University of Massachusetts, Amherst, CICS
Instructor: Yair Zick
COMPSCI 590T: Algorithmic Fairness and Strategic Behavior

Assignment 6: Stable Matching

Issued: Week 8 — Due: two weeks after the issue date in Gradescope

Important Instructions:

- Please upload your solution to Gradescope by 23:59 on the due date.
- You may work with a partner on your homework; however, if you choose to do so, **you must** write down the name of your partner on the assignment.
- Provide a mathematical proof/computation steps to all questions, unless explicitly told not to. Solutions with no proof/explanations (even correct ones!) will be awarded no points.
- You may look up combinatorial inequalities and mathematical formulas online, but please do not look up the solutions. Many problems can be found in research papers/online; if you happen upon the solution before figuring it out yourself, mention this in the submission. **Unreferenced copies of online material will be considered plagiarism.**

In what follows, we have two disjoint groups: students $S = \{s_1, \dots, s_n\}$ and hospitals $H = \{h_1, \dots, h_m\}$. Each student $s \in S$ has a complete, transitive preference order over H, denoted $\succ_s$. Given two hospitals $h, h' \in H$, $h \succ_s h$ means that s prefers hospital h to hospital h'; $h \simeq_s h'$ means that s ranks both equally; $h \succeq_s h'$ means that either $h \succ_s h'$ or $h \simeq_s h'$. Similarly, we define a complete, transitive preference order for hospitals over S.

A *matching* $M : S \to H$ is a mapping from S to H such that no two students are assigned to the same hospital (in real matching scenarios a hospital with multiple identical openings can be modeled by several copies of the hospital with identical preferences, each with a single availability). Note that $M(s)$ is the hospital that a student $s \in S$ is assigned to, and that $M^{-1}(h)$ is the student that was assigned to $h \in H$. We can add several copies of dummy students and hospitals that are ranked below all true students/hospitals to simulate being assigned no student (or no hospital).

We say that a student-hospital pair $(s, h) \in S \times H$ *blocks* M, if $h \succ_s M(s)$, and $s \succ_h M^{-1}(h)$. A matching M is *stable* if it has no blocking pairs. As we have seen in class, the Gale-Shapley algorithm outputs a stable matching ofter $\mathcal{O}(|S| \times |H|)$ rounds.

1. (40 points) Given a student $s \in S$, we say that a hospital $h \in H$ is *valid* for s, if there exists some stable matching M where s is matched to h. Similarly, we say that a student s is valid for a hospital h if there exists some stable matching M where s is matched to h. Given a student s, let *best*(s) be the hospital that s ranks highest amongst all of its valid hospitals; similarly, let *worst*(s) be the *least preferred* hospital that could be assigned to s under some stable matching. We similarly define *best*(h) and *worst*(h).
   (a) (20 points) Show that when we run the Gale-Shapley algorithm with the students proposing, each student s is matched to *best*(s).
   (b) (20 points) Show that when we run the Gale-Shapley algorithm with the students proposing, each hospital h is assigned to *worst*(h).
2. (30 points) When is there a unique stable matching?
   (a) (15 points) Show that if all hospitals have the same strict preferences over students (by strict we mean that for every two students s, s', either $s \succ_h s'$ or $s' \succ_h s$), then there exists a unique stable matching.
   (b) (15 points) Show that if the student-optimal matching and the hospital-optimal matchings (derived from letting the students propose under Gale-Shapley, and letting the hospitals propose under Gale-Shapley, respectively) are the same, then there exists a unique stable matching.

3. (15 points) Give an instance of the stable matching problem where there exists more than one stable matching. Try to come up with an instance with as few hospitals and students as possible.

4. (15 points) Consider the case where we have only one group; so we are matching students to students (for example for pairing them in group projects). The setting is similar: we have a group of students S, where each $s \in S$ has a strict preference order over S. A matching partitions S into pairs, and is called stable if no pair of students prefer each other to their own match. We assume that $|S|$ is even, so that there's no unmatched student. Does a stable matching always exist in this case? Prove or provide a counterexample.

5. (Bonus points) In this question we will explore the *lattice* structure of stable matchings. Given two stable matchings M and M', let us define a function $M \vee M'$ which works as follows: $M \vee M'$ assigns each student to their more preferred hospital from M and M', i.e. if $M(s) = h_1$ and $M'(s) = h_2$, then $M \vee M'$ assigns s to the better of h_1 and h_2 in their preference order. On the other hand, $M \vee M'$ assigns each hospital h to their *less* preferred student between M and M'.

   We can define a matching $M \wedge M'$ that assigns each student their *less* preferred hospital out of M and M', and every hospital their *more* preferred student out of M and M'.

   Let us next define an order over stable matchings: we say that $M \geq M'$ if every student prefers their match under M to their match under M'.

   (a) Prove that $M \vee M'$ is a valid stable matching, i.e. that every student is assigned to exactly one hospital and every hospital is assigned to exactly one student, and that the resulting matching is indeed stable (remember that we assume that both M and M' are stable). Similarly, $M \wedge M'$ is a valid stable matching, however this case is analogous to the analysis you will perform for $M \vee M'$, so I won't ask you to repeat it.

   Hint: To show that $M \vee M'$ is a valid matching (as opposed to an arbitrary mapping of hospitals to students and vice versa), you need to show that for every student s, if $M \vee M'(s) = h$ then $M \vee M'(h) = s$.

   (b) Is the order $\geq$ we defined complete? In other words, given two matchings M and M', is it true that $M \geq M'$ or $M' \geq M$? Prove or provide a counterexample.

   (c) What are the maximal and minimal matchings under the order $\geq$ we defined?