CSCE 580: Introduction to AI

Lecture 24-25: Planning and Reinforcement Learning

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

14TH AND 19TH NOV, 2024

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

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

Organization of Lectures 24, 25

- Introduction Segment
  - Recap of Lectures 22 and 23
- Main Segment
  - Making Sequential Decisions
  - Planning
  - Reinforcement Learning
- Concluding Segment
  - Course Project Discussion
  - About Next Lecture – Lecture 26
  - Ask me anything

Introduction Section

Recap of Lecture 22 and 23

- Topic discussed
  - Making Decisions
  - Simple Decisions
  - Complex Decisions
    - MDPs
    - Prisoner's dilemma
    - Stable Marriage

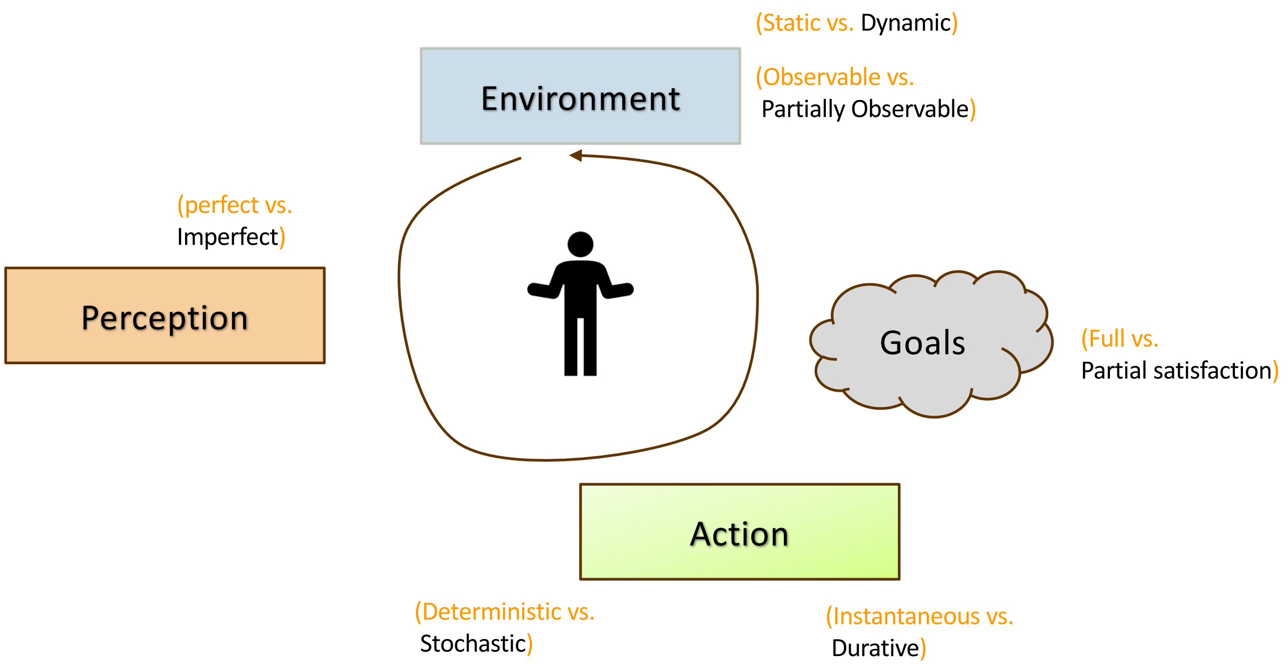
Graduate Paper Presentation

- Papers between 2022-2024 (last 4 years)
- At top AI venues: AAAI, Neurips, IJCAI, ICML, ICLR, **or discuss with instructor**

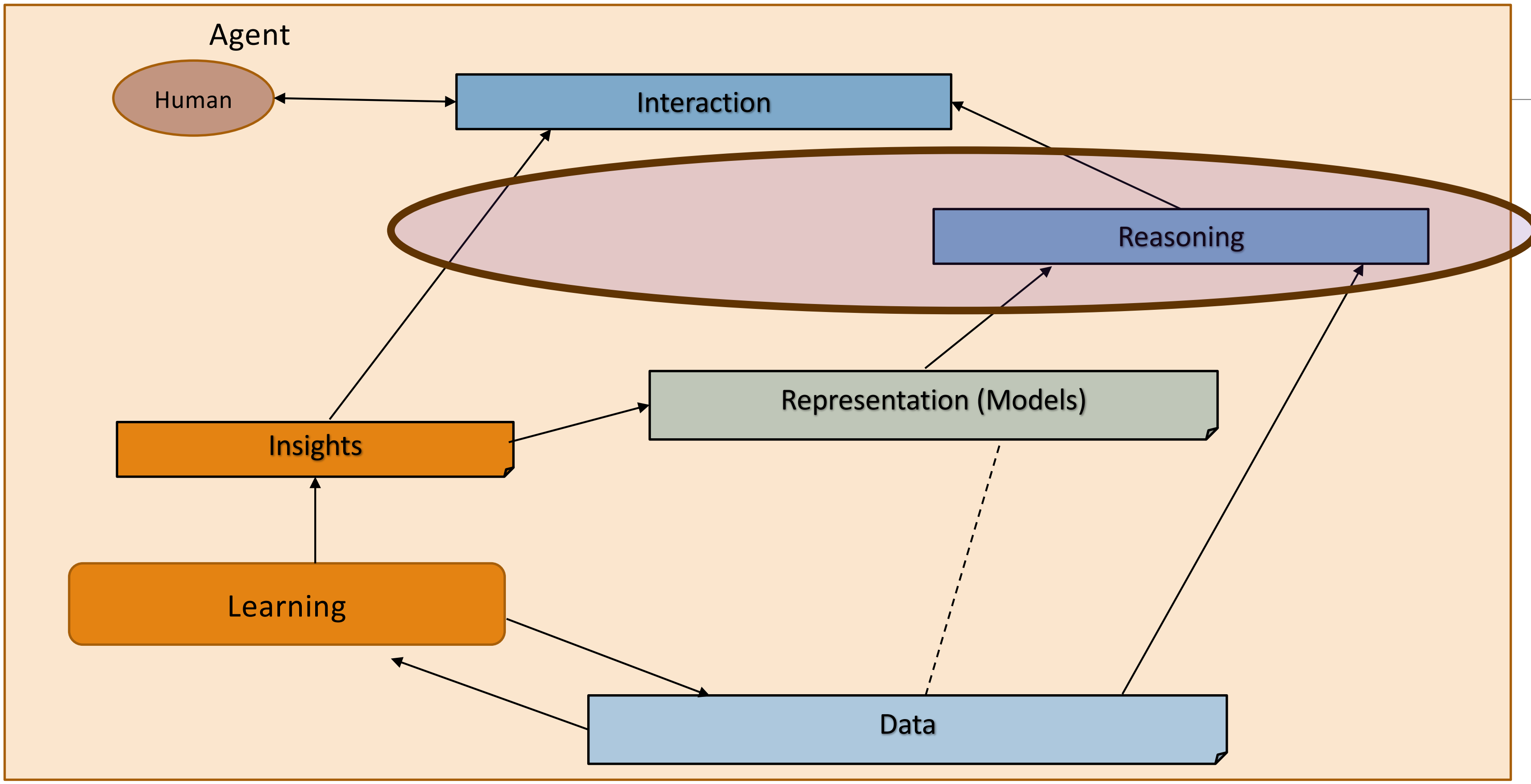
- Guideline on presentation – Nov 21, 2024 [Undergrads to attend]
  - Summary of the paper: problem, solution, related work, evaluation, contributions
  - Critique (+ves/ -ves)
  - Relevance to your and anyone else's project in the class

- Guidelines on a writeup
  - Verbalization of the presentation with three parts: summary, critique and relevance to class projects
  - A running example (from the paper or your own)

Intelligent Agent Model



Relationship Between Main AI Topics



Where We Are in the Course

CSCE 580/ 581 – In This Course

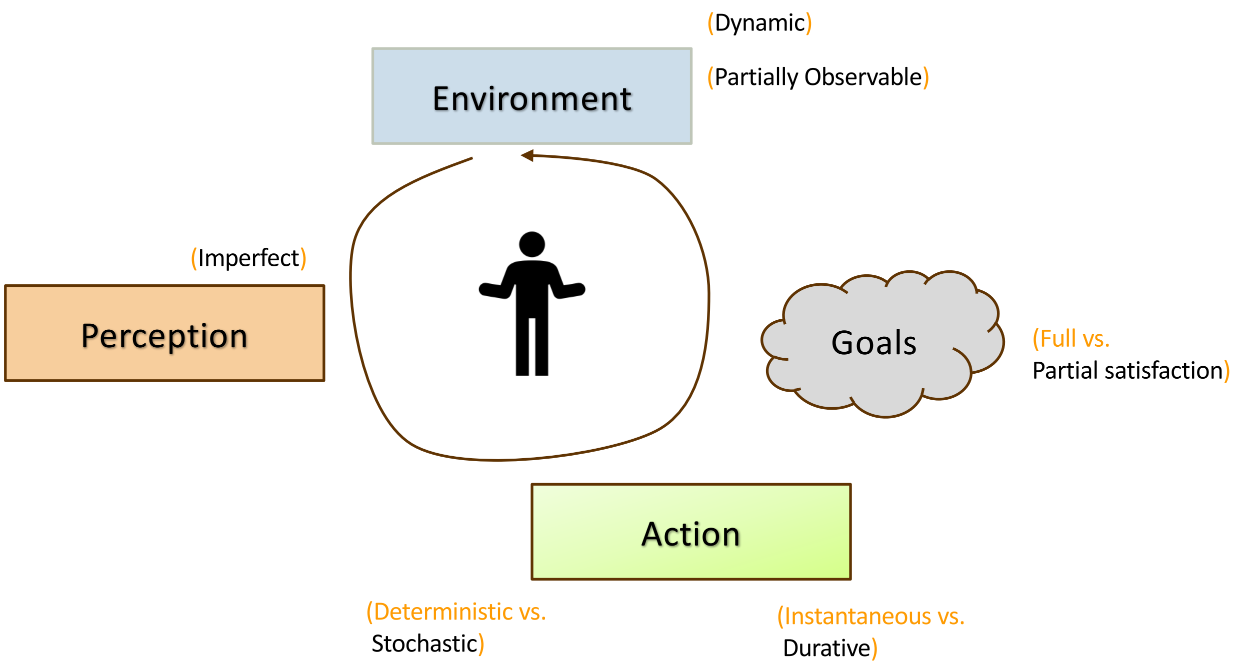
- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 4-5: Search, Heuristics - Decision Making
- Week 6: Constraints, Optimization – Decision Making
- Week 7: Classical Machine Learning – Decision Making, Explanation
- Week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

Main Section

Credit: Retrieved from internet

Complex Decisions

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



Goal-Based Agents
Generating Sequence of Actions

Model → Planning → Plan

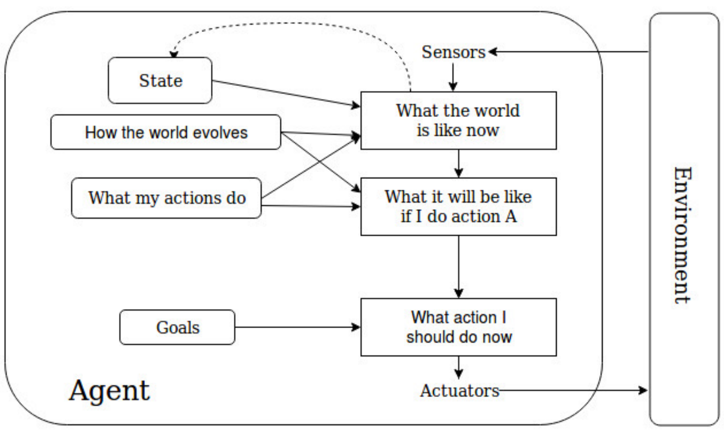
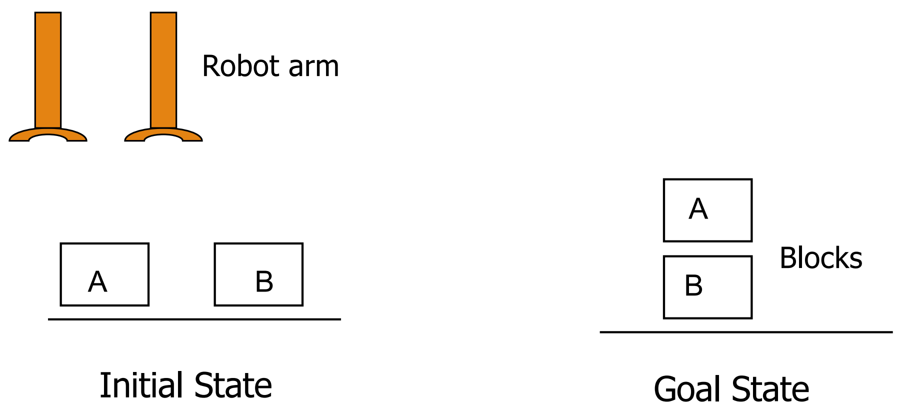


Figure Source: Russell & Norvig, AI: A Modern Approach

Reasoning Illustration - Planning Example

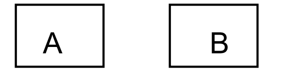
Blocks World



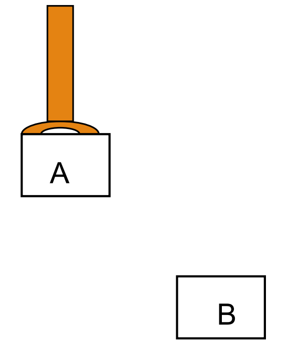
All robots are equivalent

Reasoning Illustration - Representation

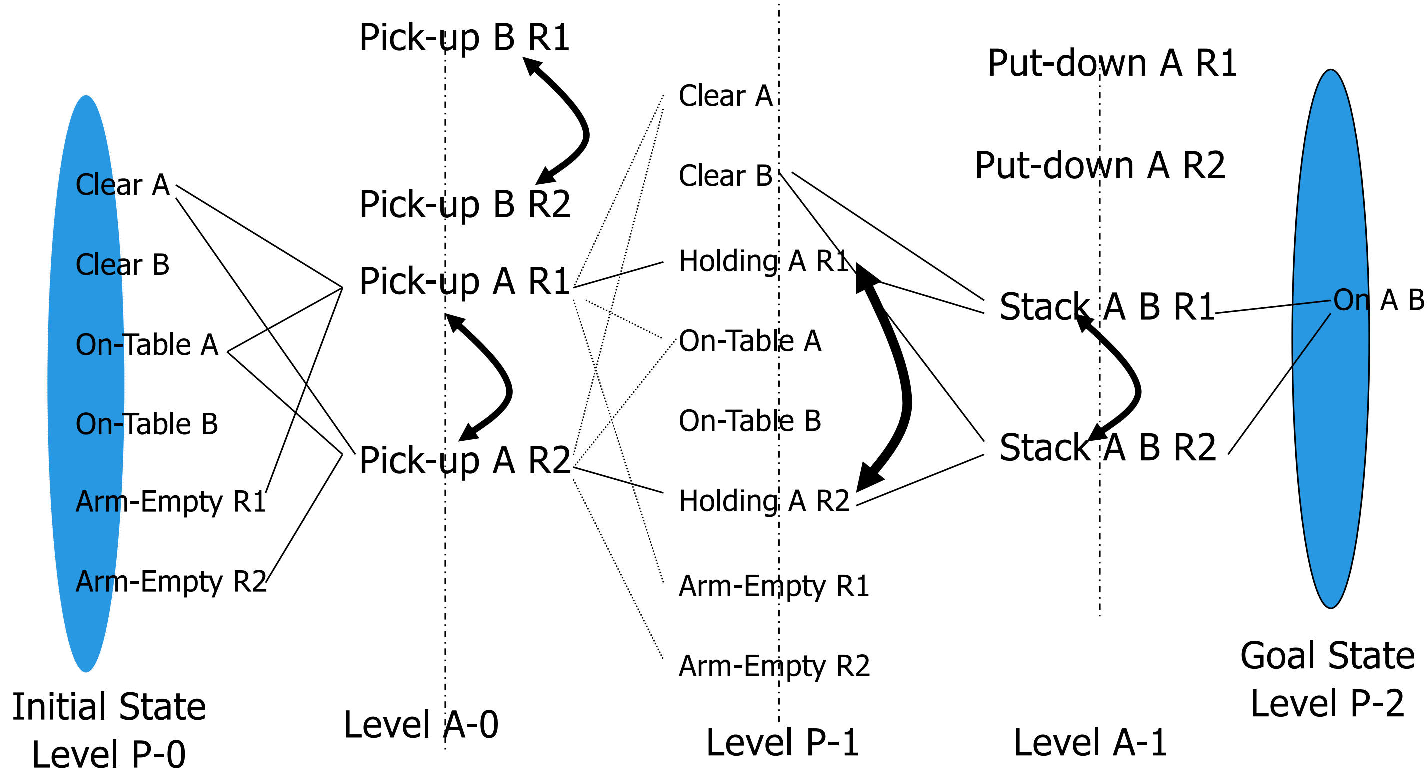
States: ((On-Table A) (On-Table B) …)



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

- What to find:
  - Any workable plan
  - Optimal plan – but then what is the criteria
  - All plans
  - Diverse plans
- How to find
  - Plan at the end
  - Plan anytime
- How to represent problem
- How to explain solution

Hands On With Planning

- Site: http://planning.domains/
  - Try the editor: https://editor.planning.domains/#

- Code example with API: https://github.com/biplav-s/course-ai-tai-f23/blob/main/sample-code/Class25-Planning/PlannerInvokerWithAPIs.ipynb

Exercise: 10 mins

- Try any domain from domain.pddl or classical planning repo: https://github.com/AI-Planning/classical-domains/tree/main/classical
- 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 $0 \ldots 1$
Fuzzy logic	degree of truth	degree of belief $0 \ldots 1$

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

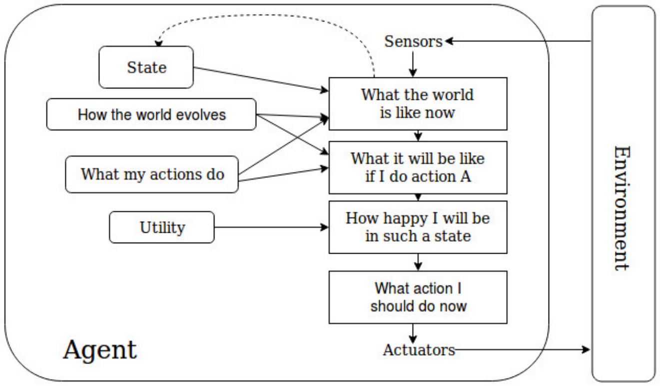
Decision Processes = create
situational policies (state-action based)

Decision-theoretic Agent

Probability theory: degree of belief in sentences

- Summarizes the uncertainty t

Utility theory: represent and reason with preferences

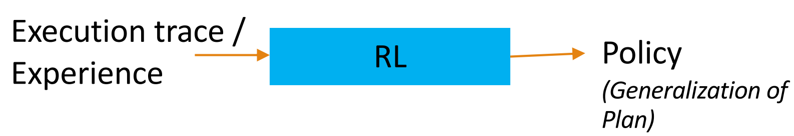


function DT-AGENT(*percept*)**returns** an *action*
  static: a set probabilistic beliefs about the state of the world

  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*

Source: Russell & Norvig, AI - A Modern Approach

Reinforcement Learning

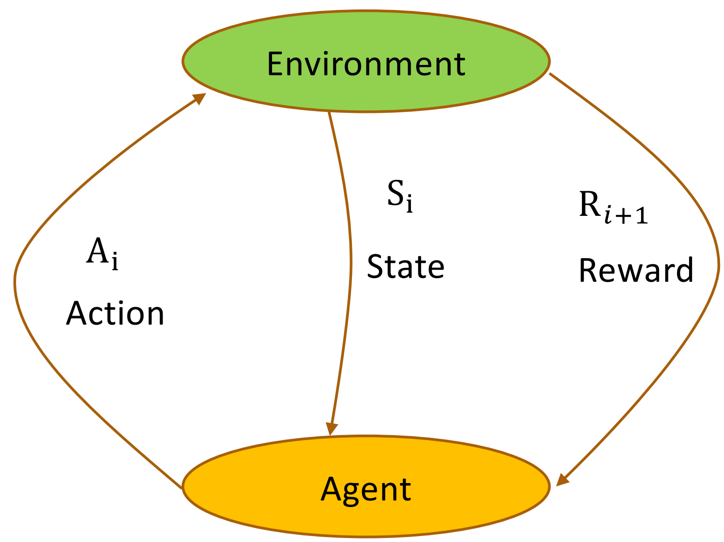


Reinforcement Learning Setting

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

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

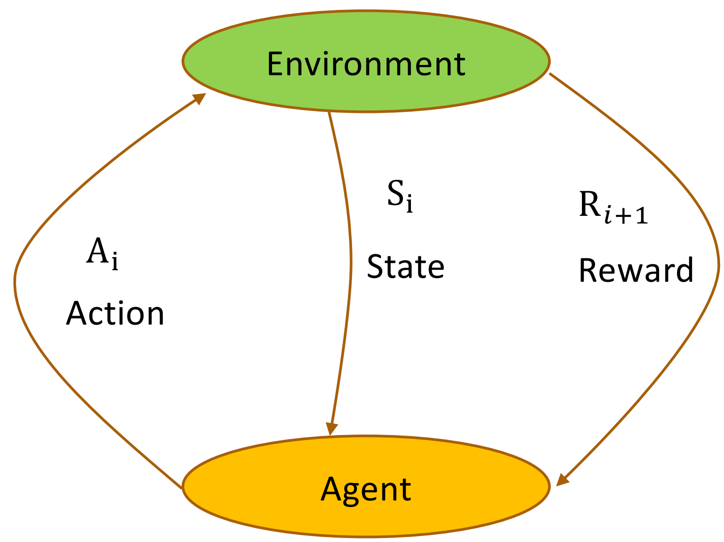
- **Objective**: learn mapping of states to actions so that the agent maximizes the reward from the environment.



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: $\pi(a|s) = P(A_i = a|S_i = s)$



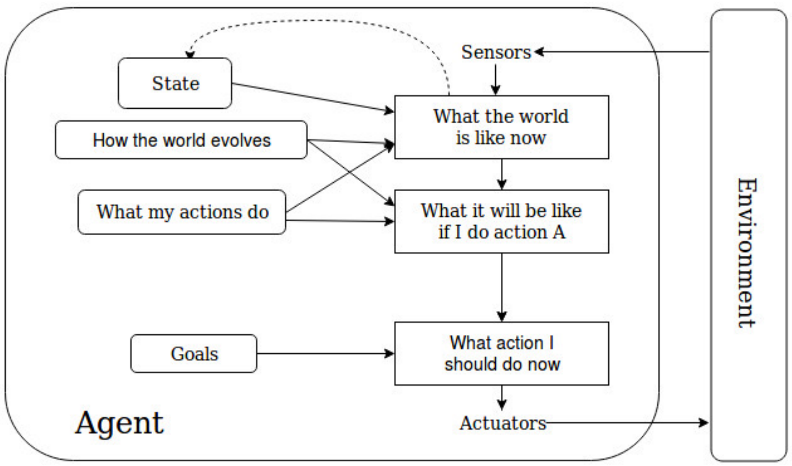
Comparison With Other Learning

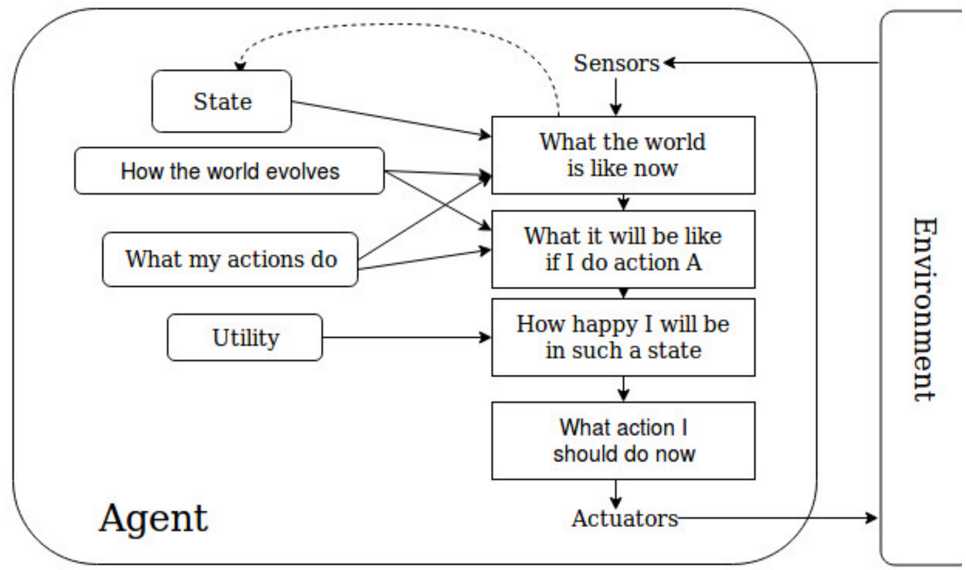
- Supervised learning
  - Training information: labels
  - Objective: learn (input-label) mapping
  - Goodness criteria: Reduce error = (Predicted label – Actual label)
- Reinforcement learning
  - Training information: reward functions
  - Objective: learn policy
  - Goodness criteria: maximal reward
- 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





Goal- and Utility-based Intelligent Agent

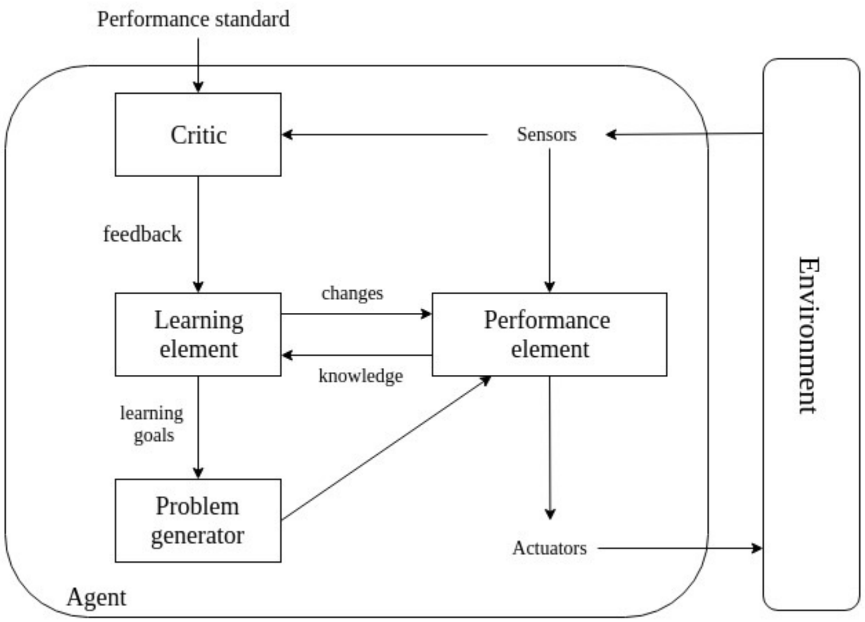
Model → Planning → Policy

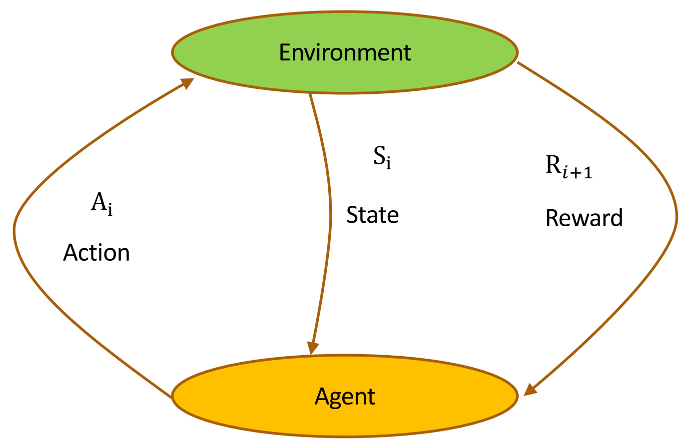
(Generalization of plan)

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





Execution trace / Experience → RL → Policy

Model → Planning → Policy
(Generalization of plan)

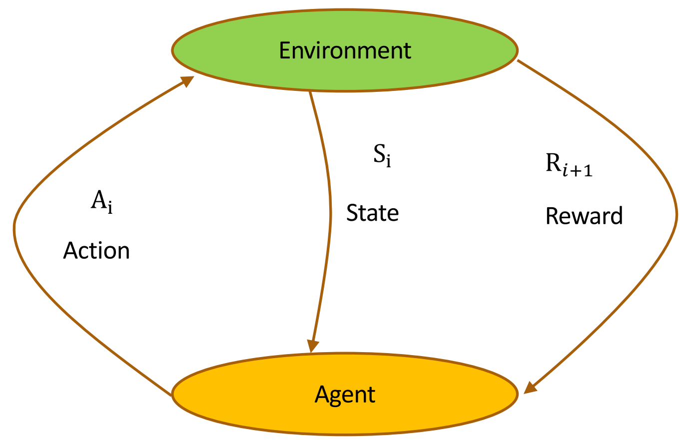
Image Source: Russell & Norvig, AI: A Modern Approach

RL as a Learning-Based Agent

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

Goal- and Utility-based Intelligent Agent

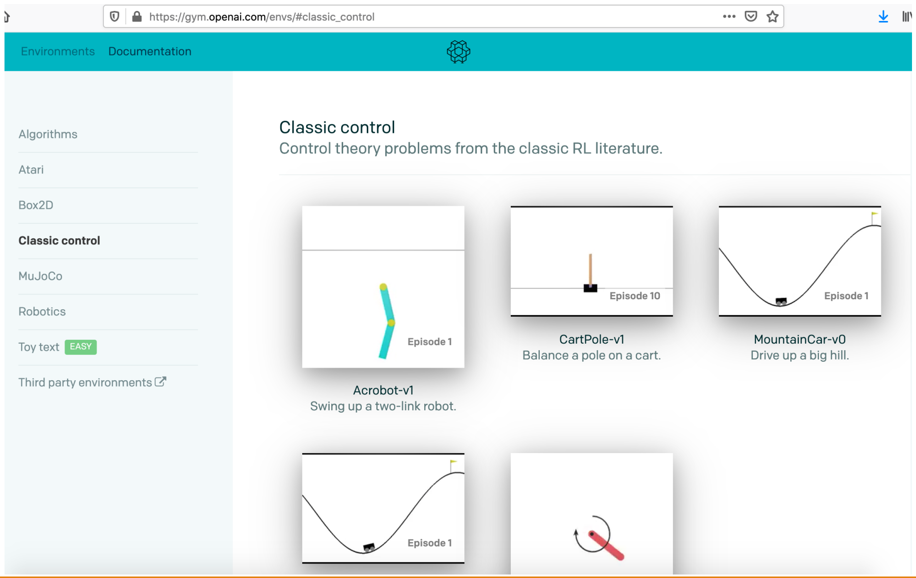
Model → Planning → Policy
(Generalization of plan)



Execution trace / Experience → RL → Policy

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



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 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**

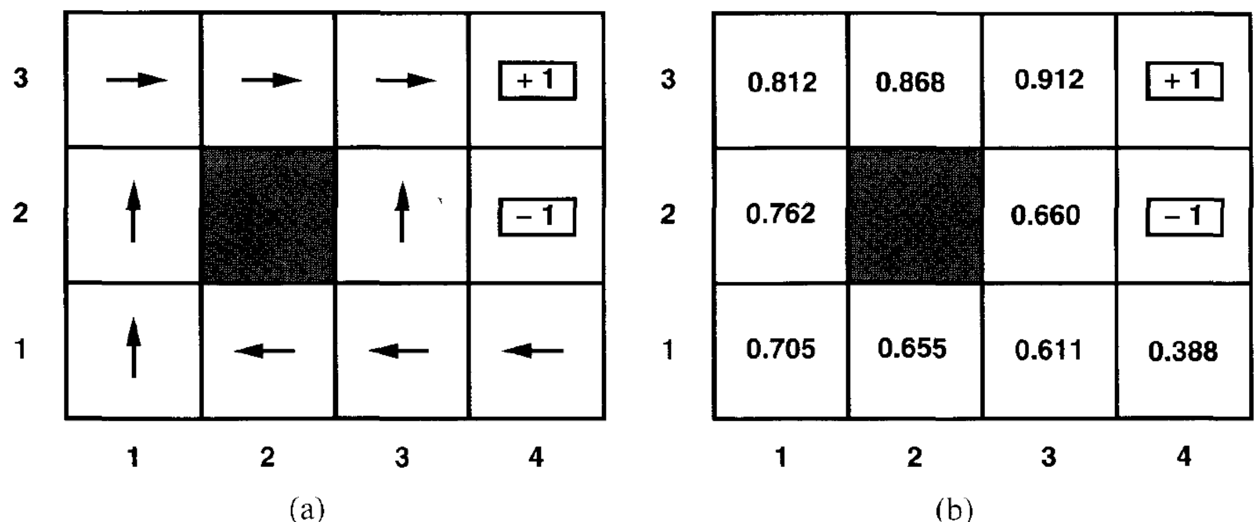
$$U^{\pi}(s) = E(\sum_{t=0}^{\inf} \gamma^t R^t(s'))$$

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
  - $\mathrm{H_t} = S_0, A_0, R_1 S_1, A_1, R_2 \ldots S_{t-1}, A_{t-1}, R_t, S_t$
- 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
  - $P(S_{t+1}|S_t) = P(S_{t+1}|S_t, S_{t-1}, \ldots S_0)$

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 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

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

Lecture 24, 25: Summary

- We talked about
  - Planning
  - Uncertainty
  - Reinforcement Learning

Concluding Section

Course Project

Discussion: Projects

- New: two projects
  - Project 1: model assignment
  - Project 2: single problem/ llm based solving / fine-tuning/ presenting result

About Next Lecture – Lecture 26

22	Nov 7 (Th)	Making Decisions - Simple
23	Nov 12 (Tu)	Making Decisions - Complex
24	Nov 14 (Th)	Sequential Decision Making: Planning, RL
25	Nov 19 (Tu)	Sequential Decision Making: Planning, RL
26	Nov 21 (Th)	Paper presentation (grad students only)
	Nov 26 (Tu)	Thanksgiving Holiday
	Nov 28 (Tu)	Thanksgiving Holiday
27	Dec 3 (Tu)	AI for the Real World – Bringing All Together; Project presentation
28	Dec 5 (Th)	Project presentation
	Dec 7 (Sa)	
29	Dec 10 (Tu)	4pm – Examination

Lecture 26: Graduate Student Presentations

- 5 presentations
  - Sample template will be given in drive (folder shared via blackboard); make a copy and edit
- Evaluation
  - By undergrads as well as instructor and TA
  - All undergraduates to attend and give response to a survey; link to be shared
  - Those undergrads not giving inputs will be given negative marks as part of the final score [-10 point per presentation]
- What to have in the report – minimum 1 page per paper (<500 words).
  - Paper summary
  - Key contributions
  - Your critique about the paper.

22	Nov 7 (Th)	Making Decisions - Simple
23	Nov 12 (Tu)	Making Decisions - Complex
24	Nov 14 (Th)	Sequential Decision Making: Planning, RL
25	Nov 19 (Tu)	Sequential Decision Making: Planning, RL
26	Nov 21 (Th)	Paper presentation (grad students only)
	Nov 26 (Tu)	Thanksgiving Holiday
	Nov 28 (Tu)	Thanksgiving Holiday
27	Dec 3 (Tu)	AI for the Real World – Bringing All Together; Project presentation
28	Dec 5 (Th)	Project presentation
	Dec 7 (Sa)	
29	Dec 10 (Tu)	4pm – Examination