1 Stable Marriage

Consider the set of men $M = \{1, 2, 3\}$ and the set of women $W = \{A, B, C\}$ with the following preferences.

Men	Women
1	A > B > C
2	B > A > C
3	A > B > C

Women	Men
A	2 > 1 > 3
B	1 > 2 > 3
C	1 > 2 > 3

Run the traditional propose-and-reject algorithm on this example. How many days does it take and what is the resulting pairing? (Show your work.)

Solution:

The algorithm takes 3 days to produce a matching. The resulting pairing is as follows. The circles indicate the man that a woman picked on a given day (and rejected the rest).

$$\{(A,1),(B,2),(C,3)\}.$$

Woman	Day 1	Day 2	Day 3
A	1,3	1	1
B	2	2,3	2
C			3

2 Universal Preference

Suppose that preferences in a stable marriage instance are universal: all n men share the preferences $W_1 > W_2 > \cdots > W_n$ and all women share the preferences $M_1 > M_2 > \cdots > M_n$.

(a) What pairing do we get from running the algorithm with men proposing? Can you prove this happens for all n?

(b) What pairing do we get from running the algorithm with women proposing?

(c) What does this tell us about the number of stable pairings?

Solution:

(a) The pairing results in (W_i, M_i) for each $i \in \{1, 2, ..., n\}$.

This result can be proved by induction:

Our base case is when $n = 1$, so the only pairing is (W_1, M_1), and thus the base case is trivially true.

Now assume this is true for some $n \in \mathbb{N}$.

On the first day with $n+1$ men and $n+1$ women, all $n+1$ men will propose to W_1. W_1 prefers M_1 the most, and the rest of the men will be rejected. This leaves a set of n unpaired men and n unpaired women who all have the same preferences (after the pairing of (W_1, M_1)). By the process of induction, this means that every i^{th} preferred woman will be paired with the i^{th} preferred man.

(b) The pairings will again result in (M_i, W_i) for each $i \in \{1, 2, ..., n\}$. This can be proved by induction in the same as above, but replacing "man" with "woman" and vice-versa.

(c) We know that male-proposing produces a female-pessimal stable pairing. We also know that female-proposing produces a female-optimal stable pairing. We found that female-optimal and female-pessimal pairings are the same. This means that there is only one stable pairing, since both the best and worst pairings (for females) are the same pairings.

3 Propose-and-Reject Proofs

Prove the following statements about the traditional propose-and-reject algorithm.

(a) In any execution of the algorithm, if a woman receives a proposal on day i, then she receives some proposal on every day thereafter until termination.

(b) In any execution of the algorithm, if a woman receives no proposal on day i, then she receives no proposal on any previous day j, $1 \leq j < i$.

(c) In any execution of the algorithm, there is at least one woman who only receives a single proposal. (Hint: use the parts above!)

Solution:

(a) The idea is to use the Improvement Lemma. The Improvement Lemma tells us that if w gets a proposal from m on day i, on every subsequent day she ends up with someone on a string who she likes at least as much as m. In particular, this means that at the end of every subsequent day, w has someone on a string, meaning that man must have proposed to her on that day.

(b) One way is to use a proof by contradiction. Assume that a woman receives no proposal on day i but did receive a proposal on some previous day j, $1 \leq j < i$. By the previous part, since the woman received a proposal on day j, she must receive at least one proposal on every day after j. But $i > j$, so the woman must have received a proposal on day i, contradicting our original assumption that she did not.

(c) Let's say the algorithm takes k days. This means that every woman must have received a proposal on day k. However, this also means that there is at least one woman w who does not receive a proposal on day $k-1$–if this were not the case, the algorithm would have already terminated on day $k-1$. Then from part (b), since w did not receive a proposal on day $k-1$, she didn't receive a proposal on any day before k. Furthermore, we know she got exactly one proposal on day k, since the algorithm terminated on that day. Thus, we have that w receives exactly one proposal throughout the entire run of the algorithm.