Tutorial 3: Dynamic Games

1. Formulating an extensive form game:

   Consider a 2 player game in which player 1 can choose A or B. The game ends if she chooses A, while it continues to player 2 if he chooses B. Player 2 can then choose C or D with the game ending if C is chosen, and continuing again to player 1 if D is chosen. Player 1 can then choose E or F, with the game ending either choice.

   (a) Model this as a game tree.

   (b) How many pure strategies does each player have?

   (c) Identity the subgames of this game.

   (d) Suppose that choice A gives utilities $(2, 0)$ (i.e., 2 to player A, 0 to pleyer E), choice C gives $(3, 1)$, choice E gives $(0, 0)$, and F gives $(1, 2)$. Then what are the pure Nash equilibria of the game? What SPNE outcome(s) does Zermelo's algorithm yield?

2. Imperfect information games:

   Consider an extensive form imperfect information game in which each player i has k information sets, that is, $|\mathcal{I}_i| = k$ for all $1 \le i \le n$.

   (a) If a player has an identical number of m possible actions in each information set, how many pure strategies does he have?

   (b) If a player has m_j actions in the j'th information set ($1 \le j \le k$) how many pure strategies does he have?

3. Imperfect information games:

   Consider the three games of imperfect information shown in Figure 1 (information sets for player I are shown via ovals rather than dotted lines).

   (a) For each of the games, what information can player I lose while playing the game?

   (b) For each game, write down the information sets and pure strategies for both players.

   (c) For game B, identify any pure strategy Nash equilibria.

4. Recall the forgetful driver example from the lecture: a 1-player "game" of imperfect recall, illustrated by the game tree in Figure 3.

   A behavioural strategy for this game can be defined as a probability p that the driver will exit (and so will drive straight with probability $1 - p$).

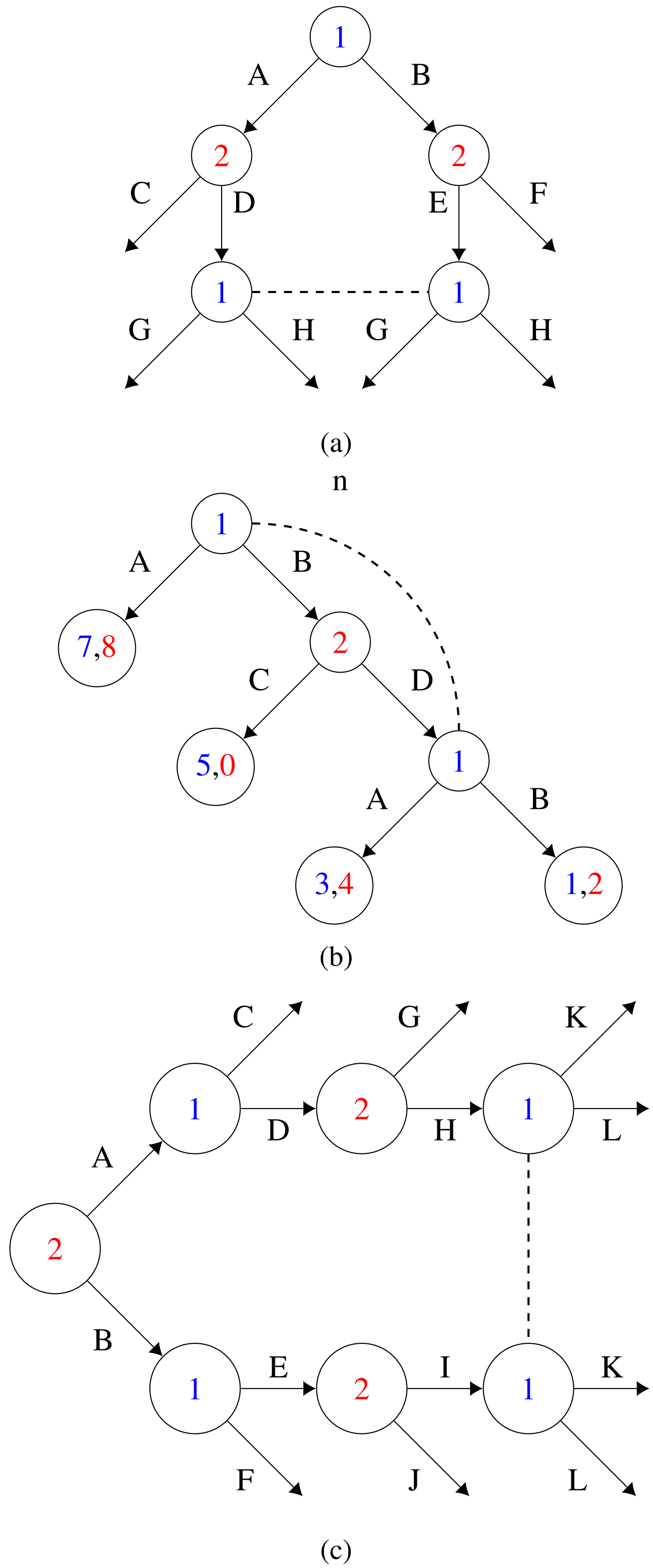

Figure 1: Three games of imperfect information.

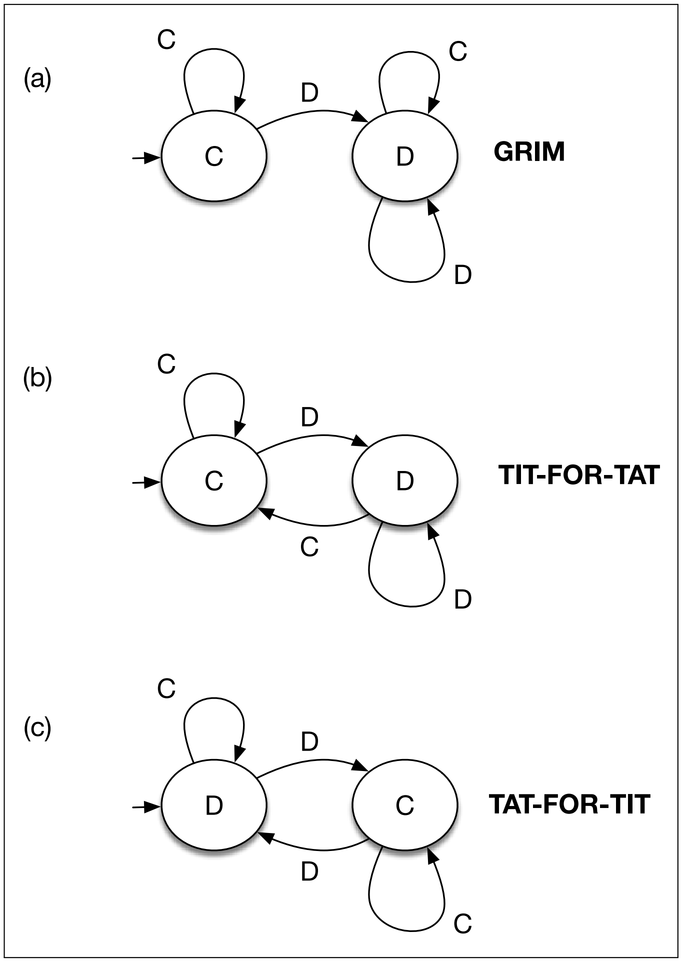

Figure 2: Three finite state machines that play the iterated prisoner's dilemma.

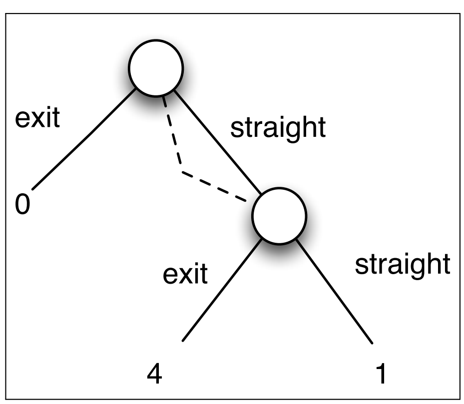

Figure 3: Game tree for the forgetful driver problem.

(a) Write down the expression for the expected utility a driver will obtain from playing a behavioural strategy p.

(b) Use the expression you derived in the first part of this question to compute a value for p that maximises expected utility for the driver (hint: use some simple calculus!)

5. Iterated games:

   Let's consider playing the infinitely repeated prisoner's dilemma, using finite state automata strategies, and measuring utility over infinite runs as the average utility obtained, as discussed in the lecture. Recall that the payoff matrix for the (one shot) prisoner's dilemma is as follows:

		2	
		defect	coop
1	defect	-2 -2	-3 0
	coop	0 -3	-1 -1

   Figure 2 shows three two-state strategies for playing the iterated prisoner's dilemma.

   (a) Informally explain what TAT-FOR-TIT does.

   (b) Consider each strategy playing against each other strategy (including itself). Compute the runs that would be generated, and identify the finite but infinitely repeating sequence of outcomes. Use this repeating sequence to compute the utility obtained by each strategy in each pairing.

   (c) Which of these pairs of strategies do you think forms a Nash equilibrium? (An informal argument will suffice.)

6. Iterated games:

   This question is for personal development only, and is not strictly speaking part of the assessment!

   Consider all possible two-state finite machines for playing the iterated prisoner's dilemma.

   (a) Suppose one of these machines has both states labelled with the same action (we have either C in both ovals, or D in both ovals). What behaviour does it generate? Can you simplify the automaton at all?

   (b) Let us say two automata strategies σ_1 and σ_2 are *distinct* if there is an automaton σ^* such that the sequence of outcomes generated by playing σ^*

against σ_1 is different to the sequence of states generated by playing σ^* against σ_2.

Claim: there are precisely 26 distinct one and two state automata. (The one-state automata are ALLD and ALLC.)

Of these 26 automata strategies, 13 will start by playing C, and the other 13 start by playing D. Draw the 13 automata that start by playing C.

Given these 13 automata, you can very easily obtain the 13 automata that start by playing D. So do it.