Chomp Solution

Aditya Neeraje

2021-06-01

1 The game

Chomp involves players alternately removing ("chomping") blocks from a board, with one corner square being poisoned. It turns out Chomp is always a $\mathscr{N}$ win, where $\mathscr{N}$ is the Next player to play. This is because $\mathscr{N}$ can always steal the winning strategy of $\mathscr{P}$ if $\mathscr{P}$ had one. If $\mathscr{N}$'s first move was to remove the corner square alone, then, if $\mathscr{P}$ had a move to a position which was a win for the player to move to that position, $\mathscr{N}$ could have moved to that position on his first move and thus attained a winning position. Since there is no possibility of a draw, every position is winning for either $\mathscr{N}$ or for $\mathscr{P}$.

2 The algorithm

The .json file contains a dictionary with the winning move for certain positions of a Chomp board. These positions cover all possible positions that are attainable if white plays according to the strategy laid out in the 4x7_winning_moves.json file.

Proof: The dictionary of winning positions has been generated (see code in chomp_solution.ipynb) by taking a $n \times m$ starting board. The basecase is when the entire board (including the poisoned block) has been eaten up, which is always a win for $\mathscr{N}$, with $\mathscr{N}$ representing the Next player to play. I have assigned a value of 1 to all such positions. Note that in Chomp, either a position is a win under optimal play for $\mathscr{N}$ or for $\mathscr{P}$, since draws are not possible. So, for a given position, if all its subpositions (found by the function generate_states()) are $\mathscr{N}$ wins, that state is a $\mathscr{P}$ win. If even one subposition is a $\mathscr{P}$ win, the Player currently playing would do well to move to that state, thus attaining a winning position for himself. In the case of multiple winning moves from one state, only the first to be found is recorded. However, no position which is a winning position could have missed being recorded under the algorithm, since each position attainable in the game has its subpositions analyzed by the algorithm.

3 How to Interpret the .json file

Each string of length 28 represents a position of the 4×7 Chomp board (4 horizontal rows, 7 vertical columns). The first 7 characters represent the bottom-most row from left to right, and so on. A full stop (.) represents a block that has not been eaten while an X represents a block that has been eaten. All keys of the dictionary are winning positions for white and the values represent the states that white should move to from these winning positions (there may be other winning moves from a particular state, this algorithm only finds one winning move from every winning state and then terminates).

4 chomp_player.py

chomp_player.py has a simple Selenium script that plays Chomp on the website https://www.mathsisfun.com/games/chomp/ with the winning moves from the .json file. I have tried to handle exceptions so that the script waits for over a minute for loading different UI elements, but it is still recommended to play with a stable internet connection. If the player wins, it terminates with the message "Hey, we won". Feel free to uncomment line 10 of the code if you just want the result printed (in approx. 20 seconds) instead of watching the entire game.