| M | PrefM |
|---|---|
| A | X Y V W Z |
| B | V X W Y Z |
| C | V Z W Y X |
| D | W V X Z Y |
| E | X Y V W Z |

| W | PrefW |
|---|---|
| V | A D C E B |
| W | A B D C E |
| X | D E C A B |
| Y | C B A E D |
| Z | A B D E C |

~~(A,X)~~ (A,Y)
~~(B,V)~~ (B,W)
~~(C,V)~~
~~(D,W)~~ (D,V)
(E,X)

**Maintenance (matches improve for welders)**

```
FreeM <- M
FreeW <- W
Invitations <- {}
Tentative <- {}
k <- 0

while there is an m in FreeM s.t. there is a w s.t. (m,w) not in Invitations
    choose such an m
    let w be m's highest ranked s.t. (m,w) not in Invitations

    add (m,w) to Invitations

    if w in FreeW then
        remove w from FreeW
        remove m from FreeM
        add (m,w) to Tentative
    else
        find m' s.t. (m', w) in Tentative
        if w prefers m to m'
            remove m from FreeM
            add m' to FreeM
            remove (m', w) from Tentative
            add(m, w) to Tentative
    k <- k+1
return Tentative
```

Suppose INV is T before loop and $\exists m \in FreeM, w$ s.t. $(m,w) \notin Invites$

Notation: $m$ s.t. $(m,w) \in Tent$ (or NIL)   w's highest pref $\hat{m}$ s.t. $(\hat{m},w) \in Invites$

f) $\forall w,\ MatchW(w) = \max_{w\,(\hat{m},w) \in Invites} \hat{m}$

every w is matched with their favorite machinist they've received an invitation from

Only change to Tent / Invites is for w (so if true before loop for other welders, still true after)

$MatchW_{old}(w) = \max_{w\,(m,w) \in Invites_{old}} m$ (INV)

case 1: $w \in FreeW_{old}$

$\neg \exists m$ s.t. $(m,w) \in Invites_{old}$ (INV b)

$MatchW_{new}(w) = m$ (code - add (m,w) to Tent)

$\max_{w\,(\hat{m},w) \in Invites_{new}} \hat{m} = \max_{w\,(\hat{m},w) \in Invites_{old} \cup \{(m,w)\}} \hat{m}$

$= m$ (m is only term in )

case 2: $w \notin FreeW_{old}$

$(m',w) \in Tent_{old}$

case a) w prefers m to m'

m is better than prev. best inviter   $m \underset{\text{preferred by } w}{>_w} m' = \max_{w\,(\hat{m},w) \in Inv_{old}} \hat{m}$

m is new best inviter   $m = \max_{w\,(\hat{m},w) \in Inv_{old} \cup \{(m,w)\}} \hat{m} = \max_{w\,(\hat{m},w) \in Inv_{new}} \hat{m}$

w is paired with m   $\underbrace{(m,w) \in Tent_{new}}_{\text{code}}$, so $MatchW_{new}(w) = m$ ✓

b) w prefers m' to m

m is not better than prev. best inviter   $m <_w m' = \max_{w\,(\hat{m},w) \in Inv_{old}} \hat{m}$

m' is still best inviter   $m' = \max_{w\,(\hat{m},w) \in Inv_{old} \cup \{(m,w)\}} \hat{m} = \max_{w\,(\hat{m},w) \in Inv_{new}} \hat{m}$

w still paired with m   $(m',w) \in Tent_{new}$, so $MatchW_{new}(w) = m'$ ✓

$(m',w) \in Tent_{old}$;
code doesn't modify in this case

**Maintenance (hard part)**

```
FreeM <- M
FreeW <- W
Invitations <- {}
Tentative <- {}
k <- 0

while there is an m in FreeM s.t. there is a w s.t. (m,w) not in Invitations
    choose such an m
    let w be m's highest ranked s.t. (m,w) not in Invitations

    add (m,w) to Invitations

    if w in FreeW then
        remove w from FreeW
        remove m from FreeM
        add (m,w) to Tentative
    else
        find m' s.t. (m', w) in Tentative
        if w prefers m to m'
            remove m from FreeM
            add m' to FreeM
            remove (m', w) from Tentative
            add(m, w) to Tentative
    k <- k+1
return Tentative
```

Suppose INV is T before loop and $\exists m \in FreeM$, $w$ s.t. $(m,w) \notin Invites$

c) Tent is a matching and stable (restricted to matched m, w)

No instabilities in $Tent_{old}$

Suppose $(\hat{m}, \hat{w})$ is an instability wrt $Tent_{new}$ but not wrt $Tent_{old}$

So either a) $\hat{m}$'s match got worse (code never does this in a single iteration)

b) $\hat{w}$'s match got worse (code only improves a welder's match)

c) $\hat{w}$ was added to Tent and $\hat{m}$ would like to switch

d) $\hat{m}$ was added to Tent and $\hat{w}$ would like to switch


Suppose $(\hat{m}, \hat{w})$ is an instability wrt $Tent_{new}$

Case c: $\hat{w} \in FreeW_{old}$, $\hat{w} \notin FreeW_{new}$

$\hat{m} \neq m$ ($\hat{m} \notin FreeM_{old}$, $m \in FreeM_{old}$)

find $w'$ s.t. $(\hat{m}, w') \in Tent_{old}$

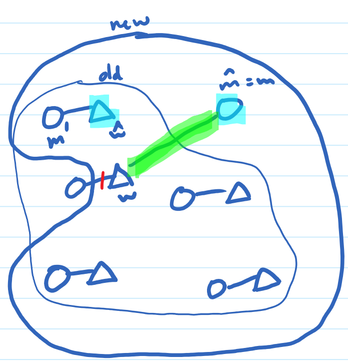
$\hat{m}$ prefers $\hat{w}$ to $w'$ (def instability)

$(\hat{m}, w') \in Invites_{old}$ [$Tent \subseteq Invites$]

$(\hat{m}, \hat{w}) \in Invites_{old}$

$\hat{w} \notin FreeW_{old}$ (INV b)

contradiction




Case d: $\hat{m} \in FreeM_{old}$ and $\hat{m} \notin FreeM_{new}$

$m = \hat{m}$ (only M that is made not free in code)

$w \neq \hat{w}$

$\hat{w} \notin FreeW_{old}$

let $m'$ be $m'$ s.t. $(m', \hat{w}) \in Tent_{old}$
so $MatchW_{old}(\hat{w}) = m'$
$MatchW_{new}(\hat{w}) = m'$

$\hat{w}$ prefers $\hat{m}^{=m}$ to $m'$

$m$ prefers $\hat{w}$ to $w$

$\| \; \hat{m}$

$(m, \hat{w}) \in Invites_{old}$

$(m', \hat{w}) \in Invites_{old}$

$\hat{w}$ prefers $m'$ to $m$ (Inv f)

$(m, \hat{w}) \in \text{Invites}_{old}$

$(m', \hat{w}) \in \text{Invites}_{old}$

$\hat{w}$ prefers $m'$ to $m$ (Inv f)

contradiction

# Termination

Termination: After $n^2$ iterations, $|Invites| = n^2$ and so $= M \times W$.
Then there is no $m$ (let alone $m \in FreeM$) s.t. $\exists w$ s.t. $(m,w) \notin Invites$

Post-condition: Suppose loop has terminated (so guard is false)
Then Tent is a perfect matching
Proof: suppose not - then $\exists$ unmatched $m$ and unmatched $w$
$m \in FreeM$, $w \in FreeW$
$(m,w) \notin Invites$ (INV b)
guard is T (contradiction)
$\therefore$ Tent is a perfect matching

Tent is stable (INV c)