GALE-SHAPLEY WRITEUP

Spring 2016, U. Chicago
Instructor: A. Drucker

This note is redundant with the treatment of Gale-Shapley in the Kleinberg-Tardos textbook. It is meant to give you an example of the type of proof writeup requested on problem sets in this class.
(We aim to be less verbose than K-T, who are trying to convey intuition to students; however, we sometimes insist on greater rigor and completeness. Still, some amount of informal language is fine as long as the essential ideas are clearly conveyed. This is partly a matter of experience.)

The scenario: There are $N$ men and $N$ women, who are to be matched off by us into $N$ monogamous, heterosexual marriages. (exciting!)

Each man has a preference order on the women, and similarly, women have preference orders on the men.

The goal is to produce a complete matching that is stable: that is, there is no pair $M,W$ of man and woman respectively who are married to other people, but each prefer each other to their assigned spouse.

We call such a pair $(M,W)$ as above an eloping pair. Thus a stable matching is one that contains no eloping pairs. Gale and Shapley proved that there always exists a stable matching, regardless of the particular preference orders of the men and women. Their algorithm can be used to find such a matching, and also furnishes a proof that this matching does exist.
(One additional note: there may be more than one stable matching in general, but not every such matching can be found by the Gale-Shapley algorithm.)

Some terminology: Before stating the algorithm, let’s define some terms. The following types of events occur in our algorithm:

- A man proposes to a woman;
- A woman accepts or rejects a proposal made to her;
- If $W$ accepts $M$’s proposal, the pair become engaged (they are fiancées);
- If $W$ accepts $M$’s proposal while previously engaged to someone else, the prior engagement is broken.
Men will only propose if they are not single, i.e., not currently engaged; so at any given time, the current set of engagements forms a partial matching between men and women. That is, each man and each woman has at most one fiancee.

We say that \( W \) snubs \( M \) if one of two things happen: either \( W \) rejects a proposal from \( M \); or, \( W \) accepts a proposal from \( M \), but subsequently breaks it off by accepting another man’s proposal.

**The Gale-Shapley algorithm:** We present the G-S algorithm in pseudocode.

```plaintext
while there exists a single man:
{
    let \( M \) be any such man;
    \( M \) proposes to his favorite woman who has not already snubbed him;
    // we claim, and will show, that there is
    // always at least one woman who has not snubbed \( M \)
    this woman, \( W \), accepts if she is single, or if she prefers \( M \) to her current fiancee, otherwise she rejects;
}
marry all engaged pairs.

END
```

In fact, these pseudocode instructions are not quite sufficiently simple to be considered primitive (unit-cost) steps, and their implementation requires further (simple) ideas, discussed in Kleinberg-Tardos.

Still, it should be clear that all the operations are at least well-defined, and each individual step can be executed in a “reasonable” number of computational steps (given the preference listings for the men and women as input data). Also, the implementation details are irrelevant to our current goal, which is to understand why this algorithm halts and produces a stable matching. We will do so by making a series of useful observations, recorded as Claims. (It can be helpful to work with numbered Claims or Lemmas to clearly organize your thoughts; this makes it easy to refer back to what you have already established. I recommend using the Theorem environments in LaTeX, which are fairly easy to learn.)

**Claim 1.** A woman, if she becomes engaged, will stay engaged thereafter.
Proof. An engagement is only broken when the woman involved accepts another proposal, so she remains engaged. □

Claim 2. For the single man, $M$, selected in the while-loop, there is always at least one woman $W$ who has not yet snubbed him.

Proof. A woman is always engaged immediately after she snubs $M$ (since a single woman accepts any proposal). By Claim 1, that woman stays engaged thereafter. So if every woman has snubbed $M$, every woman must be currently engaged. But this is impossible, since $M$ is single and there are the same number of men and women. □

Claim 3. Every man proposes to every woman at most once.

Proof. Imagine that $M$ proposes to $W$ twice. Then one of two things must have happened: either she rejected his proposal the first time, or she accepted it but later broke it off (causing $M$ to become single, which must have happened before he could make another proposal to anyone). In either case, $M$ was snubbed by $W$ before his second proposal to her. (This is just by our definition of snubbing.) But this is a contradiction, since the algorithm only allows $M$ to propose to women who have not already snubbed him. □

Theorem 1. The Gale-Shapley algorithm terminates and outputs a complete matching, after at most $N^2$ executions of its while-loop.

Proof. In every while-loop execution, a proposal is made, since the single man $M$ has at least one woman to ask who has not yet snubbed him (by Claim 2). Now by Claim 3, the same proposal can’t be made twice; and since there are only $N \cdot N = N^2$ possible distinct proposals, the while-loop can only execute $\leq N^2$ times.

By definition of the while-condition, when we exit the while-loop there are no more single men and therefore, no single women either. Thus the final step produces a complete matching. □

Theorem 1 above doesn’t tell us in an absolute sense how efficient the algorithm is, since we have not discussed the complexity of implementing all steps within the while-loop. Still, it is a great first step towards an efficiency analysis.

Theorem 2. The matching output by the Gale-Shapley algorithm (call it $\mathcal{M}$) is stable.
Proof. Consider any pair $M, W$ of man and woman; we will argue that $(M, W)$ do not form an eloping pair under $\mathcal{M}$. (Thus there are no eloping pairs, and the matching is stable.)

If $M$ and $W$ are married under $\mathcal{M}$ then they cannot form an eloping pair (by the definition). Similarly, if $M$ prefers his wife (call her $W'$) to $W$ then again, $(M, W)$ are not an eloping pair.

So assume $M$ prefers $W$ to $W'$. But by the design of the algorithm, this means that $M$ must have proposed to $W$ before proposing to $W'$. But the pair $(M, W)$ did not end up married, so $W$ must have snubbed $M$ at some point.

Immediately after doing so, $W$ was engaged to someone she preferred to $M$. Also, each step of the algorithm can only increase or maintain a woman’s “satisfaction” with her fiancee. Thus her final fiancee, who becomes her husband, must also be preferred by $W$ over $M$. Thus, again, we find that $(M, W)$ are not an eloping pair under $\mathcal{M}$. We have considered all possible cases, so we can say unconditionally that $(M, W)$ are not eloping. As discussed earlier, this completes the proof of the Theorem. □