Introduction to Game Theory

B. Nebel, R. Mattmüller, S. Wölfl
T. Schulte, D. Speck
Summer semester 2016

University of Freiburg
Department of Computer Science

Exercise Sheet P2

Due: Tuesday, July 19, 2016

Exercise P2.1 (Repeated Prisoner's Dilemma, 1 + 3 points)

Your task is to implement a strategy for the indefinitely repeated prisoner's dilemma. The Python 3 framework you are supposed to extend can be found on the course website[1]. We will run a tournament where each player competes against each of the other players. The authors of the player with the overall maximum score earn 2 additional points for this exercise. Submit your solution via email to `schultet@informatik.uni-freiburg.de`.

(a) Create a text file `description.txt` containing a concise description of your strategy.

(b) Read the comments and instructions provided in the python module. Then, implement your players strategy by specifying suitable `transition` and `nextmove` functions. Test your implementation against the players provided in the Python module (`defectingplayer` and `tftplayer`). Implement your strategy in the way you expect to perform best against other players from the game theory class. Each encounter consists of an indefinite number of rounds. After each round the game ends with probability $\frac{1}{100}$. We use the scores according to the following payoff matrix and no discounting:

		Player 2	
		C	D
Player 1	C	$3, 3$	$0, 4$
	D	$4, 0$	$1, 1$

Note: In the lecture, finite automata were used to represent players' strategies. However, you are not restricted to use this representation in this project. Specifically, you may make use of infinitely many states and of randomized strategies.

The exercise sheets may and should be worked on in groups of two to three students.

[1] `http://gki.informatik.uni-freiburg.de/teaching/ss16/gametheory/prisoners.py`