CSCE 580: Introduction to AI

Weeks 12-13 - Lectures 22, 23, 24, 25: SDP: Planning and Reinforcement Learning (RL)

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

4TH, 6TH 11TH, AND 13TH NOV 2025

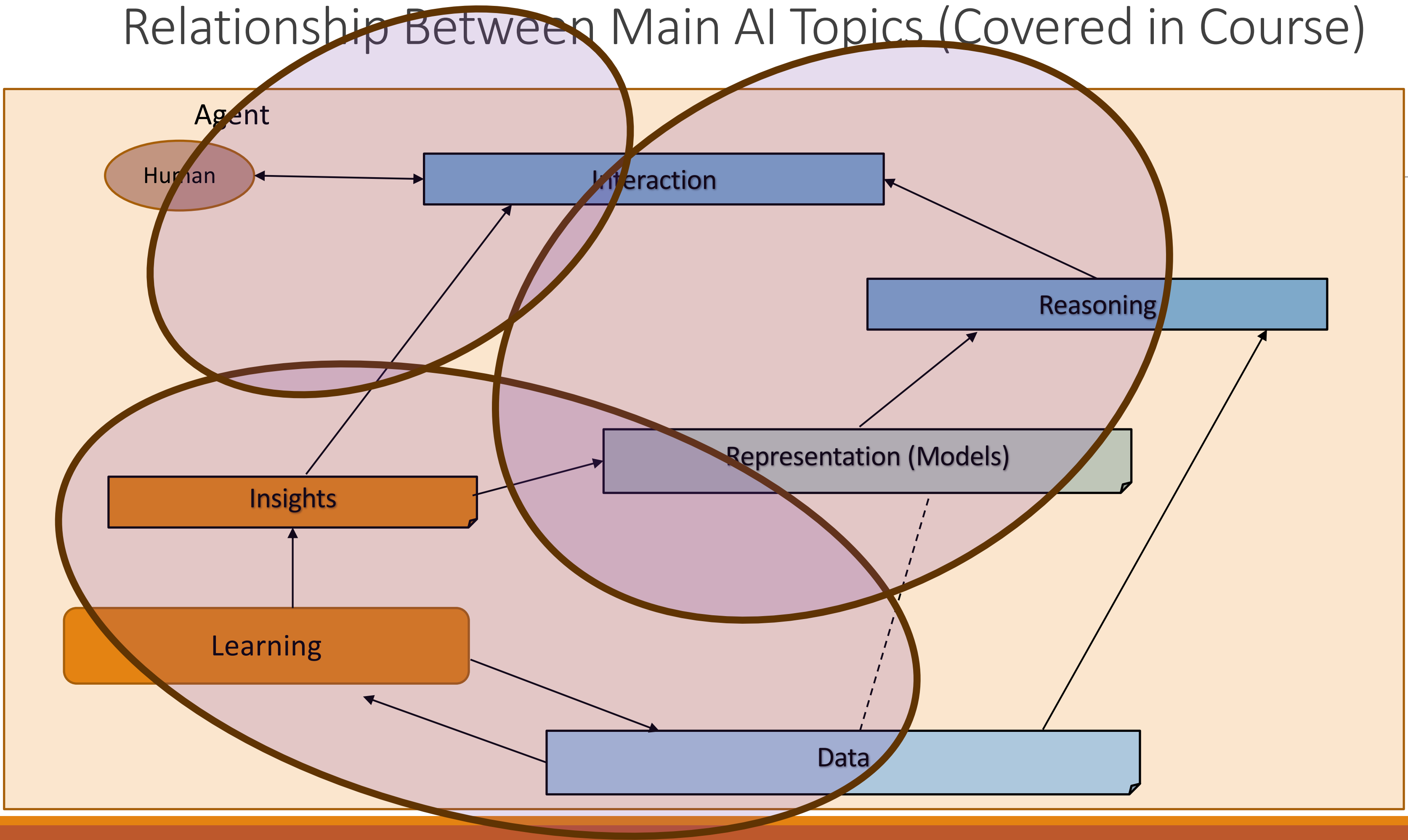
Carolinian Creed: "I will practice personal and academic integrity."

Credits: **Copyrights of all material reused acknowledged**

Organization of Week 12 - Lectures 22-25

- Introduction Section
  - Recap from Week 11 (Lectures 20 and 21)
  - AI news
- Main Section
  - Lecture 22: SPD: Planning
  - Lecture 23: Language Models and Planning
  - Lecture 24: Quiz 3
  - Lecture 25: SDP: RL
- Concluding Section
  - About next week – W14: Lectures 26, 27
  - Ask me anything

Relationship Between Main AI Topics (Covered in Course)



Recap of Week 11

We discussed

- Lecture 20: Making Decisions - Simple
- Lecture 21: Making Decisions - Complex

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 3: Machine Learning – Supervised (Classification)
- Week 4: Machine Learning - Unsupervised (Clustering) –
- Topic 5: Learning neural network, deep learning, Adversarial attacks
- Week 6: Large Language Models – Representation and Usage issues
- Weeks 7-8: Search, Heuristics - Decision Making
- Week 9: Constraints, Optimization – Decision Making
- Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
- Week 13: Trustworthy Decision Making: Explanation, AI testing
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Introduction Section

Upcoming Evaluation Milestones

- **Projects B: Sep 30 – Nov 20**
- Quiz 2: Oct 7
- **Quiz 3: Nov 11**
- Paper presentation (grad students only) : Nov 18
  - Put paper names in spreadsheed
- Finals: Dec 11

AI News

#1 NEWS – Law School Tests Trial With Jury Made Up of ChatGPT, Grok, and Claude

- **Link**: https://futurism.com/artificial-intelligence/law-school-trial-ai-jury

- University of North Carolina School of Law had a mock trial
- Chatbots acted as "jurors" who would determine the fate of a man charged with juvenile robbery - *The Trial of Henry Justus*
- AI "jurors" (ChatGPT, Grok and Claude) were given a real-time transcript of the proceedings and then "deliberated" in front of the audience
- Uneven performance
    - "most in the audience came away believing that trial-by-bot is not a good idea"

- Lawyers have been using chatbots in different use cases in law
    - https://www.404media.co/18-lawyers-caught-using-ai-explain-why-they-did-it/

#2 NEWS – Google Uncovers PROMPTFLUX Malware That Uses Gemini AI to Rewrite Its Code Hourly

- **Link**: https://thehackernews.com/2025/11/google-uncovers-promptflux-malware-that.html

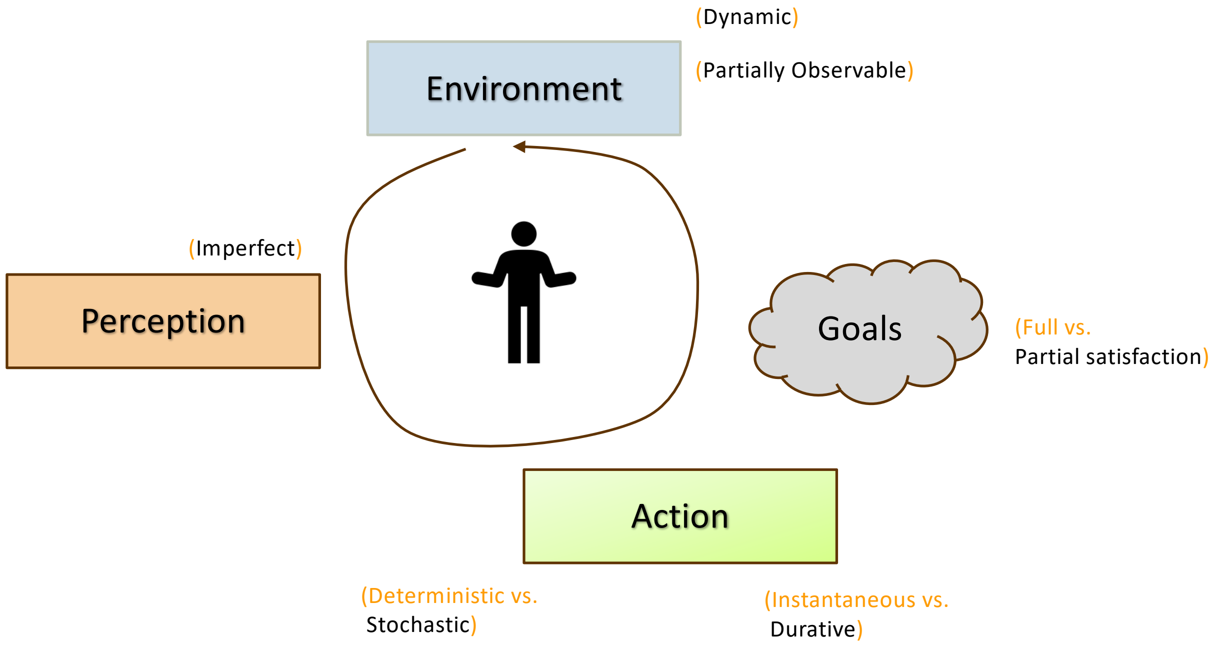
- Experimental Visual Basic Script (VB Script) malware dubbed **PROMPTFLUX**
- Interacts with its Gemini artificial intelligence (AI) model API to write its own source code for improved obfuscation and evasion
  - periodically queries the large language model (LLM), Gemini 1.5 Flash or later in this case, to obtain new code so as to sidestep detection.
  - This, in turn, is accomplished by using a hard-coded API key to send the query to the Gemini API endpoint.
  - Malware saves the new, obfuscated version to the Windows Startup folder to establish persistence and attempts to propagate by copying itself to removable drives and mapped network shares.
- Others
  - **FRUITSHELL**, a reverse shell written in PowerShell that includes hard-coded prompts to bypass detection or analysis by LLM-powered security systems
  - **PROMPTLOCK**, a cross-platform ransomware written in Go that uses an LLM to dynamically generate and execute malicious Lua scripts at runtime (identified as a proof-of-concept)
  - **PROMPTSTEAL** (aka LAMEHUG), a data miner used by the Russian state-sponsored actor APT28 in attacks targeting Ukraine that queries Qwen2.5-Coder-32B-Instruct to generate commands for execution via the API for Hugging Face
  - **QUIETVAULT**, a credential stealer written in JavaScript that targets GitHub and NPM tokens

Main Section

Lecture 22:
(Automated) Planning

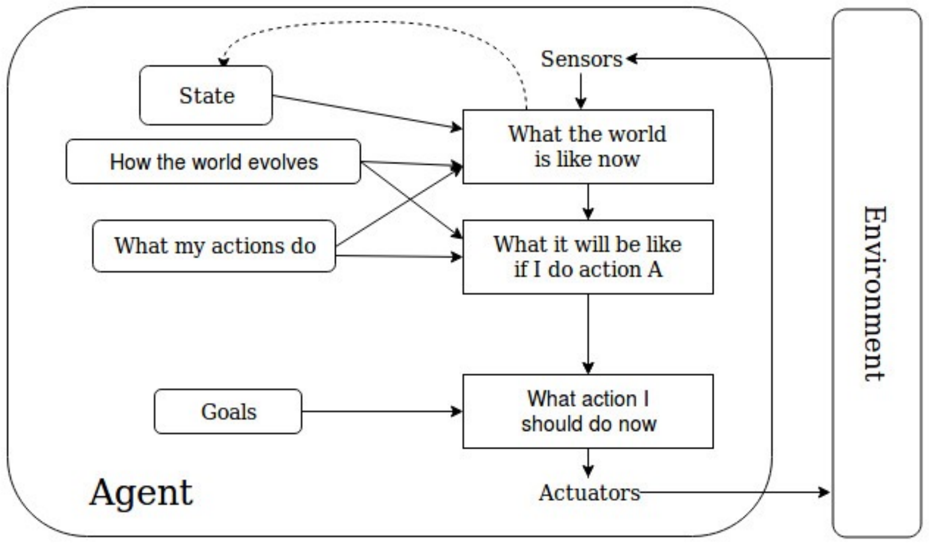
Complex Decisions

- Making a sequence of decisions
- Making a single decision but with
  - Environment changing
  - Actions not being deterministic
  - Perception not being perfect
  - …


(Dynamic)
(Partially Observable)
Environment
(Imperfect)
Perception
Goals
(Full vs. Partial satisfaction)
Action
(Deterministic vs. Stochastic)
(Instantaneous vs. Durative)

Goal-Based Agents
Generating Sequence of Actions

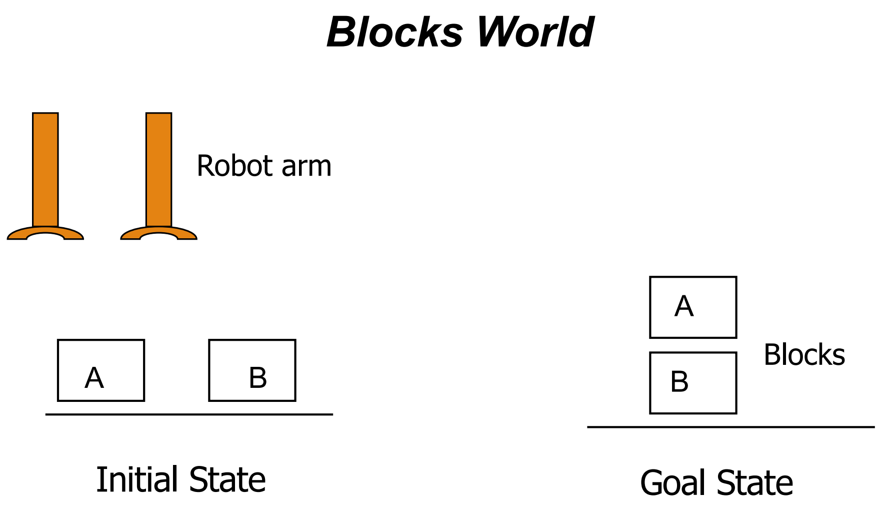
$$\begin{tikzcd}\text{Model} \arrow[r] & \boxed{\text{Planning}} \arrow[r] & \text{Plan}\end{tikzcd}$$



Sensors
State
What the world is like now
How the world evolves
What my actions do
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Figure Source: Russell & Norvig, AI: A Modern Approach

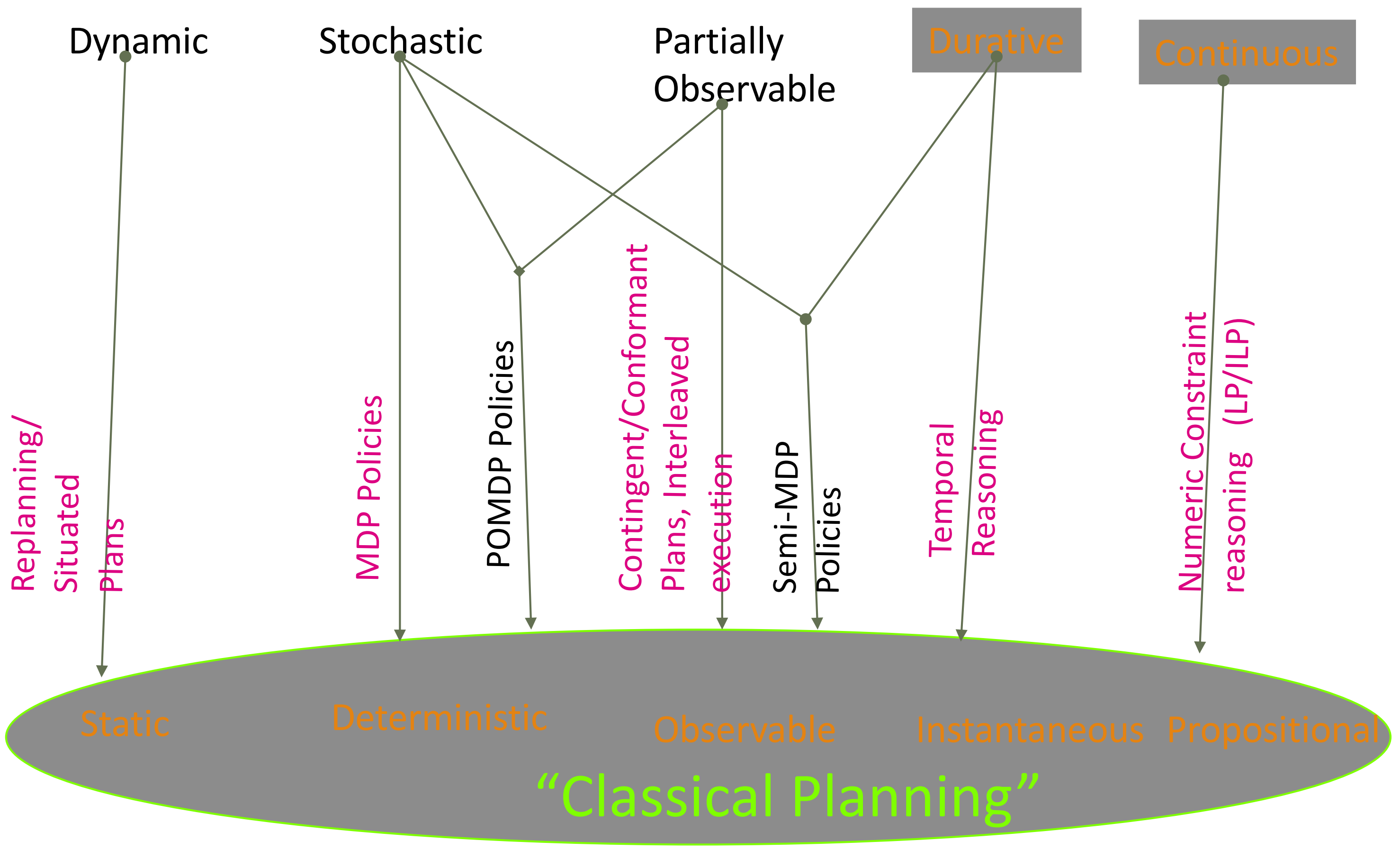
Reasoning Illustration - Planning Example



All robots are equivalent

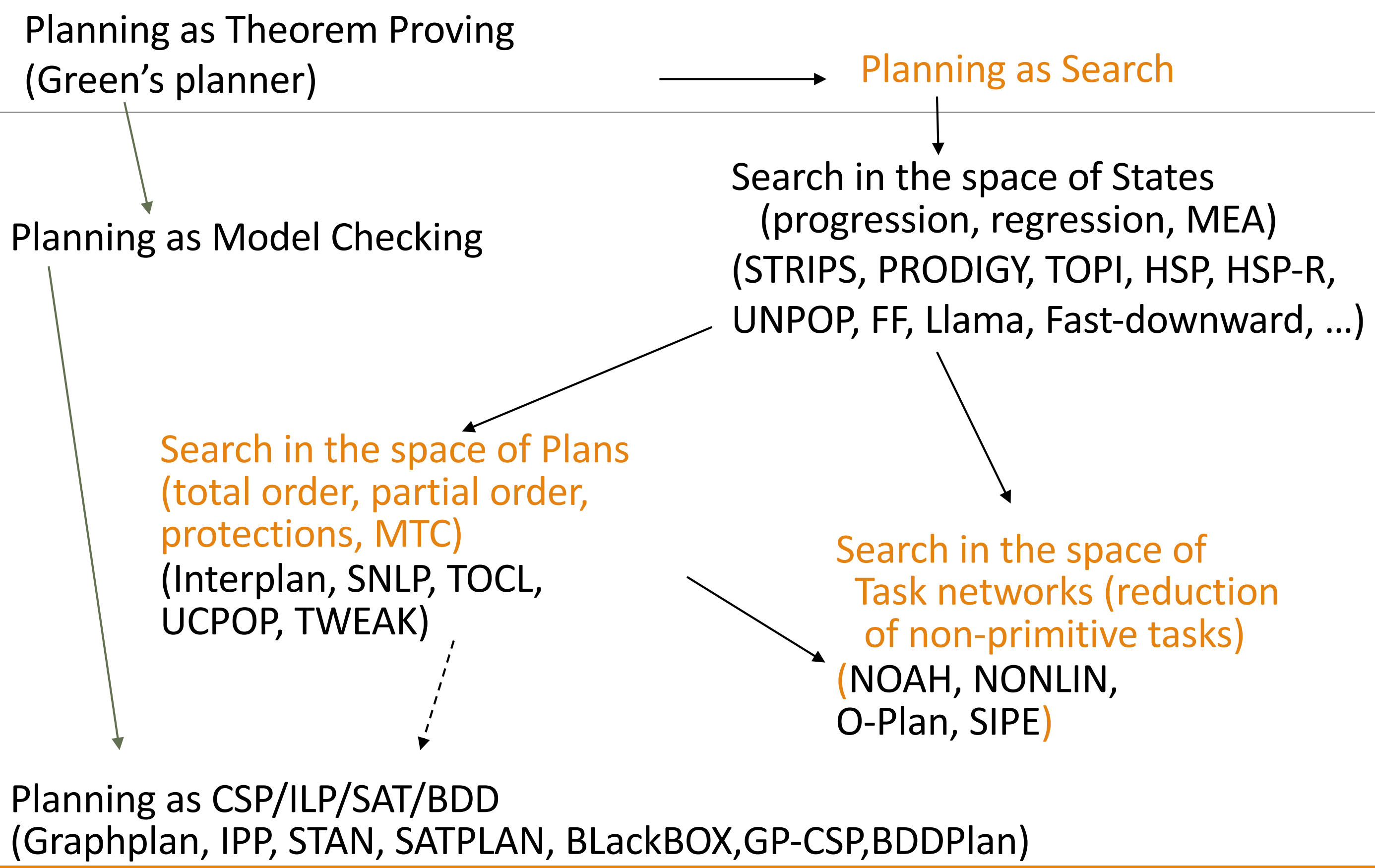
Planning Types: Procedural v/s Declarative v/s Utility

- Procedural
  - User: Tells **how** to do
  - System: does (executes) as is told to do
- Declarative
  - User: Tells **what** to do
  - System: finds **how** to do, then **does** (executes) as it finds to do
- Utility-based
  - User: Tells **what** is important
  - System: finds **what** to do, **how** to do, then **does** (executes) as it finds to do

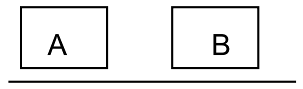


Credits: portions derived planning tutorial slides of Prof. Subbarao Kambhampati, Arizona State University

The (too) Many Brands of Classical Planners

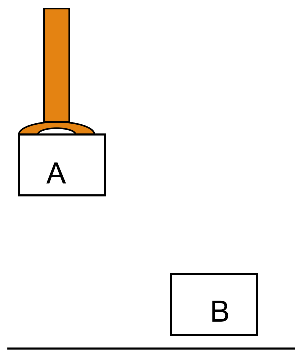

Planning as Theorem Proving
(Green’s planner)
Planning as Search
Search in the space of States
(progression, regression, MEA)
(STRIPS, PRODIGY, TOPI, HSP, HSP-R,
UNPOP, FF, Llama, Fast-downward, …)
Planning as Model Checking
Search in the space of Plans
(total order, partial order,
protections, MTC)
(Interplan, SNLP, TOCL,
UCPOP, TWEAK)
Search in the space of
Task networks (reduction
of non-primitive tasks)
(NOAH, NONLIN,
O-Plan, SIPE)
Planning as CSP/ILP/SAT/BDD
(Graphplan, IPP, STAN, SATPLAN, BLackBOX,GP-CSP,BDDPlan)

Reasoning Illustration - Representation

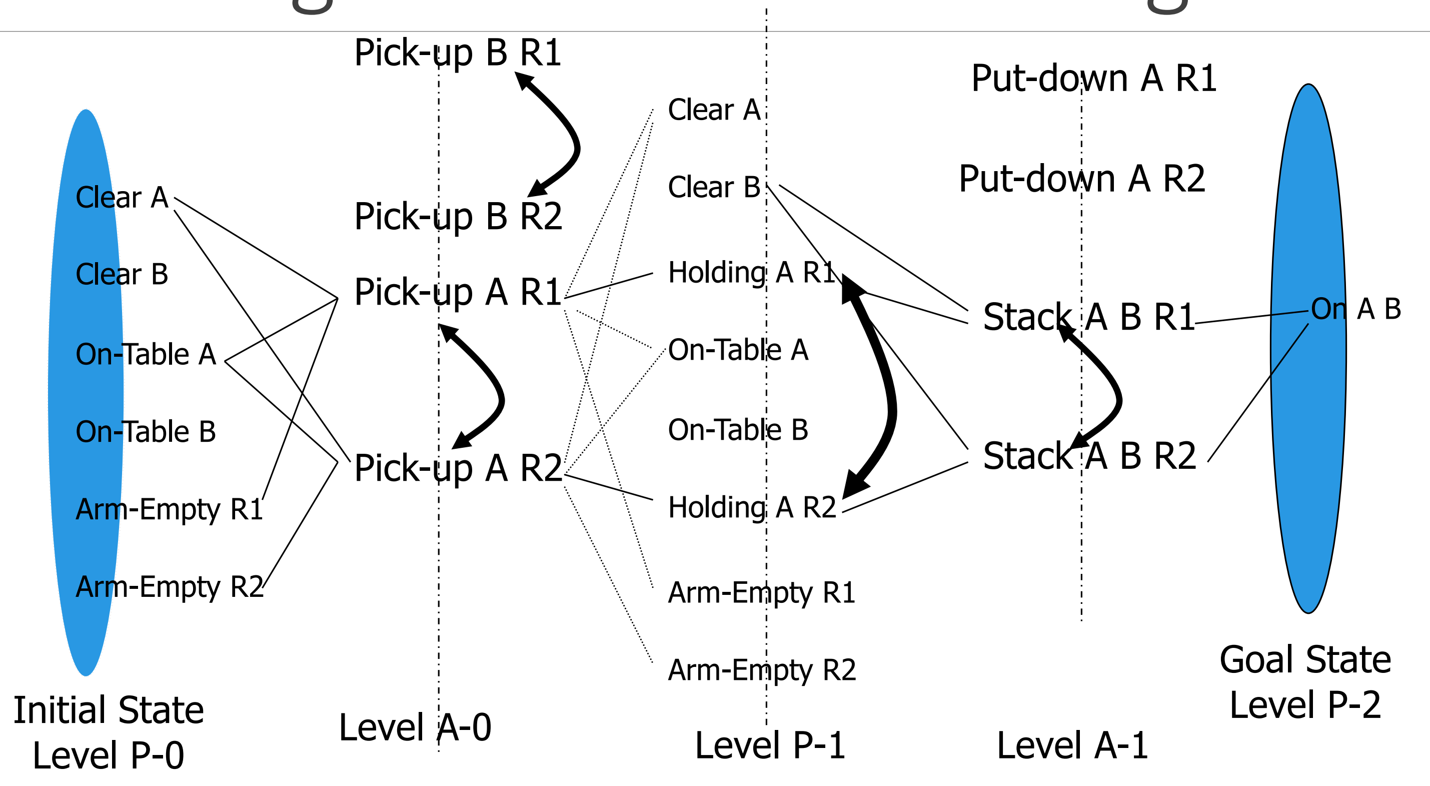


States: ((On-Table A) (On-Table B) …)

```
Actions: ((Name: (Pickup ?block ?robot)
          Precondition: ((Clear ?block)
                         (Arm-Empty ?robot)
                         (On-Table ?block))
          Add:          ((Holding ?block ?robot))
          Delete:       ((Clear ?block)
                         (Arm-Empty ?robot)))…)
```



Reasoning Illustration - Planning Process



Active Area of Research

Considerations : single-agent

- What to find:
  - **Any** workable plan
  - **Optimal** plan – but then what is the criteria
  - **All** plans
  - **Diverse** plans
- When to find (generate)
  - Plan **at the end**
  - Plan **anytime**
- How to
  - Represent problem
  - Find solution (search, learning, …)
  - Explain solution
- Applications

Considerations: multi-agent

- What to find:
  - **Any** workable plan
  - **Optimal** plan – but then what is the criteria
- When to find (generate)
  - Plan **at the end**
- How to
  - Represent problem
  - Find solution (search, learning, …)
  - Explain solution
- Applications
  - Traffic lights, automated cars
  - Warehouses

Hands On With Planning

- Coding examples: https://github.com/biplav-s/course-ai-f25/tree/main/sample-code/class22-planning
  - Api.planner
    - http://planning.domains/; Try the editor: https://editor.planning.domains/#
    - Code sample
  - Pyperplan
    - Demo

Exercise: 10 mins

- Try any domain from domain.pddl or classical planning repo: https://github.com/biplav-s/course-ai-f25/blob/main/sample-code/class22-planning/API%20Based%20Planner%20Invocation-PlanUtils.ipynb
- Change sample code with domain and problem files
- Run the sample code

Forms of Uncertainty and Planning

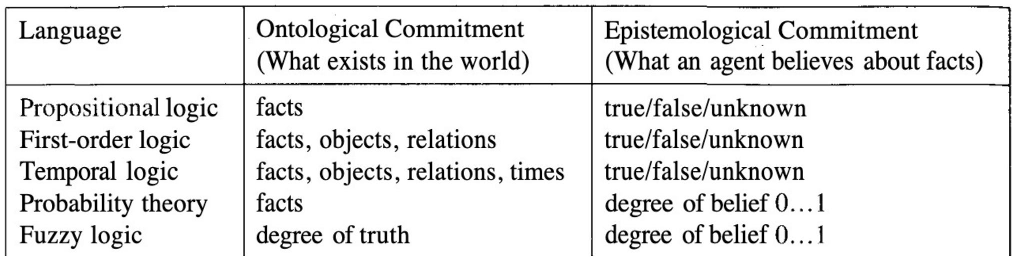
- Uncertain knowledge, caused by
  - Incomplete knowledge
  - Incorrect knowledge
- Uncertain actions, caused by
  - Physics of the domain
  - External events

Forms of Uncertainty

- Uncertain knowledge, caused by
  - Incomplete knowledge
  - Incorrect knowledge
- Uncertain actions, caused by
  - Physics of the domain
  - External events

Alternative approaches to represent

- Degree of belief: Probability. The sentence still is true or false
- Degree of truth: Fuzzy logic



Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0…1
Fuzzy logic	degree of truth	degree of belief 0…1

Credits:

- Russell & Norvig, AI - A Modern Approach
- Deepak Khemani - A First Course in AI

Forms of Uncertainty

- Uncertain knowledge, caused by
  - Incomplete knowledge
  - Incorrect knowledge
- Uncertain actions, caused by
  - Physics of the domain
  - External events

Use Probability Theory
Infer using probabilities

Decision Processes = create
situational policies (state-action based)

Decision-theoretic Agent

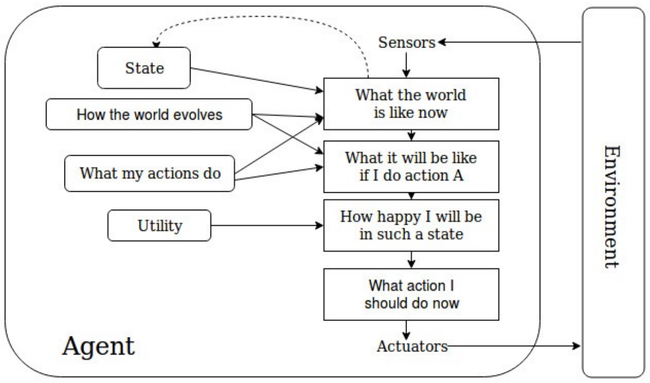
Probability theory: degree of belief in sentences

- Summarizes the uncertainty t

Utility theory: represent and reason with preferences

```
function DT-AGENT(percept) returns an action
  static: a set probabilistic beliefs about the state of the world

  calculate updated probabilities for current state based on
      available evidence including current percept and previous action
  calculate outcome probabilities for actions,
      given action descriptions and probabilities of current states
  select action with highest expected utility
      given probabilities of outcomes and utility information
  return action
```



Sensors
State
What the world is like now
How the world evolves
What it will be like if I do action A
What my actions do
Utility
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

Source: Russell & Norvig, AI - A Modern Approach

Lecture 22: Planning

- We talked about
  - Planning
  - Classical planning
  - Procedural / Declarative / Utility based planning
  - Planners and examples

Lecture 23:
Large Language Models and Planning

Lecture 24:
Quiz 3

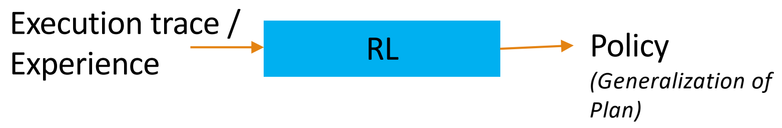
Lecture 25: SDP / RL

Lecture 23: Outline

We will discuss

- RL
- Inverse RL
- Multi-Arm Bandit problems

Reinforcement Learning

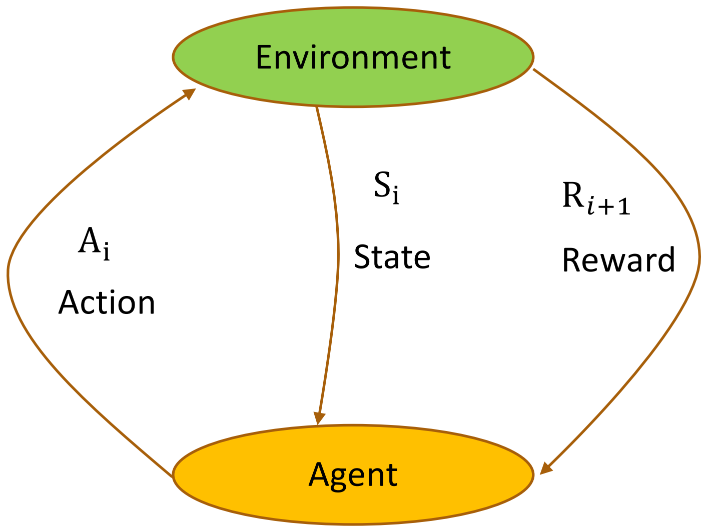


Reinforcement Learning Setting

- An agent in an environment
- Agent
  - Can see state
  - Can take action
  - Will get rewards

- Precisely, at each time step i
  - In state $\mathrm{S_i}$, agent takes action $\mathrm{A_i}$
  - Based on state s_i and action $\mathrm{a_i}$, the environment transitions to state $\mathrm{S_{i+1}}$ and outputs reward R_{i+1}

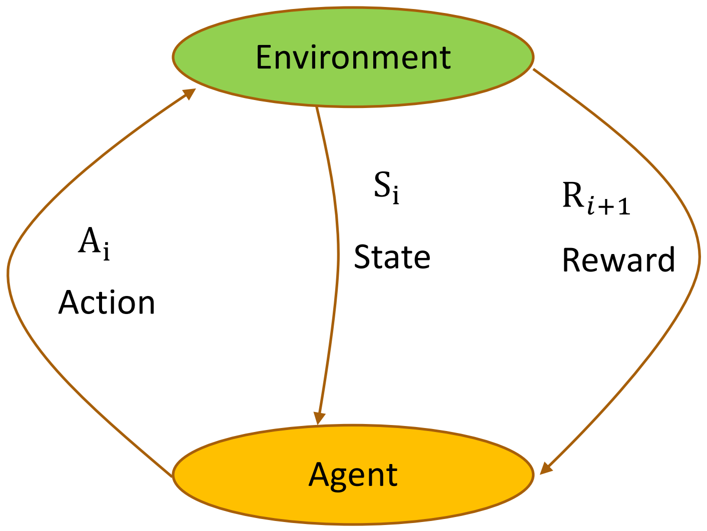
- **Objective**: learn mapping of states to actions so that the agent maximizes the reward from the environment.

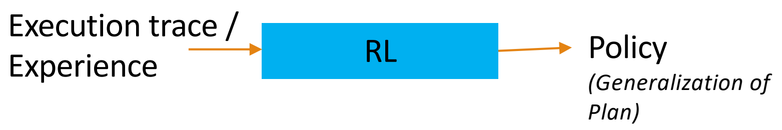


Reinforcement Learning

- **Objective**: learn mapping of states to actions so that the agent maximizes the reward from the environment.

- **Output**
  - Deterministic: $a = \pi(s)$
  - Stochastic: $\pi(a|s) = P(A_i = a|S_i = s)$





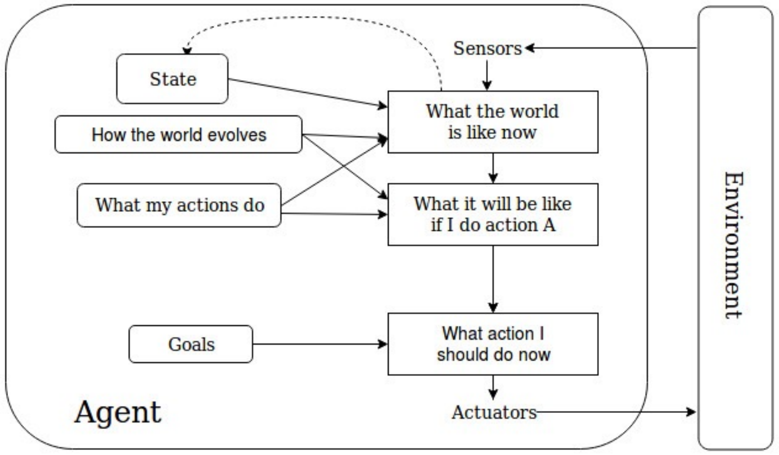
Comparison With Other Learning

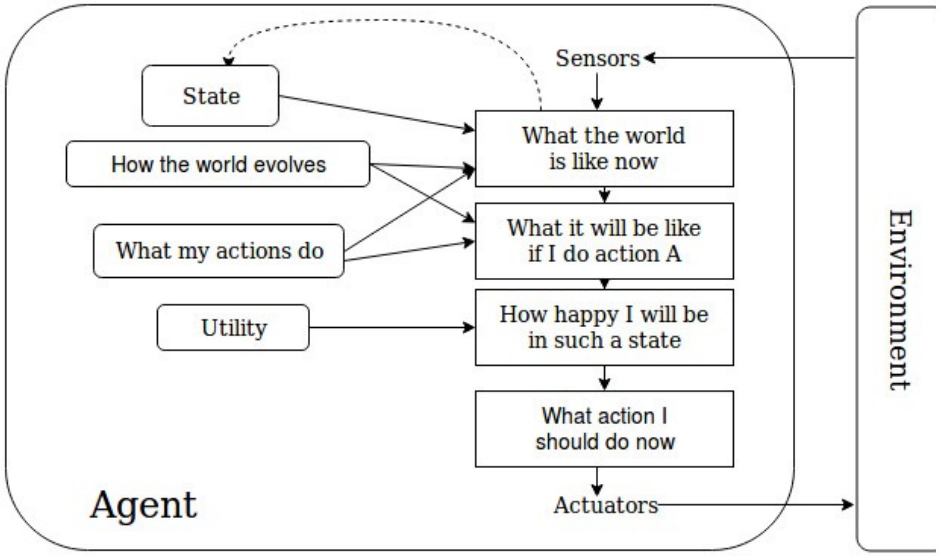
- Supervised learning
  - Training information: labels
  - Objective: learn (input-label) mapping
  - Goodness criteria: Reduce error = (Predicted label – Actual label)
- Reinforcement learning
  - Training information: reward functions
  - Objective: learn policy
  - Goodness criteria: maximal reward
- These two forms of learning are orthogonal – for different tasks

Image Source: Russell & Norvig, AI: A Modern Approach

RL as a Learning-Based Agent

A general, alternative way of solving goal-based problems from just execution traces


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Utility
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

Goal- and Utility-based Intelligent Agent

$$\text{Model} \rightarrow \boxed{\text{Planning}} \rightarrow \text{Policy}$$

(Generalization of plan)

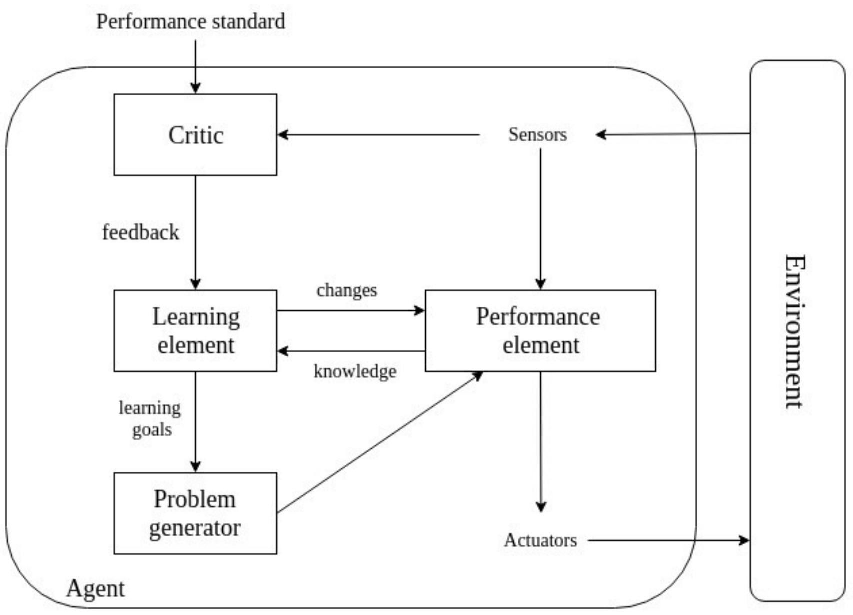
$$\text{Execution trace / Experience} \rightarrow \boxed{\text{RL}} \rightarrow \text{Policy}$$

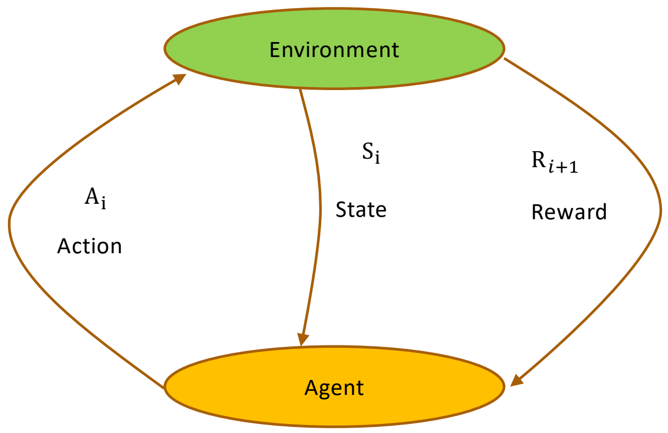
(Generalization of Plan)

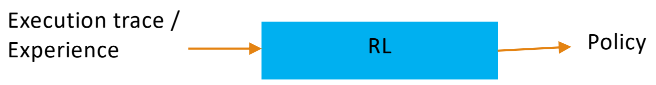
Image Source: Russell & Norvig, AI: A Modern Approach

RL as a Learning-Based Agent

A general, alternative way of solving goal-based problems from just execution traces







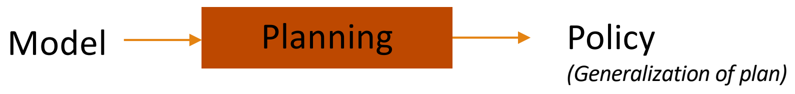


Image Source: Russell & Norvig, AI: A Modern Approach

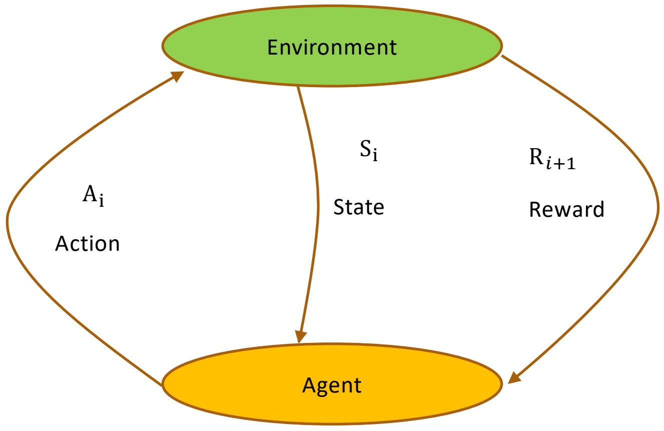
RL as a Learning-Based Agent

A general, alternative way of solving goal-based problems from just execution traces

Goal- and Utility-based Intelligent Agent

$$\begin{tikzcd}\text{Model} \arrow[r] & \boxed{\text{Planning}} \arrow[r] & \text{Policy}\end{tikzcd}$$

(Generalization of plan)



$$\begin{tikzcd}\text{Execution trace / Experience} \arrow[r] & \boxed{\text{RL}} \arrow[r] & \text{Policy}\end{tikzcd}$$

Exercise and Code – Gym RL

- RL using Open AI's Gym
  - https://gymnasium.farama.org/ (Old: https://gym.openai.com/)
  - Environments: https://gym.openai.com/envs/#classic_control
- Exercise (5 mins):
  - Look at the various categories
  - Explore the videos

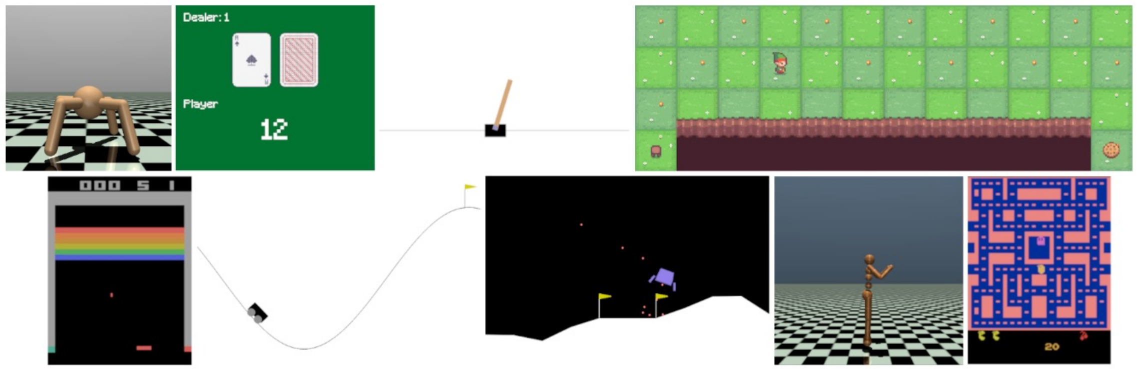


Figure 1: A subset of the available Reinforcement Learning Environments available through Gymnasium. From left to right, top to bottom: Ant-v5, Blackjack-v2, Cartpole-v1, FrozenLake-v1, ALE/Breakout-v5, MountainCar-v2, LunarLander-v3, Humanoid-v5, ALE/MsPacman-v5

Paper: Gymnasium: A Standard Interface for Reinforcement Learning Environments
https://arxiv.org/abs/2407.17032, Nov 2025

Exercise and Code – Gym RL

- RL using Open AI's Gym
  - https://gymnasium.farama.org/
  - Old: https://gym.openai.com/

- Code:
  - Latest: https://github.com/biplav-s/course-ai-f24/blob/main/sample-code/l25-rl/RL%20with%20Gym.ipynb
  - Old: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l18-learning-agent/RL%20using%20Gym.ipynb

Source: Russell & Norvig, AI: A Modern Approach

Diversity in RL Problems

- **Environment** - accessible or inaccessible
  - Accessible: states can be identified with percepts
  - Inaccessible environment: agent has to learn and maintain representation of state to track environment
- ***Knowledge*** of effects of action and utility, or learn
- **Rewards**
  - Available for all states or only terminal states
  - Actual utility or hints of increase/ decrease
- **Ability to execute actions** - Active learner or passive learner
  - A passive learner simply watches the world going by, and tries to learn the utility of being in various states
  - An active learner can take actions to explore unknown environment

Source: Russell & Norvig, AI - A Modern Approach

Passive RL

- **Input**
  - policy: π_i
  - // Has no knowledge Reward R(s) and Transition function P(s' |s, a)
- **Output**
  - Expected utility for each state, U(s)

- **Procedure**:
  - Execute a sequence of runs
  - At any instant, the agent knows only its current state and current reward, and the action it must take next. This action may lead it to more than one state, with different probabilities.
- **Expected Utility**

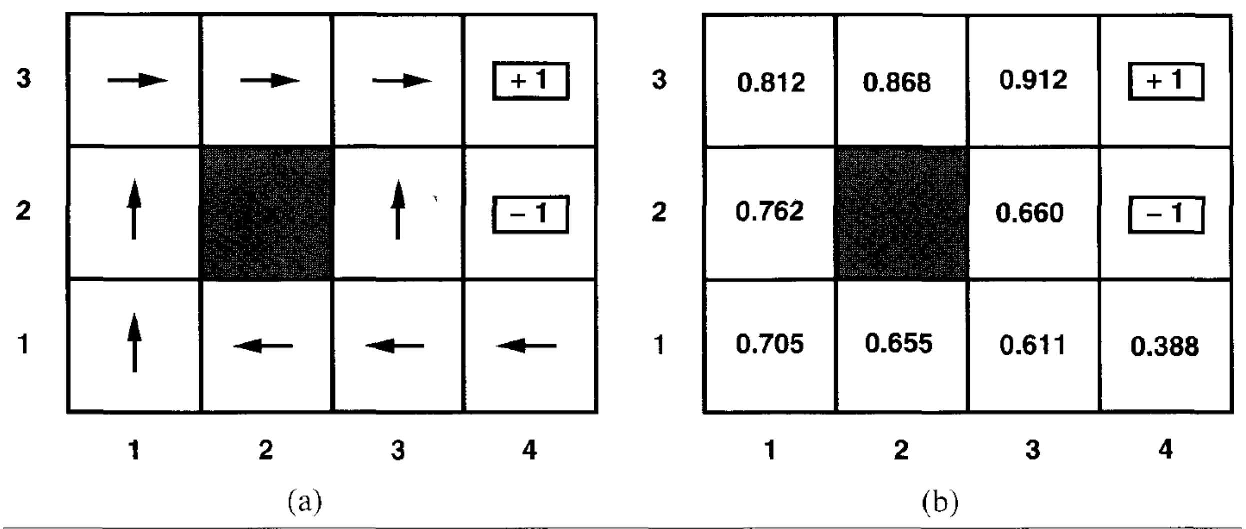
$$U^{\pi}(s) = E(\sum_{t=0}^{\inf} \gamma^t R^t(s')$$

Illustration

```
# Action Directions
north = (0, 1)
south = (0,-1)
west = (-1, 0)
east = (1, 0)

policy = {
    (0, 2): east,  (1, 2): east,  (2, 2): east,   (3, 2): None,
    (0, 1): north,                (2, 1): north,  (3, 1): None,
    (0, 0): north, (1, 0): west,  (2, 0): west,   (3, 0): west,
}
```

Policy: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l15-l16-l17-l18-agents/reinforcement_learning.ipynb



Input Policy and Output Optimal Utility

Source: Russell & Norvig, AI - A Modern Approach

The Markov Property – True of Many Domains

- **Our policy at timepoint t is only dependent on the current state s**
  - $\pi(a|s) = P(A_t = a|S_t = s)$
- Although the agent has a history up until S_t
  - $\mathrm{H_t} = S_0, A_0, R_1 S_1, A_1, R_2 \dots S_{t-1}, A_{t-1}, R_t, S_t$
- One may assume that all relevant information about the future is contained in the current state and action
  - $P(S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a) = P(S_{t+1} = s', R_{t+1} = r|H_t = h_{t+1}, A_t = a)$
- This is a generalization of the Markov property to sequential decision problems
  - $P(S_{t+1}|S_t) = P(S_{t+1}|S_t, S_{t-1}, \dots S_0)$

Source: Forest A.'s RL Course

RL with Finite States

Solving a Finite MDP

- **States:** A discrete and finite set $\mathcal{S}$
- **Actions:** A discrete and finite set $\mathcal{A}$
- **Transition Probabilities:** $P(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a)$
  - Defines the dynamics of the MDP
- The state-transition probabilities can be obtained from the transition probabilities
  - $p(s'|s,a) = \sum_{r \in \mathcal{R}} p(s', r|s,a)$ // Estimating state-transition by looking at reward of samples
- The **expected reward** can be obtained from the transition probabilities
  - $r(s, a) = \sum_{r \in \mathcal{R}} r \sum_{s' \in \mathcal{S}} p(s', r|s,a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$
    // Estimating reward from transitions seen

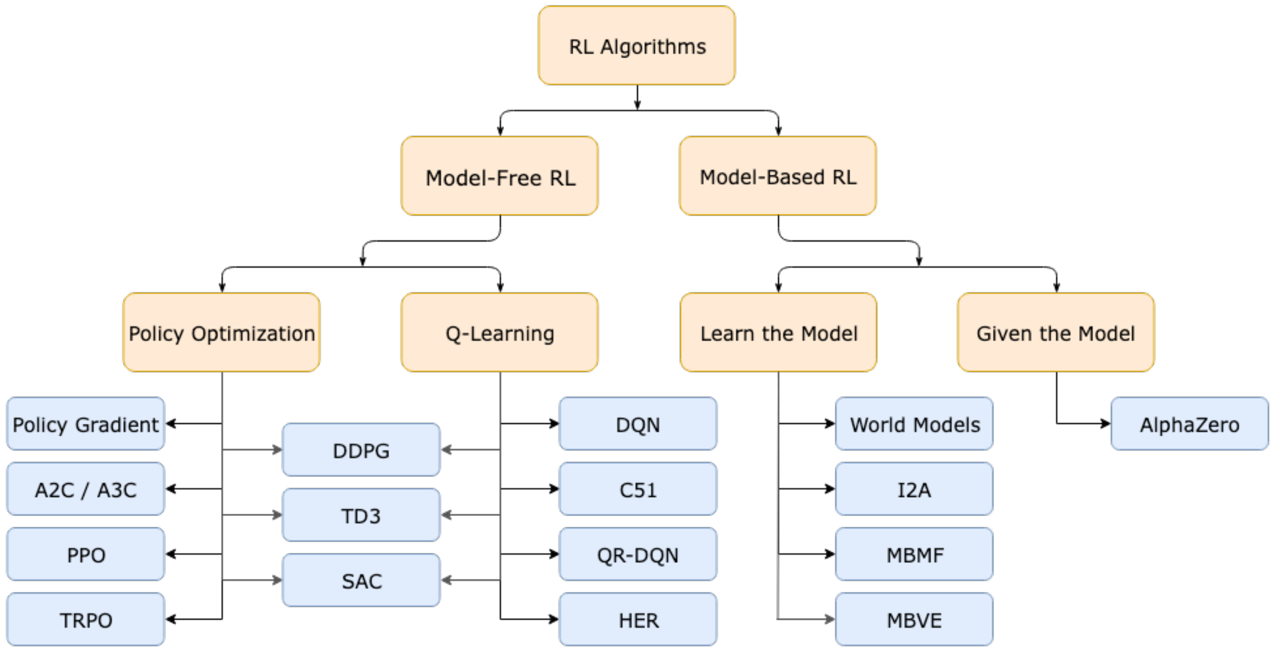
Adapted from: Forest A.'s RL Course

Model-free and Model-based RL

- **Model-free** - the agent learns action to take, but nothing more.
  - Hence, it learns action to take directly from experience without building a model of the environment.
  - Methods: Q-Learning, Policy Gradient
- **Model-based -** the agent learns some aspect of the environment (“model”) along with the action to take
  - Model includes state transitions probabilities (what follows if an action is taken), distances between states
  - Agent tries to understand its environment and creates a model for it based on its interactions with this environment.
  - Can readapt quickly if reward-scheme changes, plan future action paths, update Q-values by reasoning on model (without taking action)
  - Methods: Model predictive control (MPC)
- Both approaches can get policy generated under different assumptions. See example for both on a tennis game in [3]

Sources:
1. https://www.youtube.com/watch?v=mFyqU2dWQE0&t=114s
2. https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html
3. https://neptune.ai/blog/model-based-and-model-free-reinforcement-learning-pytennis-case-study

Model-free and Model-based RL



1. A2C / A3C (Asynchronous Advantage Actor-Critic): Mnih et al, 2016
2. PPO (Proximal Policy Optimization): Schulman et al, 2017
3. TRPO (Trust Region Policy Optimization): Schulman et al, 2015
4. DDPG (Deep Deterministic Policy Gradient): Lillicrap et al, 2015
5. TD3 (Twin Delayed DDPG): Fujimoto et al, 2018
6. SAC (Soft Actor-Critic): Haarnoja et al, 2018
7. DQN (Deep Q-Networks): Mnih et al, 2013
8. C51 (Categorical 51-Atom DQN): Bellemare et al, 2017
9. QR-DQN (Quantile Regression DQN): Dabney et al, 2017
10. HER (Hindsight Experience Replay): Andrychowicz et al, 2017
11. World Models: Ha and Schmidhuber, 2018
12. I2A (Imagination-Augmented Agents): Weber et al, 2017
13. MBMF (Model-Based RL with Model-Free Fine-Tuning): Nagabandi et al, 2017
14. MBVE (Model-Based Value Expansion): Feinberg et al, 2018
15. AlphaZero: Silver et al, 2017

Credit: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Model-free and Model-based RL

Feature	**Model-Free RL**	**Model-Based RL**
Sample efficiency	Lower (needs more real-world interactions)	Higher (can use simulations for more learning)
Complexity	Simpler to implement	More complex due to model learning
Environment type	Better for complex, hard-to-model environments	Better for environments where interactions are costly or slow (e.g., robotics)
Flexibility	Requires retraining when dynamics shift	Can adapt by modifying the model without full retraining

Model-free RL: Q-learning

- Learning action-value functions
- Q(a,i): value of doing action a in state i
- Relationship between **utility U of state and Q value**
  - U(i) = max Q(a, i)
- Finding Q value based on whether transition probability is known
  - When M (transition is known)

$$Q(a,i) = R(i) + \sum_{j} M_{ij}^{a} \max_{a'} \; Q(a',j)$$

  - Estimating with TD method

$$Q(a,i) \leftarrow Q(a,i) + a\,(R(i) + \max_{a'} Q(a',j) - Q(a,i))$$

Source: Russell & Norvig, AI - A Modern Approach

RL with Deep Learning

- For small problems, like games, state-value function (U), action- utility value (Q), and transition functions (M), and policy functions are represented using a table

- But for large and realistic problems, number of states are countably large/ practically infinite
- Deep learning are excellent function approximators
  - Estimate Q-value i.e., action-value

- Not covered in this class

Exercise and Code – RL

- RL settings and solution methods

- Code: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l18-learning-agent/RL%20using%20Gym.ipynb

Source: Russell & Norvig, AI: A Modern Approach

Inverse Reinforcement Learning

- Given π*and transition function M,
  - can we recover R

- Or, given execution traces corresponding to π*
  - can we recover R?

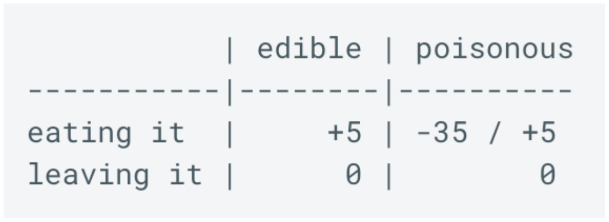
- Applications
  - Path planning
  - Automated-driving

- Reference: Pieter Abbel's course slides: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa12/slides/inverseRL.pdf

More RL – Multi-Arm Bandits

- A decision maker iteratively selects one of multiple fixed choices (i.e., arms or actions) when the properties of each choice are only partially known at the time of allocation, and may become better understood as time passes.
- Used for
  - Recommendations
  - clinical trials investigating the effects of different experimental treatments while minimizing patient losses
  - adaptive routing efforts for minimizing delays in a network,
  - financial portfolio design

Credits: https://en.wikipedia.org/wiki/Multi-armed_bandit
https://www.tensorflow.org/agents/tutorials/intro_bandit



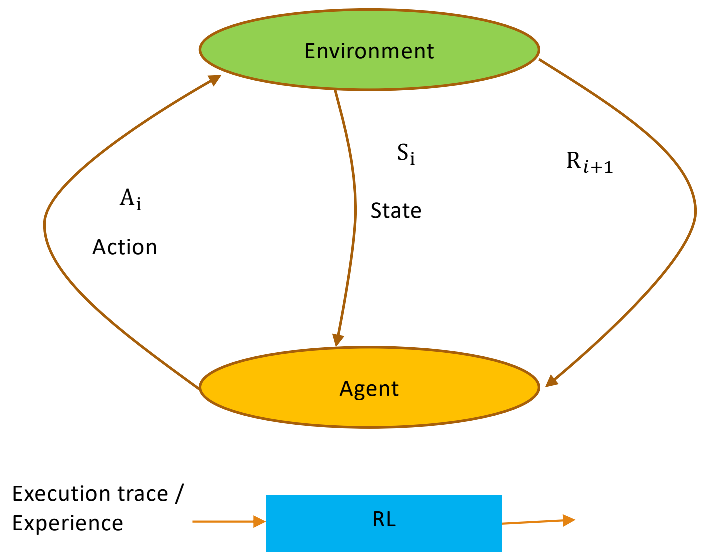
	edible	poisonous
eating it	+5	-35 / +5
leaving it	0	0

Mushroom recommendation problem

RL and MA-Bandit Connection

- In RL case, the next observation depends on the previous state and the action taken by the policy. This last part is what separates MAB from RL
- In MAB, the **next state**, which is the observation, **does not depend on the action chosen by the agent**.

- Process
  - An **environment** outputs observations, and responds to actions with rewards.
  - A **policy** outputs an action based on an observation, and
  - An **agent** repeatedly updates the policy based on previous observation-action-reward tuples.



Credits: https://www.tensorflow.org/agents/tutorials/intro_bandit

RL References

- Sutton and Barto's Book: http://incompleteideas.net/book/the-book.html
- Russell and Norvig, AI – A modern Approach
- David Silver's RL course, https://www.davidsilver.uk/teaching/

- Inverse RL
  - A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress, https://arxiv.org/abs/1806.06877, 2018
  - Pieter Abbel's course slides: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa12/slides/inverseRL.pdf

Shielded RL

Safe reinforcement learning has three categories:

- shaping (“engineering”) the reward function to encourage the agent to choose safe actions,[25]
- adding a second cost function (“constraining”),[30] and
- blocking (“shielding”) unsafe actions at runtime.[24]
  - Shielding—provides formal safety guarantees

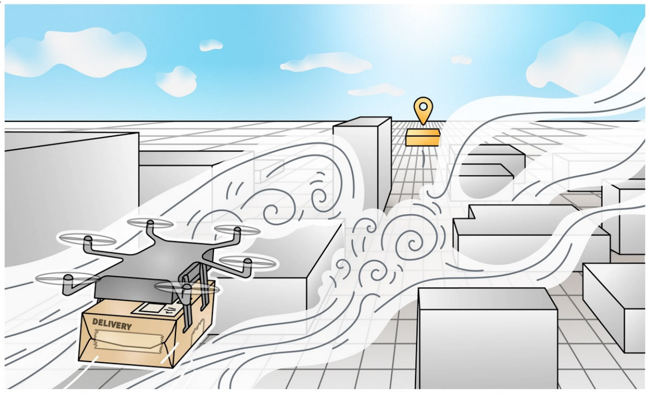


Figure 1. The agent operates an unmanned aerial vehicle (UAV) tasked with delivering a package without any collisions. Factors such as wind and other aerial vehicles add complexity to this mission.

Credit: Bettina Könighofer, Roderick Bloem, Nils Jansen, Sebastian Junges, and Stefan Pranger. 2025. Shields for Safe Reinforcement Learning. Commun. ACM 68, 11 (November 2025), 80–90. https://doi.org/10.1145/3715958

Shielded RL

Core questions of shielded RL:

- What types of *safety guarantees* can be provided by a shield, and under which *assumptions?*
- How can shields be *computed*?
- How can shields be *integrated* in RL?
- What are the *challenges* in shielded RL?

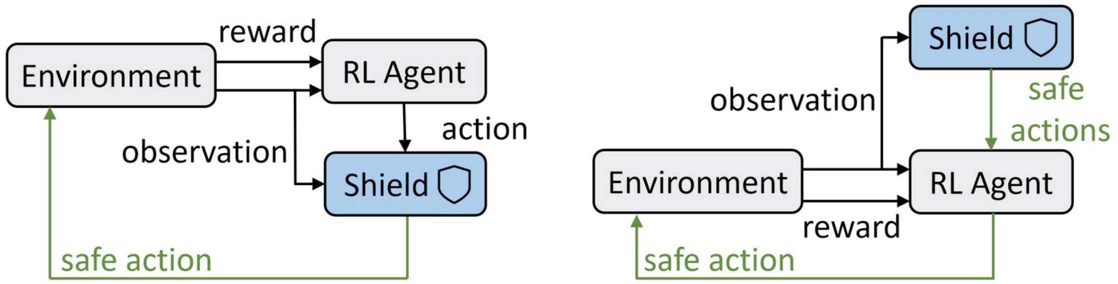


Figure 3. (Left) Post-shielding: The shield prevents unsafe actions from being executed. (Right) Pre-shielding: The shield restricts the choices of the agent.

Credit: Bettina Könighofer, Roderick Bloem, Nils Jansen, Sebastian Junges, and Stefan Pranger. 2025. Shields for Safe Reinforcement Learning. Commun. ACM 68, 11 (November 2025), 80–90. https://doi.org/10.1145/3715958

Shielded RL

Solving under different assumptions

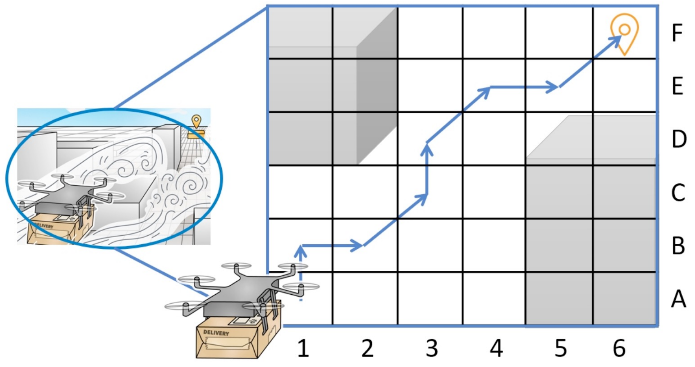


Figure 4. A snippet of the discrete model for the UAV example. The gray-shaded squares represent buildings. The UAV can move from cell to cell. The blue arrows indicate one possible path to the target location.

Credit: Bettina Könighofer, Roderick Bloem, Nils Jansen, Sebastian Junges, and Stefan Pranger. 2025. Shields for Safe Reinforcement Learning. Commun. ACM 68, 11 (November 2025), 80–90. https://doi.org/10.1145/3715958

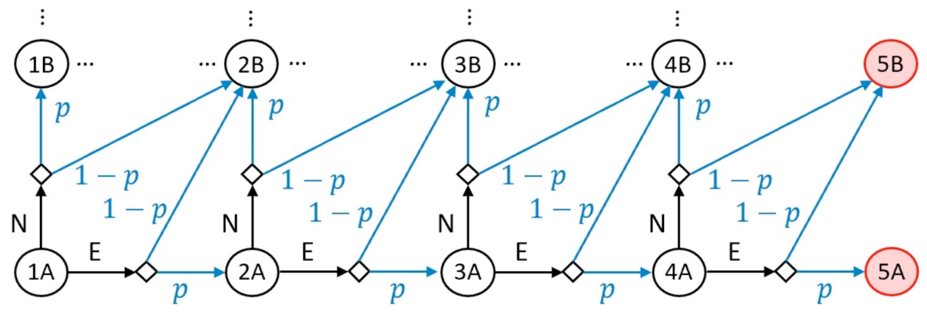


Figure 5. Markov decision process of the environment depicted in Figure 4.

MDP models represent the environment probabilistically.

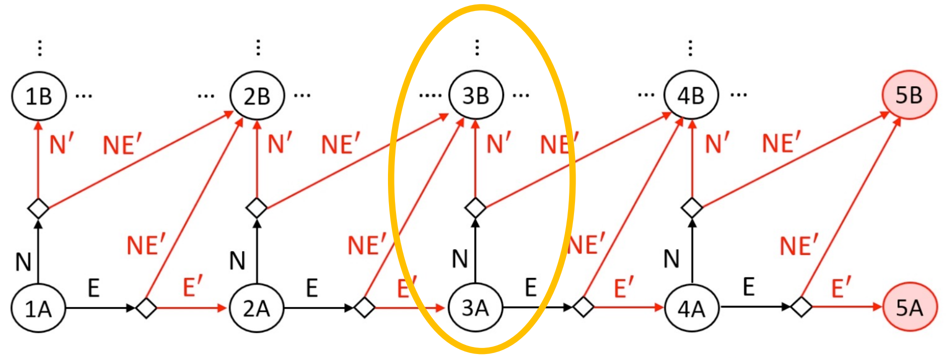


Figure 6. Two-player game representing the adversarial view of the MDP in Figure 5.

The agent-player first selects an action, followed by the environment choosing its action. These decisions determine the subsequent state. For instance, when the agent selects Action N from Cell 3A, the environment can either move the UAV to Cell 3B by choosing Action N′ or to Cell 4B by selecting Action NE′.

Lecture 22-25: Summary

- We talked about
  - Planning
  - Uncertainty
  - LLMs and Planning
  - Quiz 3
  - Reinforcement Learning

Go through the reading list on the topic: https://github.com/biplav-s/course-ai-f25/blob/main/reading-list/Readme-SDP.md

Weeks 12-13: Concluding Comments

We talked about

- Lecture 22: Planning
- Lecture 23: LLM and Planning
- Lecture 24: Quiz 3
- Lecture 25: RL

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 3: Machine Learning – Supervised (Classification)
- Week 4: Machine Learning - Unsupervised (Clustering) –
- Topic 5: Learning neural network, deep learning, Adversarial attacks
- Week 6: Large Language Models – Representation and Usage issues
- Weeks 7-8: Search, Heuristics - Decision Making
- Week 9: Constraints, Optimization – Decision Making
- Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
- Week 13: Trustworthy Decision Making: Explanation, AI testing
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Projects B: Sep 30 – Nov 20 (7 weeks; 400 points)

- End date: **Thursday, Nov 20**
  - Remember to update spreadsheet on data/ time when finished **(Column I)**

- Choices
  - Given by instructor
  - Defined by student using project-b teamplate; reviewed and approved by instructor

Upcoming Evaluation Milestones

- Projects B: Sep 30 – Nov 20
- Quiz 2: Oct 7
- Quiz 3: Nov 11
- Paper presentation (grad students only) : Nov 18
- Finals: Dec 11

About Week 14 – Lectures 26, 27

Week 13 – Lectures 26, 27

- Lecture 26: Graduate paper presentations
- Lecture 27: AI for the Real World – Bringing All Together; Advanced Topics

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2: Data: Formats, Representation, ML Basics
- Week 3: Machine Learning – Supervised (Classification)
- Week 4: Machine Learning - Unsupervised (Clustering) –
- Topic 5: Learning neural network, deep learning, Adversarial attacks
- Week 6: Large Language Models – Representation and Usage issues
- Weeks 7-8: Search, Heuristics - Decision Making
- Week 9: Constraints, Optimization – Decision Making
- Topic 10: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 11-12: Planning, Reinforcement Learning – Sequential decision making
- Week 13: Trustworthy Decision Making: Explanation, AI testing
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Note: exact schedule changes slightly to accommodate for exams and holidays.