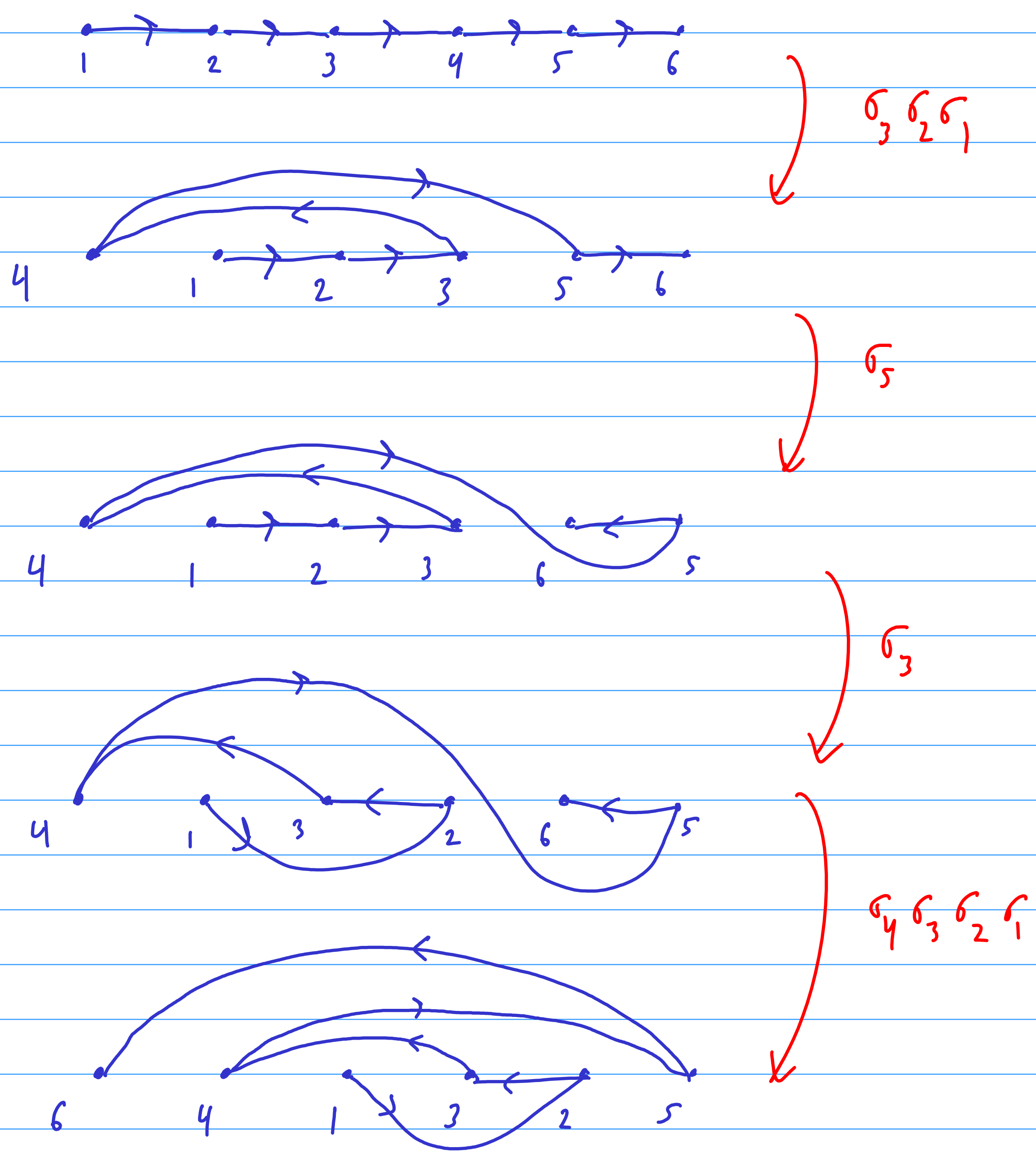
Lecture 26: The Bestvina-Handel algorithm

(from Toby Hall's notes)

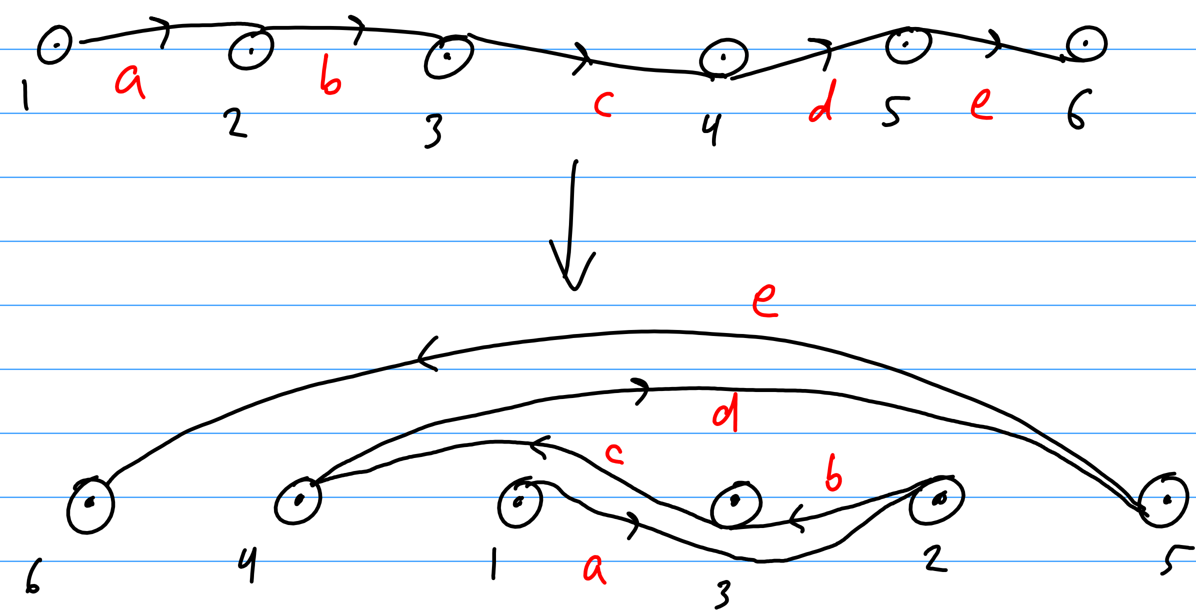
Consider the braid: $\sigma_3 \sigma_2 \sigma_1 \sigma_5 \sigma_3 \sigma_4 \sigma_3 \sigma_2 \sigma_1$

(σ defined counterclockwise for today)



all

Redo this with peripheral edges:



Graph map: $g: G \to G$

$g(a) = cd$ $\qquad$ $g(1) = 3$

$g(b) = \bar{d}$ $\qquad$ $g(2) = 5$

$g(c) = \bar{c}\,\bar{b}$ $\qquad$ $g(3) = 4$

$g(d) = bc4de$ $\qquad$ $g(4) = 2$

$g(e) = \bar{e}\,\bar{d}\,\bar{4}\,\bar{c}\,\bar{b}\,\bar{a}$ $\qquad$ $g(5) = 6$

$g(6) = 1$

Note $g(\bar{a}) = \bar{d}\,\bar{c}$, etc. $\qquad$ No cancellations

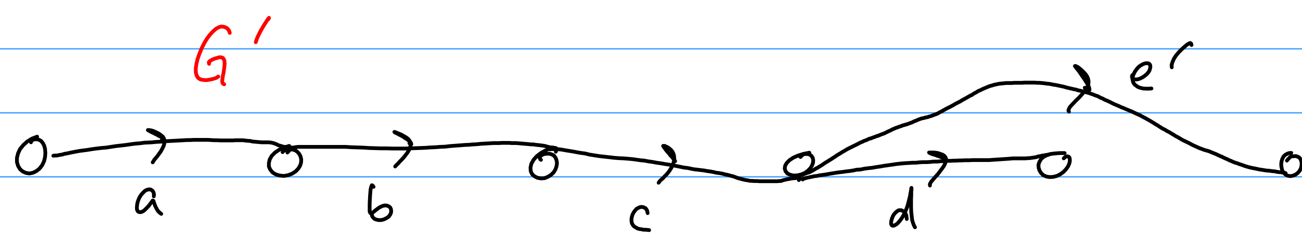
Now consider

$$g^2(d) = g(bc4de)$$
$$= (\bar{d})(\bar{c}\bar{b})(2)(bc4de)(\bar{e}\bar{d}\bar{4}\bar{c}\bar{b}\bar{a})$$
$$= \bar{d}\bar{c}\bar{b}2\bar{a}$$

cancellation!

Source of cancellation: $g(d)$ contains de, and $g(\bar{d})$ and $g(e)$ start in the same way.
$(\bar{e}\bar{d}\bar{4}\bar{c}\bar{b})$

To eliminate, identify $\bar{d}$ with the initial segment of e which has image $\bar{e}\bar{d}\bar{4}\bar{c}\bar{b}$:



$$e = \bar{d}e' \iff e' = de$$

This is called <u>folding</u> two edges

Let's see what this achieved:

$$g' : G' \to G'$$

$g'(a) = cd$

$g'(b) = \bar{d}$

$g'(c) = \bar{c}\bar{b}$

the map is simpler

$g'(d) = bc4de = bc4d(\bar{d}e') = bc4e'$

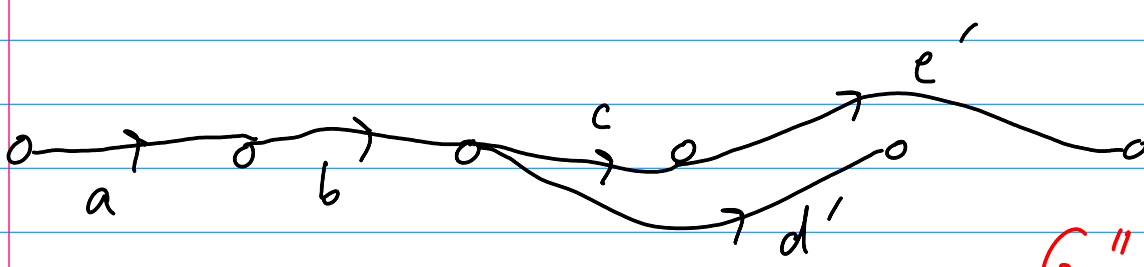
$g'(e') = g(de) = \bar{a}$

Now $(g')^2(a) = g'(c)g'(d) = \underbrace{(\bar{c}\bar{b})(bc4e')}_{\text{cancellation}}$

Still not efficient!

$g'(a)$ contains cd, and $g'(\bar{c})$ and $g'(d)$ both begin with bc.

Fold c and d: $d' = cd$



G''

$g'' : G'' \to G''$

$$g''(a) = cd = c\bar{c}d' = d'$$

$$g''(b) = \bar{d} = \bar{d}'c$$

$$g''(c) = \bar{c}\bar{b}$$

$$g''(d') = 4e'$$

$$g''(e') = \bar{a}$$

Note now that

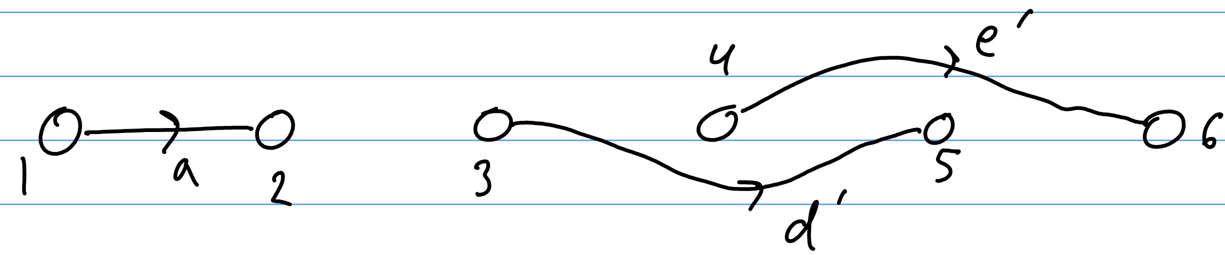
$$\begin{tikzcd}
 & a \cup 1 \cup 2 \arrow[dr, bend left, "g''"] & \\
e' \cup 4 \cup 6 \arrow[ur, bend left, "g''"] & & d' \cup 3 \cup 5 \arrow[ll, bend left, "g''"]
\end{tikzcd}$$

Check:

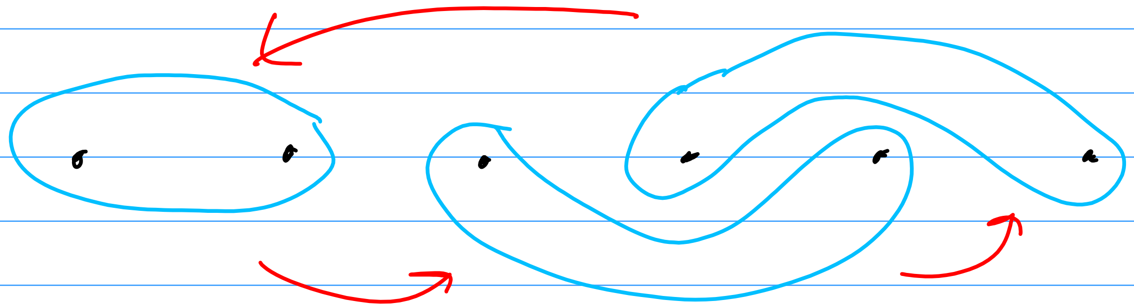
$$g''\{a, 1, 2\} = \{d', 3, 5\}$$

$$g''\{d', 3, 5\} = \{4e', 4, 6\}$$

$$g''\{e', 4, 6\} = \{\bar{a}, 2, 1\}$$



Reducing curves:



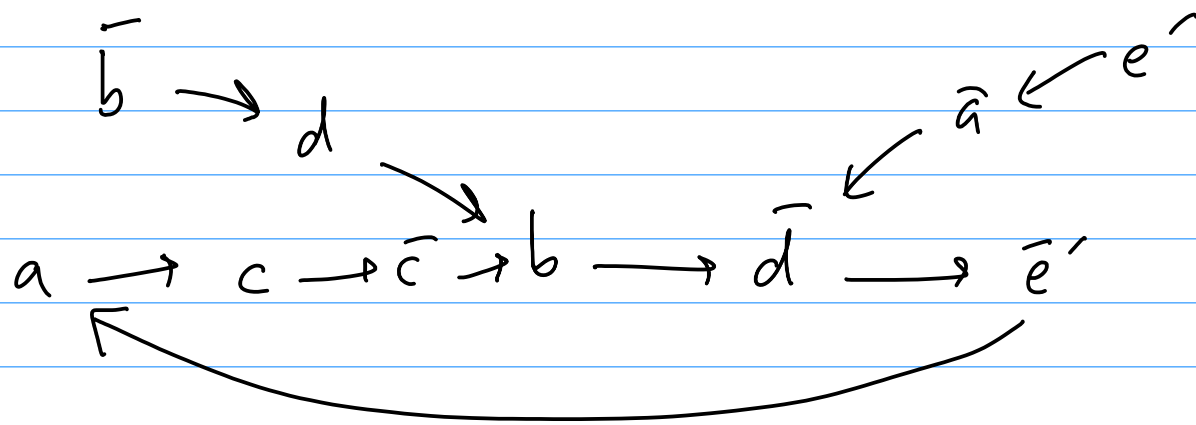
So this is a reducible braid.

So how do we check for cancellations more systematically?

Derivative map, Dg, sends each edge to the first edge traversed by its image.

For our earlier map g',

$$
\begin{array}{lll}
g'(a) = cd & Dg'(a) = c & Dg'(\bar{a}) = \bar{d} \\
g'(b) = \bar{d} & Dg'(b) = \bar{d} & Dg'(\bar{b}) = d \\
g'(c) = \bar{c}\bar{b} & Dg'(c) = \bar{c} & Dg'(\bar{c}) = b \\
g'(d) = bc4e' & Dg'(d) = b & Dg'(\bar{d}) = \bar{e}' \\
g'(e') = \bar{a} & Dg'(e') = \bar{a} & Dg'(\bar{e}') = a
\end{array}
$$



Two elements E_1 and E_2 are equivalent if

$$(Dg)^k(E_1) = (Dg)^k(E_2) \text{ for some } k \geq 0.$$

Equivalence classes are called gates.

Here the gates with >1 element are

$$\{d, \bar{c}, e'\}, \quad \{\bar{b}, c\}, \quad \{\bar{a}, b\}.$$

Lemma: no cancellations in $g^k(e)$
if $g(e) = E_1 \cdots E_r$ is such that
$\bar{E}_i$ and E_{i+1} $(i<r)$ are in different gates.
(easy)

Hence, we construct a list of bad words from the gates:

$\bar{d}\bar{c}$ bc ab
$\bar{d}e'$ $\bar{c}\bar{b}$ $\bar{b}\bar{a}$
cd
ce'
$\bar{e}'d$
$\bar{e}'\bar{c}$

and check that none of these occurs in the map
(here cd, bc, and $\bar{c}\bar{b}$ do)

$$
\begin{aligned}
g''(a) &= d' & Dg''(a) &= d' & Dg''(\bar{a}) &= \bar{d}' \\
g''(b) &= \bar{d}'c & Dg''(b) &= \bar{d}' & Dg''(\bar{b}) &= \bar{c} \\
g''(c) &= \bar{c}\bar{b} & Dg''(c) &= \bar{c} & Dg''(\bar{c}) &= b \\
g''(d') &= 4e' & Dg''(d') &= 4 & Dg''(\bar{d}') &= \bar{e}' \\
g''(e') &= \bar{a} & Dg''(e') &= \bar{a} & Dg''(\bar{e}') &= a
\end{aligned}
$$

$$
\begin{tikzcd}
 & & e' \arrow[r] & \bar{a} \arrow[d] & & & & \\
c \arrow[r] & \bar{c} \arrow[r] & b \arrow[r] & \bar{d}' \arrow[r] & \bar{e}' \arrow[r] & a \arrow[r] & d' \arrow[r] & 4
\end{tikzcd}
$$

Nontrivial gates are $\{\bar{c}, e'\}$, $\{\bar{a}, b\}$

bad words: ce', $\bar{e}'\bar{c}$, ab, $\bar{b}\bar{a}$

None of these appear above, so the map g'' is efficient.