = (D + L \cos \theta, L \sin \theta)$$

There are buildings at locations $x = 0$ and $x = 10$. If either the base of the firetruck or the tip of its ladder comes within 1 unit of a building, it crashes.

- c) Write one or more equations describing the location of the goal in configuration space. How many configurations would reach the goal?

Answer: There are an infinite number of configurations that would reach the goal; any combination of (D, L, θ) such that:

$$(D + L \cos \theta, L \sin \theta) = (5, 5)$$

