Stable matching

September 11, 2017

Overview

1. Stable matching

Stable matching

- We will begin by looking at an algorithm that demonstrates many of the ideas we will encounter during our course.
- This algorithm is due to Gale and Shapley (David Gale and Lloyd Shapley $\sim$ 1962).
  Actually, it was in use before they published it.

Stable matching

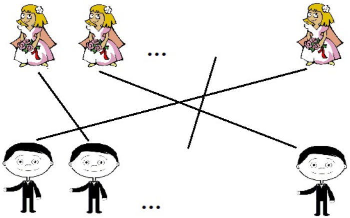
- We will begin by looking at an algorithm that demonstrates many of the ideas we will encounter during our course.
- This algorithm is due to Gale and Shapley (David Gale and Lloyd Shapley ~ 1962).
  Actually, it was in use before they published it.

Stable matching

- We begin with a simple version of the problem.
- The problem: Given a set of n women $\{w_1, ..., w_n\}$ and a set of n men $\{m_1, ..., m_n\}$, we want to find a pairing so that each women and each men can get married. We want the matching we find to be stable.
- What do we mean by a stable matching?

Stable matching

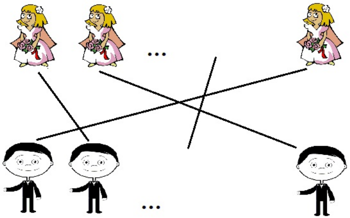
- We begin with a simple version of the problem.
- The problem: Given a set of n women $\{w_1, \ldots, w_n\}$ and a set of n men $\{m_1, \ldots, m_n\}$, we want to find a pairing so that each women and each men can get married. We want the matching we find to be stable.



- What do we mean by a stable matching?

Stable matching

- We begin with a simple version of the problem.
- The problem: Given a set of n women $\{w_1, ..., w_n\}$ and a set of n men $\{m_1, ..., m_n\}$, we want to find a pairing so that each women and each men can get married. We want the matching we find to be stable.



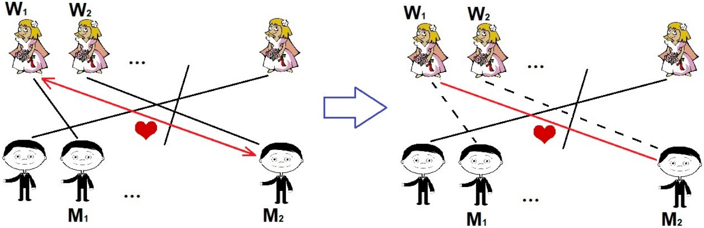
- What do we mean by a stable matching?

Stable matching

- Each woman has a list of men ordered according to her preferences.
  Similarly, each man has a list of women ordered according to his preferences.
- Assume in our matching, some women w_1 is married to w_2 but she would prefer to be married to a men m_2 who's married to a women w_2.
  If m_2 also prefer w_1 over w_2, then the matching we have is not stable, since it is likely that w_1 and m_2 will leave their partners and get married.

Stable matching

- Each woman has a list of men ordered according to her preferences.
  Similarly, each man has a list of women ordered according to his preferences.
- Assume in our matching, some women w_1 is married to w_2 but she would prefer to be married to a men m_2 who's married to a women w_2.
  If m_2 also prefer w_1 over w_2, then the matching we have is not stable, since it is likely that w_1 and m_2 will leave their partners and get married.

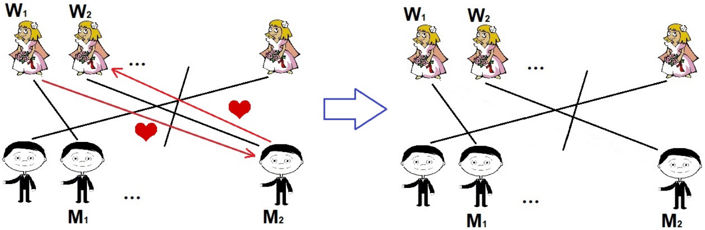


Stable matching

- We see that we would like to find a matching such that for **every** women w we have:
  If m is a men that w would prefer to marry (over the man she is currently married to), then m prefers his current wife over w.
- if this is the case, the matching is stable.

Stable matching

- We see that we would like to find a matching such that for **every** women w we have:
  If m is a men that w would prefer to marry (over the man she is currently married to), then m prefers his current wife over w.
- if this is the case, the matching is stable.



- More formally:
  For sets W and M of n women and n men, a perfect matching $S \subseteq W \times M$ is a set of n pairs of the form (w, m) with $w \in W$ and $m \in M$, such that each $w \in W$ belongs to exactly one pair and each $m \in M$ belongs to exactly one pair.
- If there are 2 pairs $(w, m), (w', m') \in S$ such that w prefers m' over m and m' prefers w over w' then we say that (m, w') is an instability with respect to S.

- More formally:
  For sets W and M of n women and n men, a perfect matching $S \subseteq W \times M$ is a set of n pairs of the form (w, m) with $w \in W$ and $m \in M$, such that each $w \in W$ belongs to exactly one pair and each $m \in M$ belongs to exactly one pair.
- If there are 2 pairs $(w, m), (w', m') \in S$ such that w prefers m' over m and m' prefers w over w' then we say that (m, w') is an instability with respect to S.

Stable matching

- Our goal:
  Given W, M and the lists of preferences, we want to find a perfect matching S such that there are **no** instabilities.
- Can we do it?
  Can we do it efficiently?

Stable matching

- Our goal:
  Given W, M and the lists of preferences, we want to find a perfect matching S such that there are **no** instabilities.
- Can we do it?
  Can we do it efficiently?

Stable matching

- We will show that there exists a sable matching.
- We will do it by giving an efficient algorithm that finds such stable matching.

Stable matching

- We will show that there exists a sable matching.
- We will do it by giving an efficient algorithm that finds such stable matching.

Stable matching

Designing the Algorithm:

- The initial situation is that everyone are unmarried.
  At this point an unmarried woman w would chose the man m that is ranked highest in her list.
  w and m become engaged.
- In a situation where some women are free and some are engaged, a free woman w, proposes to the man m that she ranked highest among the men she haven't yet proposed to.
  If m is free they become engaged. If m is engaged they become engaged or not, according to m's preferences.
- The algorithm terminates when there are no free women.
  At this point all of the engagements become final (marriages).

Stable matching

Stable matching

Designing the Algorithm:

- The initial situation is that everyone are unmarried.
  At this point an unmarried woman w would chose the man m that is ranked highest in her list.
  w and m become engaged.
- In a situation where some women are free and some are engaged, a free woman w, proposes to the man m that she ranked highest among the men she haven't yet proposed to.
  If m is free they become engaged. If m is engaged they become engaged or not, according to m's preferences.
- The algorithm terminates when there are no free women.
  At this point all of the engagements become final (marriages).

Designing the Algorithm:

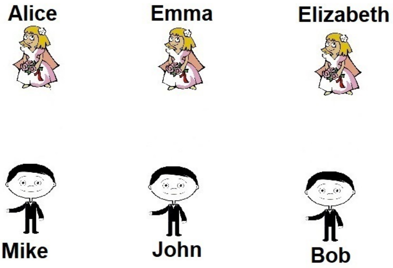
- The initial situation is that everyone are unmarried.
  At this point an unmarried woman w would chose the man m that is ranked highest in her list.
  w and m become engaged.
- In a situation where some women are free and some are engaged, a free woman w, proposes to the man m that she ranked highest among the men she haven't yet proposed to.
  If m is free they become engaged. If m is engaged they become engaged or not, according to m's preferences.
- The algorithm terminates when there are no free women.
  At this point all of the engagements become final (marriages).

Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	Mike	John	Bob
Emma	Mike	John	Bob
Elizabeth	John	Mike	Bob

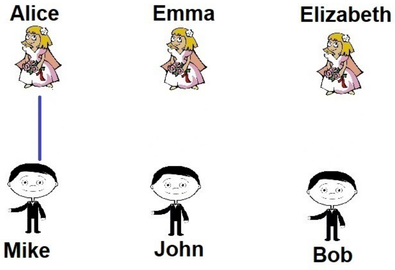


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	Mike	John	Bob
Emma	Mike	John	Bob
Elizabeth	John	Mike	Bob

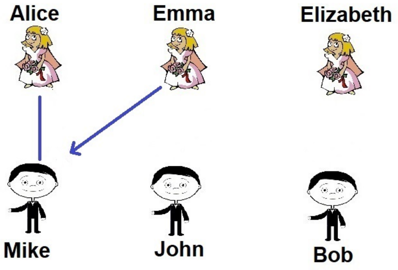


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	Mike	John	Bob
Emma	Mike	John	Bob
Elizabeth	John	Mike	Bob

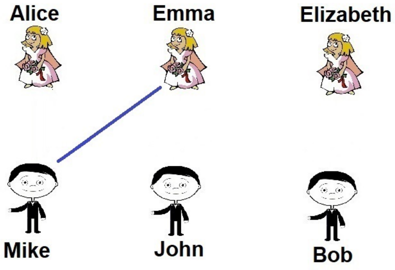


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	~~Mike~~	John	Bob
Emma	Mike	John	Bob
Elizabeth	John	Mike	Bob

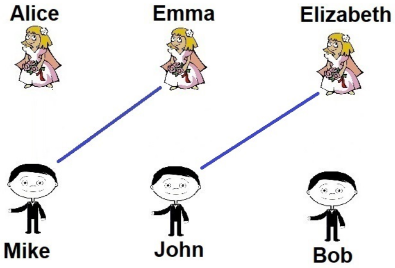


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	~~Mike~~	John	Bob
Emma	Mike	John	Bob
Elizabeth	John	Mike	Bob

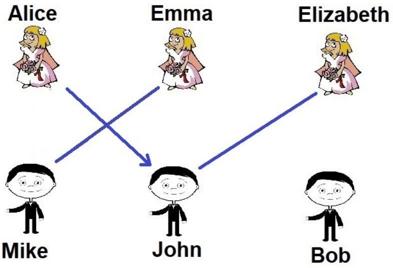


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	~~Mike~~	John	Bob
Emma	Mike	John	Bob
Elizabeth	John	Mike	Bob

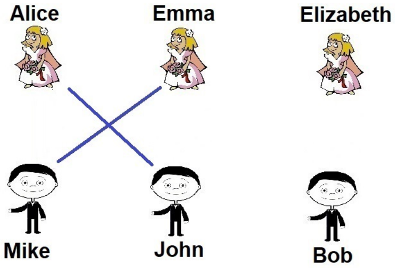


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	~~Mike~~	John	Bob
Emma	Mike	John	Bob
Elizabeth	~~John~~	Mike	Bob

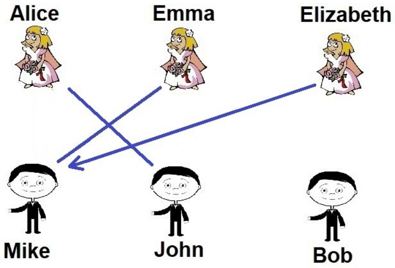


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	~~Mike~~	John	Bob
Emma	Mike	John	Bob
Elizabeth	~~John~~	Mike	Bob

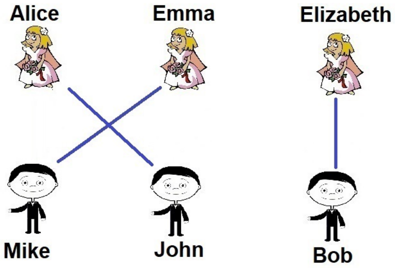


Stable matching

Preferences:

	Woman 1	Woman 2	Woman 3
Mike	Emma	Alice	Elizabeth
John	Alice	Elizabeth	Emma
Bob	Emma	Elizabeth	Alice

	Man 1	Man 2	Man 3
Alice	~~Mike~~	John	Bob
Emma	Mike	John	Bob
Elizabeth	~~John~~	~~Mike~~	Bob



- Note:
  1. It is not clear that this algorithm terminates.
  2. It is not clear that this algorithm returns a perfect matching.
  3. It is not clear that this algorithm returns a stable perfect matching.
  4. The algorithm is not fully defined. We did not specify how to chose the free woman.

- Note:
    1. It is not clear that this algorithm terminates.
    2. It is not clear that this algorithm returns a perfect matching.
    3. It is not clear that this algorithm returns a stable perfect matching.
    4. The algorithm is not fully defined. We did not specify how to chose the free woman.

Stable matching

- Note:
  1. It is not clear that this algorithm terminates.
  2. It is not clear that this algorithm returns a perfect matching.
  3. It is not clear that this algorithm returns a stable perfect matching.
  4. The algorithm is not fully defined. We did not specify how to chose the free woman.

Stable matching

- Note:
  1. It is not clear that this algorithm terminates.
  2. It is not clear that this algorithm returns a perfect matching.
  3. It is not clear that this algorithm returns a stable perfect matching.
  4. The algorithm is not fully defined. We did not specify how to chose the free woman.

Stable matching

- Note:
    1. It is not clear that this algorithm terminates.
    2. It is not clear that this algorithm returns a perfect matching.
    3. It is not clear that this algorithm returns a stable perfect matching.
    4. The algorithm is not fully defined. We did not specify how to chose the free woman.

- From the way we defined the algorithm, it is clear that from the moment a man m gets a marriage proposal for the first time, he remains engaged until the algorithm terminates.
- It is also clear that the woman to whom a man m is engaged, can only "improve" in terms m's preferences.

Stable matching

- From the way we defined the algorithm, it is clear that from the moment a man m gets a marriage proposal for the first time, he remains engaged until the algorithm terminates.
- It is also clear that the woman to whom a man m is engaged, can only "improve" in terms m's preferences.

- From the way we defined the algorithm it is clear that a woman w may be engaged or not engaged on different steps of the algorithm.
- It is also clear that for a woman w the sequence of men to which she is engaged can only get worse as the algorithm progresses.

Stable matching

- From the way we defined the algorithm it is clear that a woman w may be engaged or not engaged on different steps of the algorithm.
- It is also clear that for a woman w the sequence of men to which she is engaged can only get worse as the algorithm progresses.

- The facts that we mentioned suggest that the algorithm we describe has some "progress" in each step.
(You may think of it as if the algorithm has a direction in which it progresses.)
- Theorem: The Gale-Shapley algorithm terminates after at most n^2 steps.
(By a step we mean a proposal of a woman to a man and the decision of this man according to his preferences.)

Stable matching

- The facts that we mentioned suggest that the algorithm we describe has some "progress" in each step.
  (You may think of it as if the algorithm has a direction in which it progresses.)
- Theorem: The Gale-Shapley algorithm terminates after at most n^2 steps.
  (By a step we mean a proposal of a woman to a man and the decision of this man according to his preferences.)

Stable matching

Stable matching

- Proof:
  In each step of the algorithm some woman proposes to some man to whom she never proposed before.
  We will denote the number of iterations by t.
  If we denote by $P(t)$, the set of pairs (w, m) such that w proposed to m by the end of iteration t, then it is clear that for all t we have:

$$P(t+1) > p(t)$$

  Since there are only n^2 possible pairs (w, m), the algorithm must terminate after at most n^2 iterations. ■

Stable matching

Stable matching

- Next we want to show that the set of pairs that the algorithm returns is a perfect matching.
- What do we need to show?
- The only way a woman can remain unpaired at the end of the algorithm, is if she went over all of her list and none of the men agreed to get engaged to her.
  (Some might agree to get engaged but later reject her.)
- We will show:
  If a woman w is free at some point during the algorithm, then there is a man to which w hasn't yet proposed.

Stable matching

- Next we want to show that the set of pairs that the algorithm returns is a perfect matching.
- What do we need to show?
- The only way a woman can remain unpaired at the end of the algorithm, is if she went over all of her list and none of the men agreed to get engaged to her.
  (Some might agree to get engaged but later reject her.)
- We will show:
  If a woman w is free at some point during the algorithm, then there is a man to which w hasn't yet proposed.

Stable matching

Stable matching

- Next we want to show that the set of pairs that the algorithm returns is a perfect matching.
- What do we need to show?
- The only way a woman can remain unpaired at the end of the algorithm, is if she went over all of her list and none of the men agreed to get engaged to her.
  (Some might agree to get engaged but later reject her.)
- We will show:
  If a woman w is free at some point during the algorithm, then there is a man to which w hasn't yet proposed.

Stable matching

- Next we want to show that the set of pairs that the algorithm returns is a perfect matching.
- What do we need to show?
- The only way a woman can remain unpaired at the end of the algorithm, is if she went over all of her list and none of the men agreed to get engaged to her.
  (Some might agree to get engaged but later reject her.)
- We will show:
  If a woman w is free at some point during the algorithm, then there is a man to which w hasn't yet proposed.

Stable matching

- Proof: Assume at some point during the execution of the algorithm, a woman w is free and had already proposed to all the men.
  It follows that all the men are engaged at this point. (This follows from the fact that after a man gets a proposal he remains engaged.)
  However, since the set of engaged pairs forms a matching, it follows that there are n engaged men and n engaged women. This is a contradiction. ■

- Corollary: The set of pairs that the algorithm returns is a perfect matching.
- Proof: Clearly, the set of engaged pairs is a matching at each point in the algorithm. The only way for the algorithm to terminate with a free woman is if this woman had proposed to all the men. However, by what we showed, for a free woman there must be a man to whom she did not propose. ■

- Corollary: The set of pairs that the algorithm returns is a perfect matching.
- Proof: Clearly, the set of engaged pairs is a matching at each point in the algorithm. The only way for the algorithm to terminate with a free woman is if this woman had proposed to all the men. However, by what we showed, for a free woman there must be a man to whom she did not propose. ■

Stable matching

- It remains to show that the perfect matching that G-S algorithm produces is a stable matching.
- Theorem: If S is the perfect matching returned by the G-S algorithm, then S is a stable matching.

- It remains to show that the perfect matching that G-S algorithm produces is a stable matching.
- Theorem: If S is the perfect matching returned by the G-S algorithm, then S is a stable matching.

- Proof: If S is not a stable matching then there are 2 pairs (w, m) and (w', m') which causes an instability with respect to S.
  For these 2 pairs we have:
  1. w prefers m' over m
  2. m' prefers w over w'

  From the fact that w prefers m' over m it follows that w proposed to m' before it proposed to m. Since w ends up with a different man them m', it follows that (at some point) w was rejected by m' for another woman w'' that m' prefers over w.
  By what we said, m' can only end up with a woman he prefers over w''. However he ends up with w' which he like less than w which he likes less than w''. This is a contradiction. ■

- As we mentioned, the algorithm can be executed in different ways. We have a choice of which free woman will propose.
- We may ask:
  Do we get the same perfect matching for any possible execution of the algorithm?
- We will show that the answer to this question is positive.
- The way we will prove that we always get the same perfect matching, is by showing that regardless of how we apply the algorithm, every woman ends up with the best possible match for her.

Stable matching

- As we mentioned, the algorithm can be executed in different ways. We have a choice of which free woman will propose.
- We may ask:
  Do we get the same perfect matching for any possible execution of the algorithm?
- We will show that the answer to this question is positive.
- The way we will prove that we always get the same perfect matching, is by showing that regardless of how we apply the algorithm, every woman ends up with the best possible match for her.

Stable matching

- As we mentioned, the algorithm can be executed in different ways. We have a choice of which free woman will propose.
- We may ask:
  Do we get the same perfect matching for any possible execution of the algorithm?
- We will show that the answer to this question is positive.
- The way we will prove that we always get the same perfect matching, is by showing that regardless of how we apply the algorithm, every woman ends up with the best possible match for her.

Stable matching

- As we mentioned, the algorithm can be executed in different ways. We have a choice of which free woman will propose.
- We may ask:
  Do we get the same perfect matching for any possible execution of the algorithm?
- We will show that the answer to this question is positive.
- The way we will prove that we always get the same perfect matching, is by showing that regardless of how we apply the algorithm, every woman ends up with the best possible match for her.

Stable matching

Stable matching

Stable matching

- We say that a man m is valid for a woman w if there exists a stable matching that contains the pair (w, m).
- We say that a man m is the best valid partner for a woman w, if for any other valid man m', the woman w prefers m over m'.
  We will denote the best valid partner of w by $best(w)$.

Stable matching

- We say that a man m is valid for a woman w if there exists a stable matching that contains the pair (w, m).
- We say that a man m is the best valid partner for a woman w, if for any other valid man m', the woman w prefers m over m'.
  We will denote the best valid partner of w by $best(w)$.

- We define the following set of pairs:

$$\hat{S} = \{(w, \text{best}(w)) | w \in W\}$$

- We will prove:
  Theorem: For any possible execution of the G-S algorithm, the resulted matching equals the set $\hat{S}$.
- This shows:
  1. $\hat{S}$ is a stable perfect matching.
  2. The result of the G-S algorithm is the best result for all the women simultaneously.
  3. The result of the G-S algorithm is always the same.

Stable matching

- We define the following set of pairs:

$$\hat{S} = \{(w, \text{best}(w)) | w \in W\}$$

- We will proceed:
  Theorem: For any possible execution of the G-S algorithm, the resulted matching equals the set $\hat{S}$.
- This shows:
  1. $\hat{S}$ is a stable perfect matching.
  2. The result of the G-S algorithm is the best result for all the women simultaneously.
  3. The result of the G-S algorithm is always the same.

Stable matching

Stable matching

- We define the following set of pairs:

$$\hat{S} = \{(w, \text{best}(w)) | w \in W\}$$

- We will prove:
  Theorem: For any possible execution of the G-S algorithm, the resulted matching equals the set $\hat{S}$.
- This shows:
  1. $\hat{S}$ is a stable perfect matching.
  2. The result of the G-S algorithm is the best result for all the women simultaneously.
  3. The result of the G-S algorithm is always the same.

- We define the following set of pairs:

$$\hat{S} = \{(w, \text{best}(w)) | w \in W\}$$

- We will prove:
  Theorem: For any possible execution of the G-S algorithm, the resulted matching equals the set $\hat{S}$.
- This shows:
  1. $\hat{S}$ is a stable perfect matching.
  2. The result of the G-S algorithm is the best result for all the women simultaneously.
  3. The result of the G-S algorithm is always the same.

Stable matching

Stable matching

- We define the following set of pairs:

$$\hat{S} = \{(w, \text{best}(w)) | w \in W\}$$

- We will prove:
  Theorem: For any possible execution of the G-S algorithm, the resulted matching equals the set $\hat{S}$.
- This shows:
  1. $\hat{S}$ is a stable perfect matching.
  2. The result of the G-S algorithm is the best result for all the women simultaneously.
  3. The result of the G-S algorithm is always the same.

Stable matching

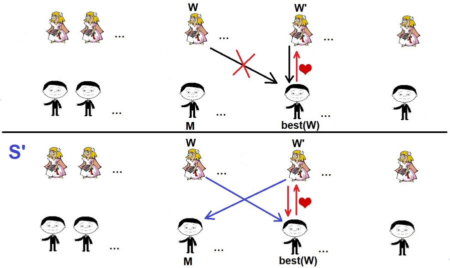
- We define the following set of pairs:

$$\hat{S} = \{(w, \text{best}(w)) | w \in W\}$$

- We will prove:
  Theorem: For any possible execution of the G-S algorithm, the resulted matching equals the set $\hat{S}$.
- This shows:
  1. $\hat{S}$ is a stable perfect matching.
  2. The result of the G-S algorithm is the best result for all the women simultaneously.
  3. The result of the G-S algorithm is always the same.

Stable matching

- Proof: Assume the theorem is false. It follows that there exists an execution of the algorithm such that some woman is paired with a man that is not her best possible partner.
  Since the women propose to the men in decreasing order of their preferences, the fact that some woman w end up with a man which is not her best possible match, it means that at some point, w is rejected by best(w).
  Consider the first point in the execution of the algorithm where some woman w is rejected by a valid partner. It must be that w is rejected by $best(w)$. At this point, $best(w)$ is either engaged to a woman he prefers over w, or he is engaged to w and has been proposed to, by a woman he prefers over w. We denote the woman that best(w) prefers over w by w'.
  Since best(w) is a valid partner of w, it follows that there exists a stable matching S' that has the pair $(w, \text{best}(w))$. In S', the woman w' is paired with some $m \neq \text{best}(w)$. Since we are considering the first rejection by a valid partner in the execution of the algorithm, it follows that at this point w' had not been rejected by a valid partner. Since w' proposes in decreasing order and m is a valid partner of w', it follows that w' prefers best(w) over m.
  On the other hand, we know that best(w) prefers w' over w.
  It follows that the 2 pairs: $(w, \text{best}(w))$ and (w', m) are an instability in S'. This is a contradiction.
  ■



Stable matching

- We may define the analog notions of valid partner and worst partner of a men.
- We can show:
  Theorem: The G-S algorithm pairs each man with his worst partner.

Stable matching

- We may define the analog notions of valid partner and worst partner of a men.
- We can show:
  Theorem: The G-S algorithm pairs each man with his worst partner.

Stable matching

- <u>Proof:</u> We know that the result of G-S algorithm is $\hat{S}$.
  If there is a pair (w, m) in $\hat{S}$ such that $w \neq \text{worst}(m)$, then there is a stable matching S' in which m is paired with $\text{worst}(m)$ and w is paired with some $m' \neq m$.
  In S': $(\text{worst}(m), m), (w, m')$.
  But m is the best for w and clearly m prefers w over $\text{worst}(m)$. It follows that the above is an instability in S'.
  This is a contradiction. $\blacksquare$

Stable matching

- Some questions to think of:
  1. Is it true that there always exists a stable matching such that someone gets her/his first choice ?
  2. Do we have a unique stable matching? How can we tell if the stable matching found by the G-S algorithm is unique?
  3. How do we generalize the algorithm?
     (For example if there are n applicants to m jobs.)

Stable matching

- Some questions to think of:
  1. Is it true that there always exists a stable matching such that someone gets her/his first choice ?
  2. Do we have a unique stable matching? How can we tell if the stable matching found by the G-S algorithm is unique?
  3. How do we generalize the algorithm?
     (For example if there are n applicants to m jobs.)

Stable matching

- Some questions to think of:
  1. Is it true that there always exists a stable matching such that someone gets her/his first choice ?
  2. Do we have a unique stable matching? How can we tell if the stable matching found by the G-S algorithm is unique?
  3. How do we generalize the algorithm? (For example if there are n applicants to m jobs.)

Stable matching

- Some questions to think of:
  1. Is it true that there always exists a stable matching such that someone gets her/his first choice ?
  2. Do we have a unique stable matching? How can we tell if the stable matching found by the G-S algorithm is unique?
  3. How do we generalize the algorithm?
     (For example if there are n applicants to m jobs.)