Here is a concrete description of the Gale-Shapley algorithm, which depicts a state of the algorithm.

Initially, all $m \in M$ and $w \in W$ are free.

While there is a man $m$ who is free and hasn't proposed to every woman:

- Choose such a man $m$.
- Let $w$ be the highest-ranked woman in $m$'s preference list to whom $m$ has not yet proposed.
- If $w$ is free then,
  - $(m, w)$ become engaged
- Else $w$ is currently engaged to $m'$.
  - If $w$ prefers $m'$ to $m$ then
    - $m$ remains free
  - Else $w$ prefers $m$ to $m'$
    - $(m, w)$ become engaged
    - $m'$ becomes free
- Endif

Endif

Endwhile

Return the set $S$ of engaged pairs.

**Def:** $w$ is a valid partner to $m$ $\iff$ $(m, w)$ is a stable matching with $(m, w)$

$w$ is a best valid partner to $m$ $\iff$ $w$ a valid partner and

$\forall w': w$ a valid partner to $m$

$m$ ranks $w$ above $w'$.

[why?]
\[ S^* = \{(b, \text{best}(b))\} \quad \text{[matching? why?]} \]

**CLAIM:** \( G - S \) returns \( S^* \)
- \( S^* \) is a stable perfect matching
- Order in execution of \( G - S \) doesn't matter

**proof:** (by contradiction)

Suppose not.

Then \( \exists E, \) execution of \( G - S \) where some \( b \) is matched with some girl \( h \neq \text{best}(b) \)

\[ \Rightarrow \exists \text{ time } t \]

in execution \( E \) which is the first instance that \( b \) is rejected by a valid partner \( g \)

In fact \( g = \text{best}(b) \)

[Why?]

Boys propose in decreasing order of preference

[Why is there a rejection by a valid partner?]

**REJECTION**

- at proposal;
  - \( b \) engaged to \( b' \);
  - someone she rates higher
- at proposal \( b, g \) engaged;
  - but at time \( t \) \( g \) gets a proposal by someone she rates higher \((b')\)

Now \( g \) is a valid partner of \( b \) - so \( \exists \) stable matching \( S' \) with \((b, g)\)

[Is \( S' \) produced by \( G - S \)?]

In \( S' \), \( b' \) has a partner \( g' \).

\((b' , g') \) in \( S' \)

\((b, g)\)

In \( E \) \( b \) gets rejected by \( g \), in favor of \( b' \)

So \( b' \) must rank \( g \) over \( g' \) (otherwise \( b' \) would have proposed to \( g' \) first & must have been rejected by her, in order for \( b' \) get engaged to \( g - X \))

And \( g \) prefers \( b' \) to \( b \)

So \( S' \) is unstable \( X \)