Lecture 12.4

Gale-Shapley Theorem

Theorem: Every instance of the stable matching problem has a solution i.e. a stable matching.

Proof: Men-proposing deferred acceptance algorithm.

- Initially all men and women are unmatched.
- $M \leftarrow \phi$
- While there exists a man m who is unmatched and has not proposed all women {

- Let w be the most preferred woman of m whom he has not proposed yet.
- man m proposes woman w
- If w was unmatched then w accepts m's proposal. We have $M \leftarrow M \cup \{(m,w)\}$
- Else if w is matched with m' in M, but $m \succ_w m'$, then $M \leftarrow (M \setminus \{(m',w)\}) \cup \{(m,w)\}$

}

- Return m.

proof of correctness:- The while loop can terminate in two ways: (1) all men (and thus all women) are matched

(2) there exists a man who is unmatched, but he has proposed to all the women.

why while loop can not terminate because of (2)?

Observe: Any woman who has received a proposal, always remains matched in the successive iterations of the algorithm although she can change her partner.

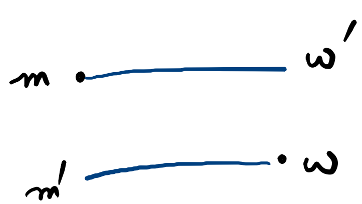
If there exists any man who has proposed to all women, then every woman has received at least one proposal. And thus all women are matched. But then all men are also matched.

The algorithm always outputs a perfect matching.

To prove: the output matching is stable.

That is there is no blocking pair.

For the sake of finding a contradiction, let us assume that there exists a blocking pair (m, w) for the output matching M.



$$w \succ_m M(m) = w'$$

$$\boxed{m \succ_w M(w) = m'}$$

The man m must have proposed w'. Since $w \succ_m w'$, the man m must have proposed woman w

in some iteration i. After i-th iteration the woman w remains matched. Moreover the partner of the woman w from (i+1)-th iteration is at least as preferable as m to the woman w. Hence, the final partner $M(w) = m'$ is more preferred to the woman w than m. But for (m, w) to be a blocking pair, we assumed $m >_w m'$. This is a contradiction

$\blacksquare$

Runtime: The while can iterate for $O(n(n-1)+1) = O(n^2)$ ◻