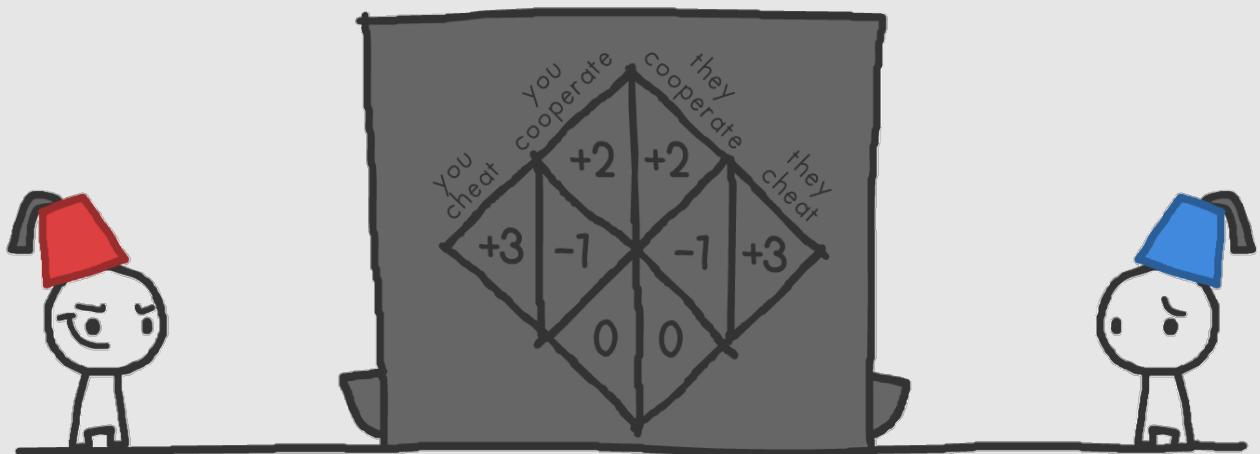


OFFICIAL GUIDEBOOK

THE PRISONER'S DILEMMA GAME



CHEAT

COOPERATE



Contents

1	Introduction to Prisoner's Dilemma	2
2	Strategies	3
3	Devising Your Strategy	3
3.1	What's an FSA?	3
3.2	Typing up your FSA	5
4	How Does the Game Run?	6
5	An Example of a Strategy	7



The Prisoner's Dilemma Game

The Official Guide to Strategy-Making

Kartikey Awasthi

1 Introduction to Prisoner's Dilemma

Prisoner's dilemma (here on abbreviated as P.D) is an extensively studied game with extremely simple rules of play- Two players have an independent choice of making a move from the given option of two- Cooperate (abbreviated 'C') and Defect (abbreviated 'D'). From these choices, come the results of the game:-

1. If both play C, both earn 3 points.
2. If both play D, both earn 1 point.
3. If one plays D and other C, the former earns 5 points and latter earns 0.

Given below is the payoff matrix of PD graphically –

		player	
		COOPERATE	DEFECT
player	COOPERATE	3 3	5 0
	DEFECT	5 0	1

Figure 1: Payoff matrix of Prisoner's Dilemma Game

The above given point distribution will be used for our game as well.

2 Strategies

A ‘move’ refers to the choice made in a round of the PD game. Ours is an Iterated PD game wherein we have multiple rounds of the game-meaning the players get to move more than once. The points per round are summed for the final result.

A ‘strategy’ is what will really play the game and what the participants need to devise. It is a set of rules that will define how to choose the moves for every round of a match. So, instead of two people sitting on the computers thinking of what move to select for multiple rounds, they simply send in pre-formed strategies that substitute in their place in the game.



The next section explains how to make strategies for this game. Section 4 would go on and explain the rules for participation and sending in your strategy, while Section 5 will give example strategies for a clearer understanding.

3 Devising Your Strategy

To play this game, we will be using an open-source, pedagogical software called ‘Oyun’ (pronounced OY-oon) which was created by Charles H. Pence for teaching concepts of evolutionary game theory. Oyun literally means ‘game’ in Turkish.

Oyun allows strategy input in an FSA format written as a.txt file.



Figure 2: Logo of OYUN- A PD game simulator

3.1 What’s an FSA?

FSA stands for ‘finite state automata’, and this will be the format that will help the simulator understand our strategy. FSA has ‘states’ which, in our case, represent the moves to play. Counting of states begins from 0. The first move which the strategy plays in the game is the one in state 0. From then on, the ‘reader’ can go to another state via ‘arrows’ connecting them. In our case, the arrows will tell the opponent’s last move.

Here is an example of an FSA showing the TIT FOR TAT strategy- an in-built strategy that has rules like this- ‘Cooperate the first turn and then on mimic the opponent’s last move’.

Keep these rules in mind while designing the automaton for your strategy-

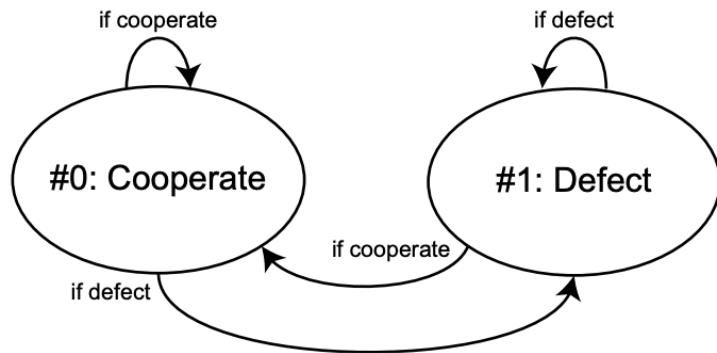


Figure 3: An FSA diagram depicting the TIT FOR TAT strategy

1. Keep numbering your states from 0.
2. Make sure the ‘reader’ always has a state to go to, i.e. each state must have one ‘if the opponent played C then’ and one ‘if the opponent played D then’ arrows.
3. We don’t need a ‘final’ state. Just make sure the above rule is completed.
4. Like in the example, you don’t need two separate states for every round. You can always circle back to the same state or go back to the state you’ve been to.
5. The FSA here is a ‘deterministic’ model which means you need to make sure there aren’t two (or more) arrows of the same kind emanating from one state (say State 0 is connected to both State 1 and State 5 with an ‘if C’ arrow).

Remember, a strategy must have a rule since the end of the game is probabilistic. That’s right, with each round, there is a 0.00346 chance of the game to end (Axelrod’s game end probability). With no sure end, making a strategy where every round is accounted for separately is a moot endeavour.

Put on your thinking caps and start cracking!

3.2 Typing up your FSA

Coding out your FSA is a piece of cake for Oyun once you've chalked out its design. Follow these steps and make your strategy complete for submission into the tournament! Read up Section 5 to see an example of a strategy.

1. The file to send is a .txt file. You can make one on 'Notepad' if using Windows; TextEditor if using macOS and on the Terminal itself if using Linux.
2. In your .txt file, type out the following-

```
Strategy Author
Strategy Name
Number of states (N)
State 0
State 1
...
State N-1
```

Here, each state information should be given as below-

<move>, <next state if opponent moves C>, <next state if opponent moves D>

As an example-

```
IISER Mohali
TIT FOR TAT
2
C, 0, 1
D, 0, 1
```

3. Check the above example with the FSA diagram given above to get a feel. Your file must have only the aforementioned text and nothing else. Remember, each state MUST be associated with a number and a move to play, i.e. either C or D.
4. And just like this, you have yourself a strategy complete and ready to take part in the tournament!

4 How Does the Game Run?

Once all the strategies are compiled, a round-robin tournament begins on Oyun (including a built-in RANDOM strategy of the simulator). The final scores are tallied as the arithmetic mean of points the strategy received from all its matches. The winning strategy would be one with the highest average. The winning strategy would be one with the highest average.

Another important thing is the ‘survival scaling’ that would occur for all strategies. Oyun also plugs in all the strategies in a simulated evolutionary environment. Successful strategies (the ones gaining high average points) translate to high fitness traits and proportionately increase in number in the virtual population of the next generation.



With this, every strategy’s growth is monitored on a graph. Every strategy starts with the same fraction. Natural selection through competition drives the change in the distribution by proportion of all strategies. It’s run for 200 generations, and the final fraction of the strategy in the population is multiplied to 1000 and is added to the mean score from the original game.

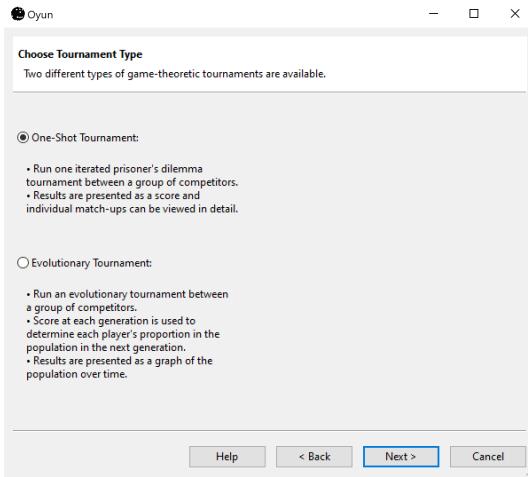


Figure 4: We will use both options of Oyun for the tournament

5 An Example of a Strategy

Given below is the PARASITE strategy

Charles Pence

Parasite

12

C, 1, 11
C, 2, 11
C, 3, 11
D, 7, 4
D, 5, 6
D, 5, 5
C, 6, 6
C, 8, 8
C, 9, 9
C, 10, 10
C, 10, 11
D, 10, 11

This code corresponds to the PARASITE strategy whose FSA diagram is given below-

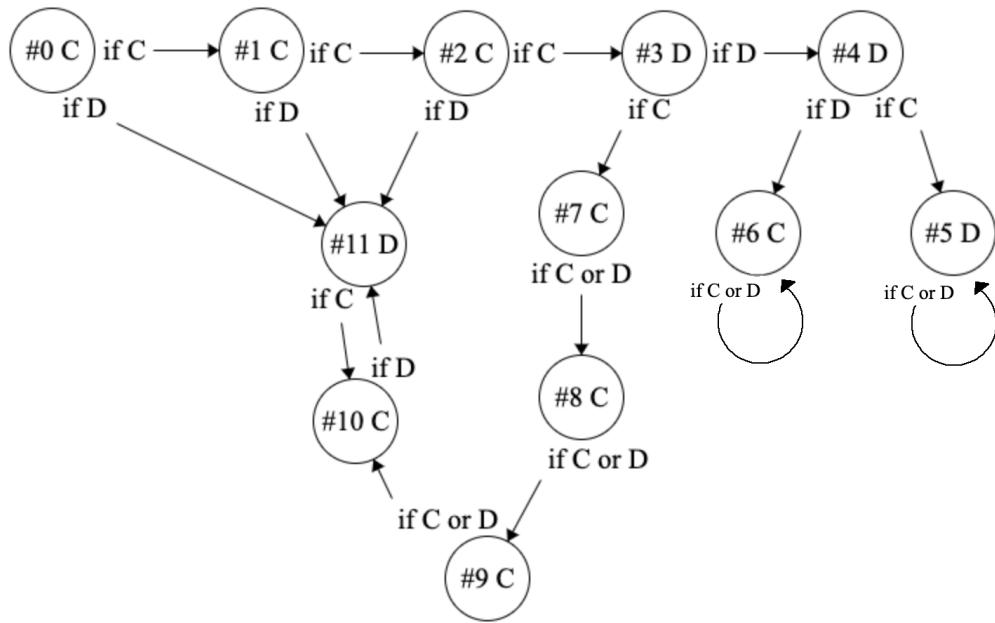


Figure 5: FSA Diagram for the PARASITE strategy

Here is the same PARASITE code with comments to understand the steps-

```

Charles Pence
Parasite
# Lead in to handshake -- if this fails, fall back to TfT
#0 C, 1, 11
#1 C, 2, 11
#2 C, 3, 11
# The handshake itself, if it fails, make up and then play
TfT
#3 D, 7, 4
# Handshake was successful, this is the identification turn:
D means we're a parasite, C means we're a host
#4 D, 5, 6
# I'm a parasite and he's a host--take advantage loop
#5 D, 5, 5
# Two parasites--cooperate loop
#6 C, 6, 6
# Three cooperates -- the apology
#7 C, 8, 8
#8 C, 9, 9
#9 C, 10, 10
# Standard TfT
#10 C, 10, 11
#11 D, 10, 11

```

References

- [1] Oyun User Manual at <https://charespence.net/oyun/>
- [2] Pence, C.H., Buchak, L. Oyun: A New, Free Program for Iterated Prisoner's Dilemma Tournaments in the Classroom. *Evo Edu Outreach* 5, 467–476 (2012). <https://doi.org/10.1007/s12052-012-0434-x>
- [3] FSA diagram drawn using Finite State Machine Designer by Evan Wallace (2010) <http://madebyevan.com/fsm/>
- [4] Axelrod, Robert (1984), *The Evolution of Cooperation*, Basic Books, ISBN 0-465-02122-0