Mathematical Modeling and Markov Decision Process (MDP) Formulation

To computationally derive an optimal strategy for the game of Chomp, the physical mechanics of the perfect-information, deterministic game must be rigorously formalized into a Markov Decision Process (MDP). The MDP establishes the mathematical environment in which the Reinforcement Learning (RL) agents operate, defined by a tuple of States, Actions, Transitions, and Rewards.

1. State Space Representation and Dimensionality Reduction

The formulation of the state space represents the critical mathematical optimization within this model, designed to circumvent the curse of dimensionality.

- **The Computational Bottleneck:** If the 4x5 chocolate bar were modeled as a standard binary grid (representing each of the 20 squares as a 1 or 0), the theoretical state space would encompass 2^20 (over 1,000,000) possible combinations. Navigating a state space of this magnitude would be computationally intractable for standard tabular RL methods.
- **Tuple Representation:** The mechanics of Chomp dictate that taking a bite at coordinate (r, c) removes all squares below and to the right. This "gravity" constraint ensures that the length of any given row can never exceed the length of the row immediately above it. Consequently, the model abandons the binary grid in favor of a "Tuple of Row Lengths" to represent the board's state. For instance, the initial full board is strictly defined as the tuple (5, 5, 5, 5).
- **State Space Quantification:** By applying combinatorial mathematics to calculate the possible paths of the boundary line separating the eaten and uneaten sections of the grid, the exact number of reachable states can be deduced. For a board with R rows and C columns, the formula is:

$$\text{Total States} = (R + C \; choose \; R) = (4 + 5 \; choose \; 4) = \frac{9!}{4!5!} = 126$$

Optimization Conclusion: This mathematical mapping successfully collapses the initial 1,000,000+ theoretical states into an optimized state space of exactly 126 valid tuple-states. This profound dimensionality reduction allows the Q-table to remain small, enabling the agent to converge on an optimal policy in seconds rather than days.

2. Action Space Definition

In any given state, an action is defined as the selection of a valid coordinate (r, c) representing the top-left corner of the intended bite. The action space is dynamically masked; the agent is mathematically constrained to only select from squares that currently exist within the specific tuple-state, effectively eliminating invalid explorations.

3. Transition Dynamics and Induced Stochasticity

A defining characteristic of this specific MDP formulation is the treatment of the Transition Function. While Chomp itself is a deterministic game, the transition probability within the MDP is modeled as stochastic.

- **Opponent Encapsulation:** In standard single-agent MDPs, an action leads directly to the next state. However, as Chomp is a two-player game, this model encapsulates the opponent's behavior directly into the environment.
- **The Transition Timeline:** When the RL agent executes an action, the board transitions into an intermediate state. The environment (acting as the opponent) then processes this intermediate state and executes a counter-move. Only after the opponent's reaction does the board enter the final state presented to the agent for its next turn.
- **Justification for Stochasticity:** Because the opponent's policy is particularly during initial training phases against a Random Bot which acts as an unpredictable variable, the exact resulting state following the agent's action cannot be guaranteed. Therefore, from the perspective of the single learning agent, the transition dynamics P (s' | s, a) represent a probability distribution rather than a deterministic mapping.

4. Reward Formulation and the Sparse Reward Problem

The agent's objective is formalized through a sparse, delayed reward signal, which aligns with the Misère play conditions of the game.

- **Reward Design:**
    - +1 if the opponent is forced to eat the poison.
    - -1 if the agent eats the poison.
    - 0 for every other move.

  Because the reward is 0 until the very end of the game, this is called a sparse, delayed reward problem.

- **Discount Factor:** Because Chomp is a strictly finite game guaranteed to reach a terminal state, the discount factor is maintained at 1.0. This ensures that the

terminal reward signal propagates backward through the state-action pairs without decay, a factor that becomes vital during the implementation of the SARSA algorithm's backward pass.