## 14.12 Game Theory – Midterm II
Answers

Instructions. This is an open book exam; you can use any written material. You have one hour and 20 minutes. Each question is 35 points. Good luck!

1. Two hunters go hunting, where they will play a stag-hunt game, in which each hunter simultaneously decides whether to go after a Rabbit or a Stag and the payoffs are given by

	S	R
S	6,6	0,4
R	4,0	4,4

Before the hunt, Hunter 1 can give a gift to Hunter 2 which is worth 1 utile to Hunter 2 and costs 1 utile to Hunter 1. Before they start hunting, the hunters know whether the gift is given, and all these are common knowledge. Hence, the game is as follows.
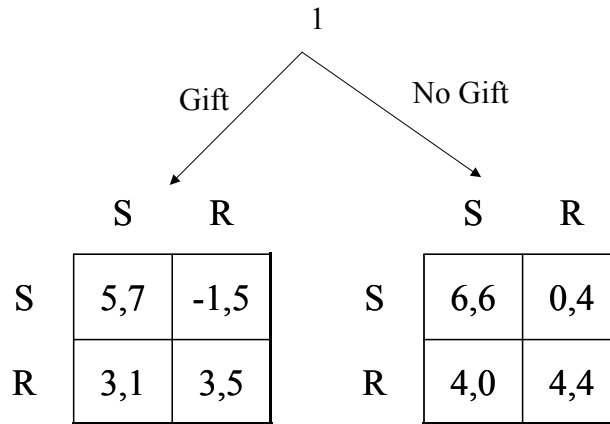
(a) Find all subgame-perfect equilibria in pure strategies.

Eachg subgame has two Nash equilibria in pure strategies: (S,S), (R,R). Therefore, there are 4 SPE in pure strategies. To describe these, let us first name the strategies for players. Player 1 has 8 strategies: GSS, GSR, GRS, GRR, NSS, NSR, NRS, and NRR, where the first letter indicates whether he gives gift (G) or not (N), and the second and third letters indicate the actions he takes in case of gift and no gift, respectively. Similarly, player 2 has four strategies: SS, SR, RS, and RR. The subgame perfect equilibria are: (NSS,SS), (NRR,RR), (NRS,RS), and (GSR,SR).

(b) Using forward induction iteratively eliminate all of these equilibria except for one.

Firstly, GRR and GRS are dominated by NRR. Moreover, GSS is not dominated. Hence, if player 1 gives a gift, then player 2 should understand that player 1 will play S (the first forward-induction argument). In that case, player 2 must play S.

1

Now, if player 1 gives a gift, then he gets 5. After these eliminations, NRR and NSR are strictly dominated by GSR. Clearly, NSS is not dominated. Therefore, if player 1 does not give a gift, then player 2 must understand that player 1 will play S (the second forward-induction argument). In that case, player 2 must also play S. Therefore, (NSS, SS) is the only subgame perfect equilibrium that remains.

2. Below, there are pairs of stage games and strategy profiles. For each pair, check whether the strategy profile is a subgame-perfect equilibrium of the game in which the stage game is repeated infinitely many times. Each agent tries to maximize the discounted sum of his expected payoffs in the stage game, and the discount rate is $\delta = 0.99$.

   (a) **Stage Game:**

1\2	S	R
S	6,6	0,4
R	4,0	4,4

   Strategy profile: Each player plays S in the first round and in the following rounds he plays what the other player played in the previous round (i.e., at each $t > 0$, he plays what the other player played at $t - 1$).

   This is a version of Tit – for –tat; it is not a subgame perfect equilibrium. Consider the case that player 1 has played S and player has 2 played R. Now, if player 1 sticks to his strategy and plays R, then the continuation play will be (RS,SR,RS,SR...), which yields $4/(1 - \delta^2)$ for player 1. If he deviates and plays S, the continuation play will be (SS, SS, SS, ...), which yields $6/(1 - \delta)$. The deviation is clearly beneficial.

   (b) **Stage Game:**

1\2	L	M	R
T	3,1	0,0	-1,2
M	0,0	0,0	0,0
B	-1,2	0,0	-1,2

   Strategy profile: Until some player deviates, player 1 plays T and player 2 plays L. If anyone deviates, then each plays M thereafter.

   This is a subgame perfect equilibrium. After the deviation, the players play a Nash equilibrium forever. Hence, we only need to check that no player has any incentive to deviate on the path of equilibrium. Player 1 has clearly no incentive to deviate. If player 2 deviates, he gets 2 in the current period and gets zero thereafter. If he sticks to his equilibrium strategy, then he gets 1 forever. The present value of this is $1/(1 - \delta) > 2$. Therefore, player 2 doesn't have any incentive to deviate, either.

3. We consider a game between two software developers, who sell operating systems (OS) for personal computers. (We also have a PC maker and the consumers, but their strategies are already fixed.) Each software developer i, simultaneously offers "bribe" b_i to the PC maker. (The bribes are in the form of contracts.) Looking at the offered

bribes b_1 and b_2, the PC maker accepts the highest bribe (and tosses a coin between them if they happen to be equal), and he rejects the other. If a firm's offer is rejected, it goes out of business, and gets 0. Let i^* denote the software developer whose bribe is accepted. Then, i^* pays the bribe b_{i^*}, and the PC maker develops its PC compatible only with the operating system of i^*. Then in the next stage, i^* becomes the monopolist in the market for operating systems. In this market the inverse demand function is given by

$$P = 1 - Q,$$

where P is the price of OS and Q is the demand for OS. The marginal cost of producing the operating system for each software developer i is c_i. The costs c_1 and c_2 are independently and identically distributed with the uniform distribution on $[0, 1]$, i.e.,

$$\Pr\left(c_i \leq c\right) = \begin{cases} 0 & \text{if } c < 0 \\ c & \text{if } c \in [0, 1] \\ 1 & \text{otherwise.} \end{cases}$$

The software developer i knows its own marginal costs, but the other firm does not know. Each firm tries to maximize its own expected profit. Everything described so far is common knowledge.

(a) What quantity a software developer i would produce if it becomes monopolist? What would be its profit?

Quantity is

$$q_i = \frac{1 - c_i}{2}$$

and the profit is

$$v_i = \left(\frac{1 - c_i}{2}\right)^2.$$

(b) Compute a symmetric Bayesian Nash equilibrium in which each firm's bribe is in the form of $b_i = \alpha + \gamma \left(1 - c_i\right)^2$.

We have a first price auction where the valuation of buyer i, who is the software developer i, is $v_i = \left(1 - c_i\right)^2 / 4$. His payoff from paying bribe b_i is

$$U_i\left(b_i; c_i\right) = \left(v_i - b_i\right) \Pr\left(b_j < b_i\right),$$

where

$$
\begin{aligned}
\Pr\left(b_j < b_i\right) &= \Pr\left(\alpha + \gamma\left(1 - c_j\right)^2 < b_i\right) = \Pr\left(\left(1 - c_j\right)^2 < \left(b_i - \alpha\right)/\gamma\right) \\
&= \Pr\left(1 - c_j < \sqrt{\left(b_i - \alpha\right)/\gamma}\right) = \Pr\left(c_j > 1 - \sqrt{\left(b_i - \alpha\right)/\gamma}\right) \\
&= 1 - \Pr\left(c_j \leq 1 - \sqrt{\left(b_i - \alpha\right)/\gamma}\right) = 1 - \left[1 - \sqrt{\left(b_i - \alpha\right)/\gamma}\right] \\
&= \sqrt{\left(b_i - \alpha\right)/\gamma}.
\end{aligned}
$$

Hence,

$$U_i\left(b_i; c_i\right) = \left(v_i - b_i\right) \sqrt{\left(b_i - \alpha\right)/\gamma}.$$

3

But maximizing $U_i(b_i; c_i)$ is the same as maximizing

$$\gamma U_i(b_i; c_i)^2 = (v_i - b_i)^2 (b_i - \alpha).$$

The first order condition yields

$$2(b_i - v_i)(b_i - \alpha) + (b_i - v_i)^2 = 0,$$

i.e.,

$$2(b_i - \alpha) + (b_i - v_i) = 0,$$

i.e.,

$$b_i = \frac{1}{3}v_i + \frac{2}{3}\alpha = \frac{1}{12}(1 - c_i)^2 + \frac{2}{3}\alpha.$$

Therefore,

$$\gamma = \frac{1}{12} \text{ and } \alpha = \frac{2}{3}\alpha \implies \alpha = 0,$$

yielding

$$b_i = \frac{1}{3}v_i = \frac{1}{12}(1 - c_i)^2.$$

(Check that the second derivative is $2(3b_i - 2v_i) = -2v_i < 0$.)

(c) Considering that the demand for PCs and the demand of OSs must be the same, should PC maker accept the highest bribe? (Assume that PC maker also tries to maximize its own profit. Explain your answer.)

A low-cost monopolist will charge a lower price, increasing the profit for the PC maker. Since low-cost software developers pay higher bribes, it is in the PC maker's interest to accept the higher bribe. In that case, he will get higher bribe now and higher profits later.