In-Depth Summer of Science: Plan of Action

(Revised with Mentor Feedback & Guidelines)

Aadeshveer Singh
24B0926
24b0926@iitb.ac.in

May 15, 2025

Week 1: Logic, Automata, and Foundations (Approx. May 20th - May 26th)

Topics:

- **Propositional Logic:**
  - Syntax, semantics, truth assignments.
  - Logical connectives, truth tables, normal forms (CNF, DNF).
  - Tautologies, contradictions, satisfiability (SAT problem intro).
  - Logical entailment. *(Optional: Formal proof systems - overview)*
- *(Optional: Predicate Logic (First-Order Logic - Introduction))*
  - *(Optional: Quantifiers, variables, predicates, functions.)*
  - *(Optional: Syntax and semantics (briefly, to appreciate its expressive power).)*
- **(Core Focus: Finite Automata)**
  - Deterministic and Nondeterministic Finite Automata (DFA/NFA): formal definitions, transition functions, language acceptance.
  - Equivalence of DFA and NFA (constructive proof).
  - Regular expressions: syntax, semantics, Kleene's Theorem (equivalence with FA - understand the proof sketch).
  - *(Optional: Non-regular languages: Pumping Lemma and its applications)*
  - *(Optional: Myhill-Nerode Theorem (conceptual understanding))*
- **Connections:** Discuss how these formalisms (especially FA) model aspects of computation, state, and transitions, laying groundwork for sequential decision-making.

Week 2: Stochastic Processes - Markov Chains In-Depth (Approx. May 27th - June 2nd)

Topics:

- **Markov Chains (MCs):**
  - Formal definition, state space (discrete and continuous time - focus on discrete), transition matrix/kernel.
  - Chapman-Kolmogorov equations.
  - N-step transition probabilities.
- **Classification of States:**
  - Accessibility, communicating classes.
  - Recurrence (positive/null), transience, periodicity.
  - Irreducible MCs, aperiodic MCs, ergodic MCs.
- **Long-Term Behavior:**
  - Limiting distributions, stationary distributions: existence and uniqueness (conditions like ergodicity).

  - Convergence to stationary distribution. *(Optional: Rate of convergence - briefly, e.g., spectral gap)*
  - First passage times, mean recurrence times.

- **Absorbing Markov Chains:**
  - Canonical form, fundamental matrix, absorption probabilities, expected time to absorption.

- **Implementation:** Simulate a few MCs, compute stationary distributions, analyze absorbing chains for small examples.

Week 3: Markov Decision Processes - Foundations and Exact Solutions (Approx. June 3rd - June 9th)

Topics:

- **MDP Components:** Thorough understanding of states, actions, transition probabilities (model dynamics $P(s'|s,a)$), rewards $R(s,a,s')$.

- **Policies and Value Functions:**
  - Deterministic and stochastic policies ($\pi(a|s)$).
  - State-value function $V^\pi(s)$, Action-value function $Q^\pi(s,a)$.

- **Bellman Equations:**
  - Bellman expectation equation for V^π and Q^π (derive them).
  - Bellman optimality equation for V^* and Q^* (derive them).
  - Bellman operators (T^π, T^*) and their properties (e.g., contraction mapping, monotonicity).

- **Exact Solution Algorithms:**
  - **Value Iteration (VI):** Algorithm, proof of convergence (using contraction mapping property).
  - **Policy Iteration (PI):** Algorithm (policy evaluation, policy improvement), proof of convergence, relationship to VI.
  - Generalized Policy Iteration (GPI).

- **Computational Complexity:** Analyze the complexity of VI and PI.

- *(Optional: Linear Programming Formulation for MDPs: Understand how MDPs can be solved using LP.)*

- **Implementation:** Implement VI and PI for grid-world environments. Analyze their convergence.

Week 4: Buffer Week & Midterm Report Preparation (Approx. June 10th - June 16th)

Activities:

- Consolidate understanding of Weeks 1-3.
- Work on implementations and debug.
- **Midterm Report Submission (Target: Mid-June):** Report should include theoretical summaries (focusing on covered topics), derivations (e.g., Bellman equations), and small implementation results (e.g., MC simulation, VI/PI on a grid world). Discuss challenges faced and learnings.

Week 5: Formulating RL Problems & Advanced MDP Concepts (Approx. June 17th - June 23rd)

Topics:

- **Introduction to Hidden Markov Models (HMMs):** *(Moved here; Optional or overview per Mentor suggestion)*
  - Definition, key problems (filtering, smoothing, decoding).
  - Contrast with observable MCs and fully observable MDPs.
- **Reward Engineering:**
  - Principles of good reward design.
  - Sparse vs. dense rewards.
  - Potential-based reward shaping (Ng, Harada, Russell, 1999) - theory and benefits (policy invariance).
  - Common pitfalls and unintended consequences.
- **Problem Formulation:**
  - Episodic vs. Continuing tasks.
  - Horizon: Finite, infinite, first-exit.
  - Discounting factor (γ): role, interpretation, impact on optimality.
- **Case Studies & Gym MDPs:**
  - Analyze structure of various OpenAI Gym (or Gymnasium) environments (e.g., Cart-Pole, MountainCar, Acrobot). Understand their state/action spaces and reward functions.
  - Discuss how to model real-world problems as MDPs.
- **Handling Large State Spaces (Motivation for Approximation):**
  - The curse of dimensionality.
  - Need for function approximation.
- *(Optional: Partially Observable MDPs (POMDPs) - Deeper Dive (Beyond HMM intro))*
  - *(Optional: Formal definition, belief states ($b(s)$).)*

- *(Optional: Value functions over belief states.)*
- *(Optional: Challenges: Intractability of exact solutions. Overview of common approaches (e.g., point-based VI, policy gradient for POMDPs).)*

- *(Optional: Inverse Reinforcement Learning (IRL) - Deeper Dive)*
  - *(Optional: Concept: Learning rewards from expert demonstrations.)*
  - *(Optional: Key algorithms: MaxEnt IRL, Bayesian IRL (overview of principles, assumptions, and challenges).)*

Week 6: Model-Free Reinforcement Learning - Core Algorithms & Nuances (Approx. June 24th - June 30th)

Topics:

- **Monte Carlo (MC) Methods:**
  - First-visit vs. Every-visit MC prediction.
  - MC control (exploring starts, on-policy, off-policy).
  - *(Optional: Off-policy MC control using importance sampling (ordinary and weighted).)*
- **Temporal Difference (TD) Learning:**
  - TD(0) prediction.
  - Advantages of TD over MC.
  - SARSA (On-policy TD control): Algorithm, convergence properties.
  - Q-Learning (Off-policy TD control): Algorithm, convergence proof sketch. Difference from SARSA.
  - *(Optional: Expected SARSA.)*
- *(Optional: N-step TD Learning:)*
  - *(Optional: N-step TD prediction.)*
  - *(Optional: N-step SARSA.)*
- *(Optional: Eligibility Traces:)*
  - *(Optional: TD(λ): forward view and backward view (accumulating and replacing traces).)*
  - *(Optional: Watkins's Q(λ), Peng's Q(λ), SARSA(λ).)*
- **Exploration vs. Exploitation:**
  - ϵ-greedy, ϵ-decreasing strategies.
  - Optimistic initialization.
  - *(Optional: Upper Confidence Bound (UCB) action selection.)*
  - *(Optional: Softmax (Boltzmann) exploration.)*
- **Implementation:** Implement Q-learning, SARSA, and potentially MC control for simple Gym environments. Experiment with different exploration strategies.

Week 7: Advanced RL - Function Approximation, Policy Gradients (Approx. July 1st - July 7th)

Topics:

- **Function Approximation in RL:**
  - Value function approximation: $\hat{V}(s, \mathbf{w}) \approx V^\pi(s)$, $\hat{Q}(s, a, \mathbf{w}) \approx Q^\pi(s, a)$.
  - Linear function approximation: features, gradient descent methods (SGD).
  - Gradient MC, Semi-gradient TD(0), Semi-gradient SARSA. *(Optional: The deadly triad.)*
  - **Deep Q-Networks (DQN):**
    * Architecture using Neural Networks.
    * Experience Replay.
    * Target Networks.
    * *(Optional: Variations: Double DQN, Dueling DQN (overview).)*
- **Policy Gradient Methods:**
  - Policy approximation $\pi(a|s, \theta)$.
  - Policy Gradient Theorem (derive or understand derivation).
  - REINFORCE algorithm (Monte Carlo Policy Gradient).
  - *(Optional: REINFORCE with Baseline.)*
- *(Optional: Actor-Critic Methods:)*
  - *(Optional: Concept: Separate actor (policy) and critic (value function).)*
  - *(Optional: Advantage Actor-Critic (A2C).)*
  - *(Optional: Asynchronous Advantage Actor-Critic (A3C) - conceptual overview.)*
- *(Optional: Model-Based RL (Overview):)*
  - *(Optional: Learning a model of the environment (P, R).)*
  - *(Optional: Dyna-Q: Integrating planning, acting, and learning.)*
  - *(Optional: Comparison: Model-based vs. Model-free RL.)*
- **Applications (Conceptual Overview):**
  - AlphaGo/AlphaZero: MCTS, neural network architecture, self-play.
  - Robotics: Challenges in continuous state/action spaces, sim-to-real transfer.

Week 8: Buffer Week, Project Completion & Endterm Report (Approx. July 8th - July 14th)

Activities:

- Intensive work on the coding project (Flappy Bird with DQN or advanced Q-learning).
- Debugging, experimentation, hyperparameter tuning.
- **Endterm Report Submission (Target: Mid-July):** Comprehensive document detailing theoretical understanding (based on covered topics), project design, implementation details, experimental results (learning curves, performance metrics), challenges, and future work.
- Prepare a short presentation of your project if required.

Main Coding Project: Mini Game with RL Agent

Description:

- **Game:** Create a simplified version of Flappy Bird (or similar simple game).
- **Agent Baseline:** Implement with tabular Q-learning (discretized state space if needed).
- **Agent Advanced:** Implement with Deep Q-Learning (DQN) using a simple neural network (e.g., PyTorch or TensorFlow/Keras).
- **Experimentation:**
  - Compare performance, learning speed, and stability of Q-learning vs. DQN.
  - Analyze the effect of different network architectures, replay buffer size, target network update frequency for DQN.
- **Visualization:** Plot learning curves (e.g., rewards per episode). Demonstrate the agent improving over episodes.

Additional Optional Mini-Projects (to reinforce weekly concepts):

- Grid World Solver (Value Iteration& Policy Iteration) - Week 3/4
- Markov Chain Analyzer - Week 2
- Tabular Q-Learning/SARSA on Gym's FrozenLake/CliffWalking - Week 6

References and Learning Resources

This Plan of Action will primarily draw upon the following resources, supplemented by additional papers and online materials as needed.

Primary Textbooks:

1. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction (2nd ed.)*. MIT Press. (Core RL concepts from Week 4 onwards)
2. Huth, M., & Ryan, M. (2004). *Logic in Computer Science: Modelling and Reasoning about Systems (2nd ed.)*. Cambridge University Press. (Week 1: Propositional Logic, Automata)
3. Baier, C., & Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press. (Week 1: Automata; Week 2-3: Markov Chains, MDPs)

Key Online Lecture Series& Slides:

1. David Silver (DeepMind/UCL) - RL Lecture Series: https://youtube.com/playlist?list=PLqYmG7hTraZDVH599EItlEWsUOsJbAodm (RL from Week 4 onwards)
2. Balaraman Ravindran (NPTEL IIT Madras) - RL Lecture Series: https://youtube.com/playlist?list=PLwRJQ4m4UJjNymuBM9RdmB3Z9N5-0IlY0 (RL from Week 4 onwards)
3. Dave Parker (University of Birmingham) - Probabilistic Model Checking Lectures (including MDPs): https://www.prismmodelchecker.org/lectures/pmc/ (MDPs - Week 3)

Software& Libraries:

1. Python 3.x
2. Gymnasium (OpenAI Gym fork)
3. NumPy, Matplotlib
4. PyTorch or TensorFlow/Keras
5. Pygame (for custom environments/visualizations)

Additional Support:

1. Practice problem sheets provided by SoS organizers.
2. Discussions with mentor and peers.