max matching - Blossom's algorithm

Finding max matching

→ start with $m = \phi$

→ Find aug path P in (G, m)

→ if $P == [\ ]$, return m, as max matching

→ else $m = m \oplus P$

Finding augmenting path in (G, m)

U = unmatched vertices

creating a forest with each tree being just a root

for each $v \in U$
create a tree with root(v)
label v as EVEN label

Forest F

mark each edge in $E \setminus m$ as unexplored, each edge in m as explored.

will contain even labeled vertices

Queue = $\tilde{V}$ = vertices to be explored = $\cup$

while $\tilde{V}$ is not empty :

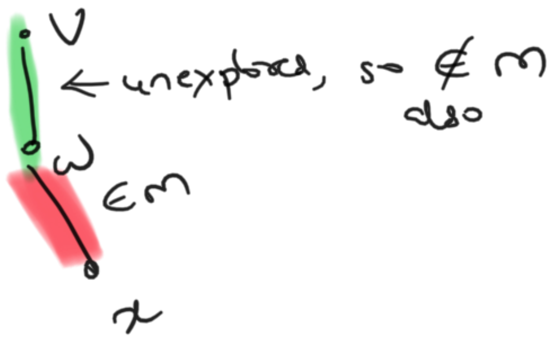
$v = \tilde{V}$.pop ( )

for every unexplored edge

$v - w$

$w \notin F$

(so w matched vertex)



* label w odd
* label x even
* add $v-w$, $w-x$ to tree (v) (and hence to F)
* so $w, x \in F$ also now
* mark parent(x) = w
  parent(w) = v
* add x to $\tilde{V}$.

$w \in F$

w label even

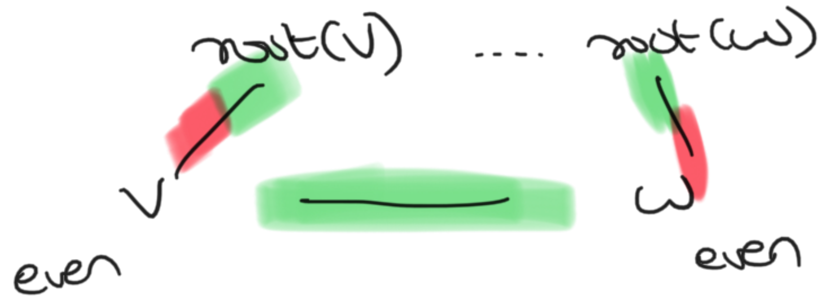
Do *

w label odd

Do nothing

mark $v - w$ as explored.

w has label even

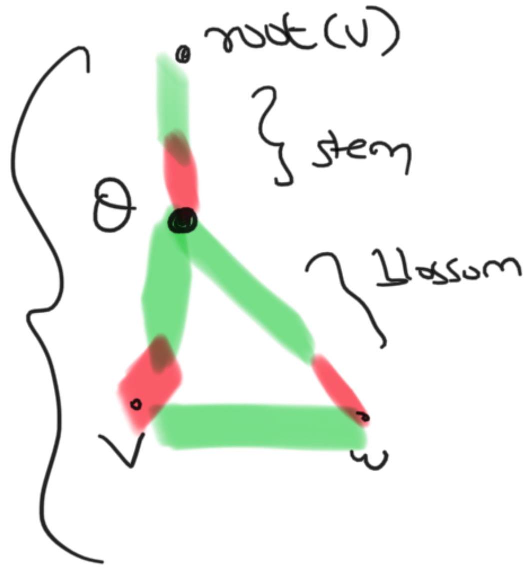


root(v) $\neq$ root(w)

root(v) = root(w)

$\boxed{\text{aug path found}}$

root(v) —— v — w —— root(w) !



Bloom found

θ = least common ancestor of v,w

B = blossom

=

$P' =$ Find aug path in $(G/B, M/B)$

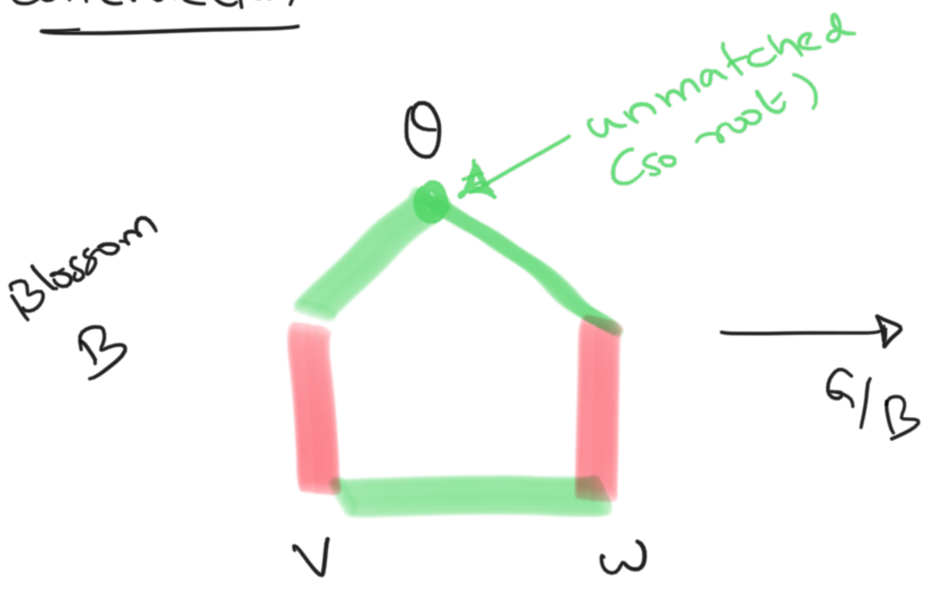
where $B \longrightarrow w$ ...

if $w \notin P'$, return P'

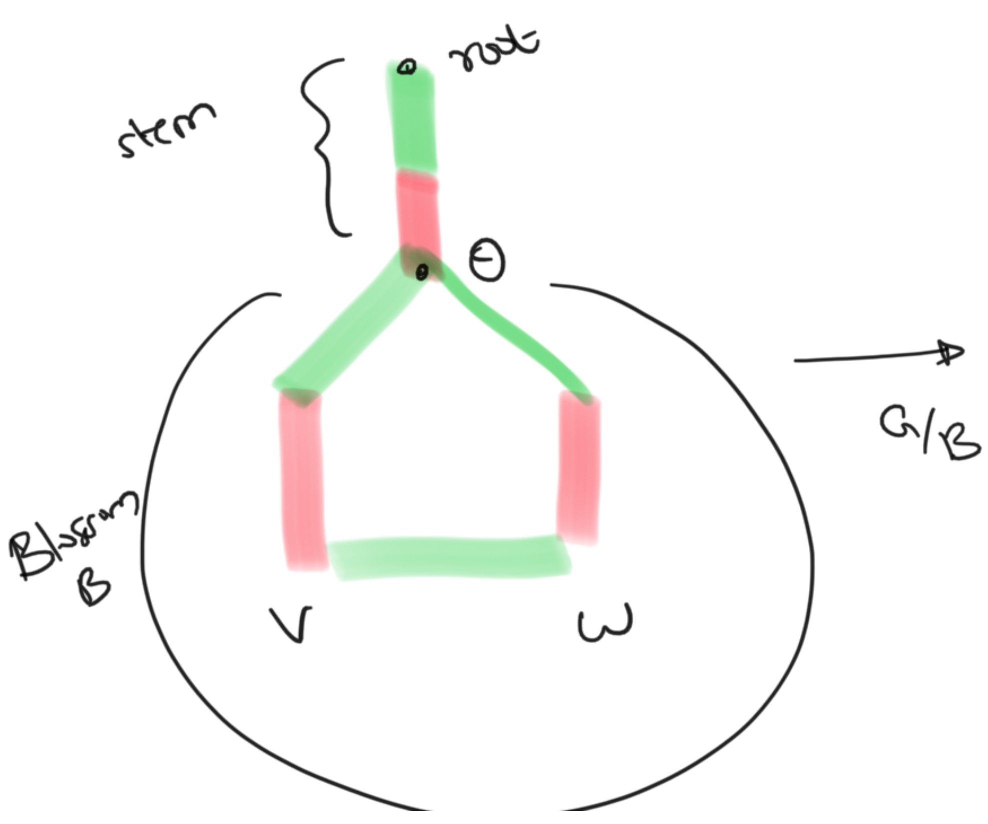
else $P =$ lift correct (P'), return P

more details

contraction



contract to unmatched vertex b



if you contract B,

...

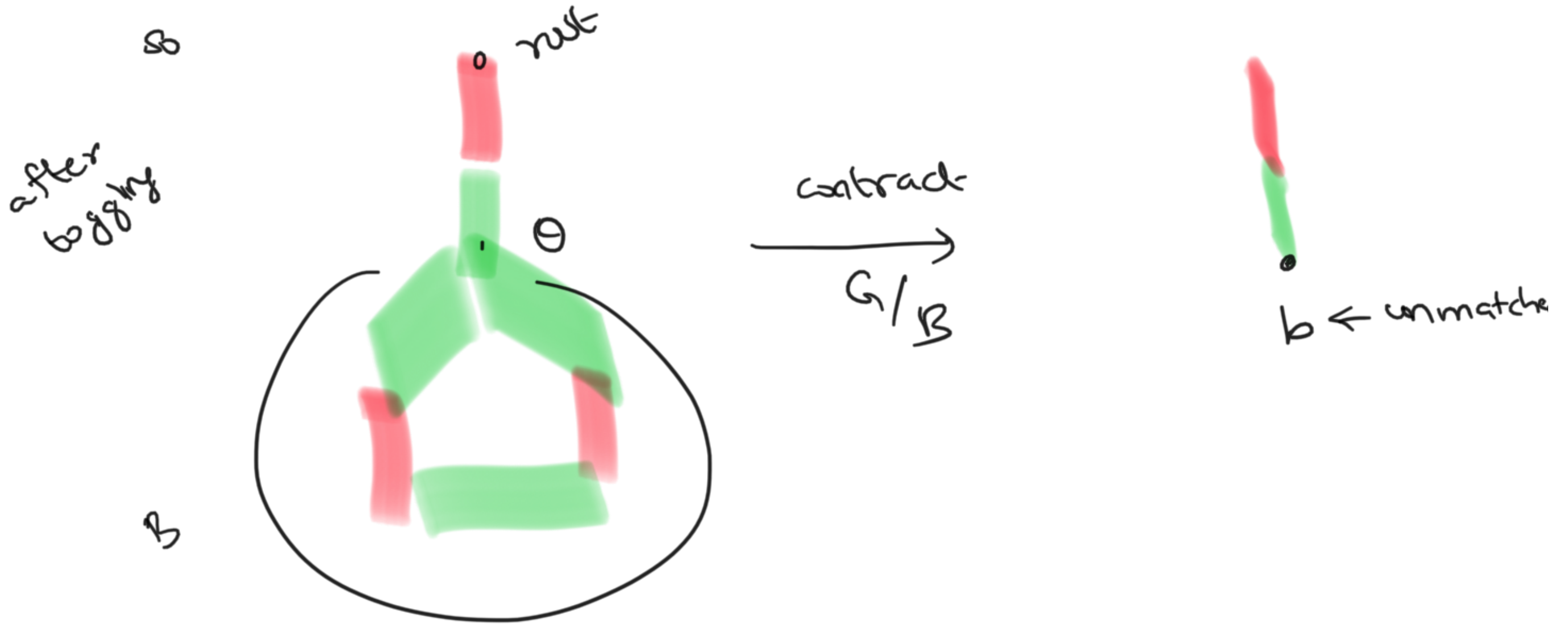
...

To avoid this, if you have a stem,

<u>change</u> the matching by

$$m = m \oplus \text{stem} \quad (\text{"toggling stem"})$$

so



Find aug path p' in G/B

if $p' \not\ni b$ — return p'

if $p' \ni b$ — since b unmatched it will have to be an end vertex of p'