Homework #2

Question 1 (10 marks)

The purpose of this exercise is to investigate what happens to Arrow's Theorem, in its formulation for resolute social choice functions discussed in class, if we replace the Pareto Principle by the seemingly more basic surjectivity condition. Recall that we had defined surjectivity in the context of our discussion of the Muller-Satterthwaite Theorem.

(a) Show that the Pareto Principle is strictly stronger than surjectivity. That is, show that every Paretian resolute social choice function is surjective and that there exists a surjective resolute social choice function that is not Paretian.

(b) Show that Arrow's Theorem ceases to hold when we replace the Pareto Principle by surjectivity. That is, show that there exists a resolute social choice function that is surjective, independent, and nondictatorial.

Question 2 (10 marks)

Consider a scenario in which $n \geqslant 3$ voters are situated on a social network and each voter can observe at most k of the other voters. We want to run an election to choose an alternative from a set $A = \{a, b, c\}$ of three alternatives using a voting rule F. If necessary, we will break ties lexicographically, meaning that any tie involving alternative a will be broken in favour of a, and any tie involving b but not a will be broken in favour of b. We are concerned that one of the voters may have gained knowledge of the voting intentions of the other voters she can observe and is considering to manipulate the election. But we also know that she will report an untruthful preference only in case (i) she considers it *possible* that doing so will yield a better election outcome for herself than voting truthfully and (ii) she is *certain* that doing so will not yield a worse election outcome for herself than voting truthfully.

Intuitively speaking, the lower the number k, the less we have to worry about manipulation. This exercise is about trying to better understand this link between the voting rule F and the value of k (relative to n). Given n, let us call F *safe* for k in case we can be certain that no manipulation will occur when F is used for the election and no voter can observe more than k of the other $n-1$ voters. Answer the following questions:

(a) Suppose $n = 4$. Show that the Borda rule is not safe for $k = n - 2 = 2$.

(b) For any given n, characterise the range of values of k (in relation to n) for which the plurality rule is safe. An approximate characterisation (abstracting away from issues such as whether n is odd or even) would be acceptable.

(c) Intuition suggests that every voting rule should be safe for $k = 0$. Either prove that this intuition is correct or provide a counterexample.

(d) By the Gibbard-Satterthwaite Theorem, no voting rule that—when combined with our lexicographic tie-breaking rule—is both surjective and nondictatorial can possibly be safe for $k = n - 1$. This raises the question of whether we can do better for $k = n - 2$.

Either present a voting rule that (in combination with our tie-breaking rule) not only is surjective and nondictatorial but also safe for $k = n - 2$ or show that no such rule exists. (In the former case, your rule should be well-defined for any $n \geqslant 3$. In the latter case, a counterexample for one specific value of n is sufficient.)

Can you comment on the significance of your finding?

Question 3 (10 marks)

When investigating the computational complexity of problems, it is often more convenient to consider *decision problems* rather than *search problems*, even if the latter is what we are ultimately interested in. This exercise is about formulating a suitable decision problem for a specific search problem at hand. Let F be a *social welfare function* (SWF) that returns a single linear order for every given profile of linear orders, i.e., it is a function of the form $F : \mathcal{L}(X)^n \to \mathcal{L}(X)$. We are interested in the search problem of finding $F(\boldsymbol{R})$ when given a profile $\boldsymbol{R} \in \mathcal{L}(X)^n$ as input. Answer the following questions:

(a) Formulate a decision problem Q such that you can construct $F(\boldsymbol{R})$ in polynomial time by making multiple queries to an oracle for Q. Assume that every such query to the oracle requires only a single unit of time. Be very explicit about the formulation of Q: specify what the *input* is, and specify what the *question* is that is to be decided. Then explain *how* your can solve the original search problem in polynomial time if you have access to an oracle for Q (this is called a *polynomial-time Turing reduction*).

(b) Suppose Q is decidable in polynomial time. What can you conclude about the complexity of the search problem of finding $F(\boldsymbol{R})$ when given $\boldsymbol{R}$ as input? Why? Explain.

(c) Now, instead, suppose that Q is not decidable in polynomial time. Can you show that the problem of finding $F(\boldsymbol{R})$ is also not solvable in polynomial time? If so, explain how. If not, explain why not.