Assignment II - Model Checking

PRISM property and the result

the property is

```
Pmax = ?[F win = 1]
```

the result takes 3021.70 seconds to get the value in the initial state: 0.9989457292305014.

```
Type:        MDP
States:      11775953 (1 initial)
Transitions: 64252110
Choices:     36979950

Transition matrix: 675444 nodes (10 terminal), 64252110 minterms, vars:
39r/39c/5nd

Prob0A: 10 iterations in 4.83 seconds (average 0.482700, setup 0.00)

Prob1E: 59 iterations in 65.81 seconds (average 1.115339, setup 0.00)

yes = 4547024, no = 1807331, maybe = 5421598

Computing remaining probabilities...
Engine: Hybrid

Building hybrid MTBDD matrices... [nm=16, levels=39, nodes=2426980] [111.1 MB]
Adding sparse bits... [levels=10-37, num=66240, compact=16/16] [11.4 MB]
Creating vector for yes... [dist=2, compact] [22.5 MB]
Allocating iteration vectors... [3 x 89.8 MB]
TOTAL: [414.5 MB]

Starting iterations...
Iteration 6: max relative diff=1.#INF00, 5.82 sec so far
Iteration 12: max relative diff=0.324286, 11.44 sec so far
Iteration 17: max relative diff=0.087683, 16.68 sec so far
Iteration 23: max relative diff=0.023668, 22.28 sec so far
Iteration 29: max relative diff=0.006777, 27.76 sec so far
Iteration 35: max relative diff=0.001958, 33.18 sec so far
Iteration 41: max relative diff=0.000565, 38.63 sec so far
Iteration 47: max relative diff=0.000163, 44.01 sec so far
Iteration 53: max relative diff=0.000047, 49.55 sec so far
Iteration 59: max relative diff=0.000013, 55.09 sec so far
Iteration 65: max relative diff=0.000004, 60.49 sec so far
Iteration 71: max relative diff=0.000001, 65.80 sec so far

Iterative method: 72 iterations in 3021.70 seconds (average 0.927611, setup
2954.91)

Value in the initial state: 0.9989457292305014

Time for model checking: 3093.298 seconds.
```

Strategy that is optimal for both goals

I believe that there is a strategy that is optimal for both players' objectives, the idea being that neither player desires to win, by changing the opponent's strategy from completely random to the same as the player's, easily by the property Pmax = ? [F ! win = 1 & ! win = 2] to get a 100% chance of a draw, which ensures that neither player nor opponent loses, but if the aim is to ensure that both players have the best chance of winning rather than a draw, there is no strategy, because if both players always do the best move, they are essentially preventing their opponent from winning all the time.

Research with Markov chains

The first research I choose is using Markov Decision Process based strategies to develop convergent solutions and evaluate the performance of these solutions in games like Tic-tac-toe(Albritten, L. A. (2022)). These experiments use similar to my mdp process in question 1, only one player used the MDP strategy and played 1M games, the result shows that the MDP strategy is capable of performing well, getting a winning rate of 85.5% for the whole 1.1M games.
The second research is using Markov chains for "Count Your Chickens"(David & Lori (2019)), it uses Markov chains to analyze this cooperative game, the author mentioned they get an 81.77% chance to win by building transition matrix P for a Markov chain, this module allows players to know the win probability after each move, and when player maximises chances of winning by painting the square with the highest probability of landing during the game blue.

Reference

Albritten, L. A. (2022). Game theoretical solutions in blackjack and chess via a markov decision process algorithmic analysis (Order No. 29253418). Available from ProQuest Dissertations & Theses Global. (2681835371). Retrieved from https://liverpool.idm.oclc.org/login?url?url=https://www.proquest.com/dissertations-theses/game-theoretical-solutions-blackjack-chess-via/docview/2681835371/se-2

David McCune & Lori McCune (2019) Counting Your Chickens With Markov Chains, Mathematics Magazine, 92:3, 163-172, DOI: 10.1080/0025570X.2019.1561097