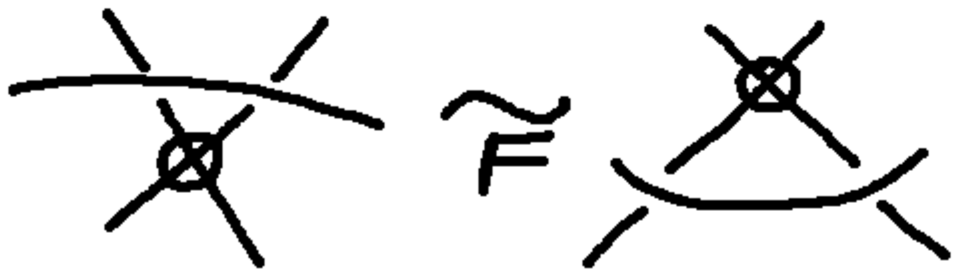
Problem Set

1.(a) Recall that the move



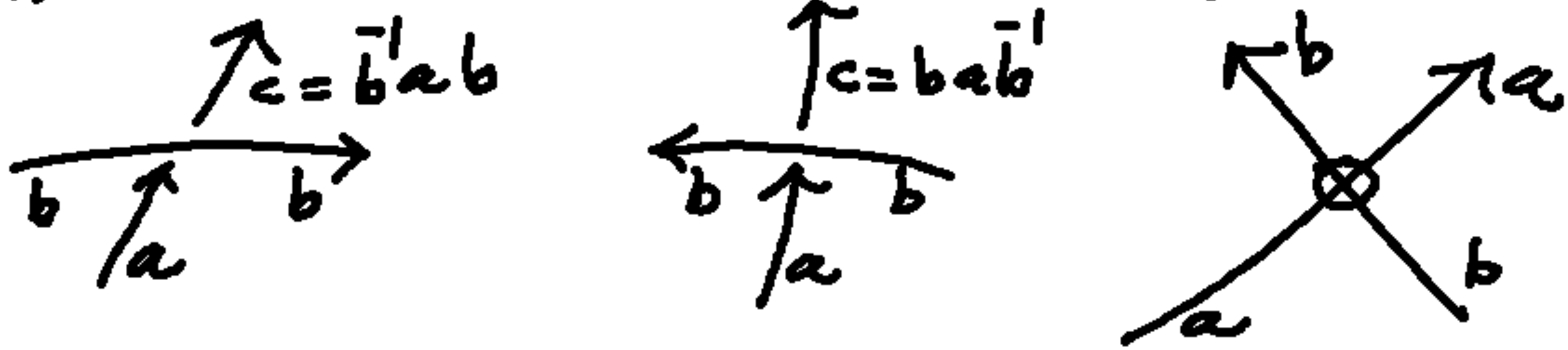
is forbidden in VKT (virtual knot theory).

Prove directly that

K is equivalent to the unknot U

in VKT + (F) = WKT (welded knot theory).

1. (b) We have defined WKT = VKT + (F), that is, one allows the overcrossing forbidden move F along with the other virtual moves (Reid moves + detour moves). In WKT define the <u>fundament group</u> $\pi(K)$ for an oriented diagram K via generators and relations (one generator for each arc in the diagram. one relation for each classical crossing).



1.(b) (continued)

(i) Show that $\pi\Big($  $\Big)$ is $\cong \mathbb{Z}$.

(ii) Show that

$\pi\Big($  $\Big)$ is <u>not</u> $\cong \mathbb{Z}$.

(iii) Show that $\pi(K)$ is invariant (up to isomorphism) under (a) Reidemeister moves.

(b) detour moves.

(c) the forbidden move F.

<u>Open Problem</u>. For K with one component does $\pi(K) \cong \mathbb{Z} \Rightarrow K \underset{WKT}{\sim} O$?

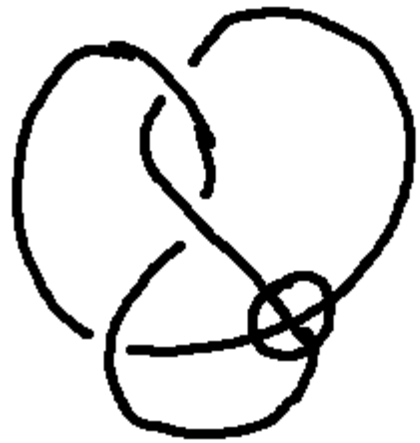
2. We know that $\hat{E}$ below

$\hat{E}$ has unit Jones polynomial.

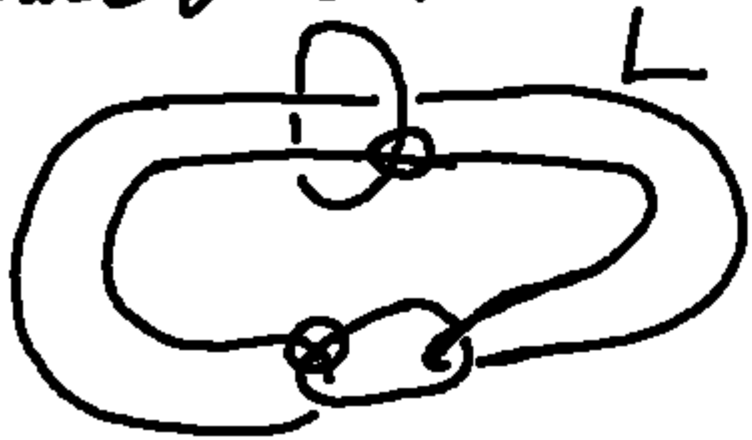
Calculate the affine index polynomial of $\hat{E}$. Use your result to show that $\hat{E}$ is a non-trivial, non-classical virtual knot.

What can you say about the mirror image $\hat{E}^*$ of $\hat{E}$?

3. (a) Compute the mod 2 Khovanov Homology for



(b) Find generators for the Lee Homology of L.



Use the Montesinos version with cut points + colorings.

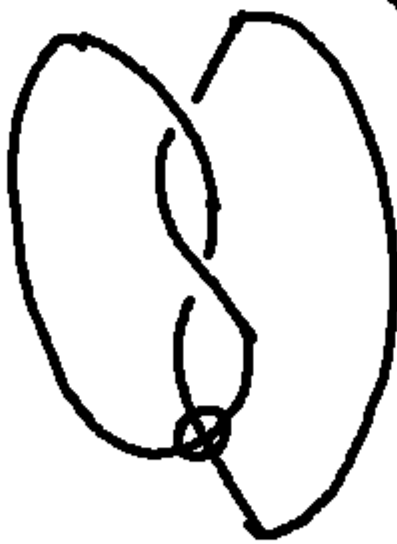
4. (a) Let $\mathcal{A}$ be the algebra defined by $\mathcal{A} = \mathbb{Z}[x]/(x^2)$,

$$\left\{ \begin{array}{l} \Delta(x) = x \otimes x \\ \Delta(1) = 1 \otimes x + x \otimes 1 \\ \varepsilon(x) = 1 \\ \varepsilon(1) = 0 \end{array} \right\}$$

Write an exposition about this algebra, explaining its relationships with surface cobordisms (cobordisms whose boundaries are 1-manifolds) and how it is used to define the Khovanov Chain Complex.

4.(b) Determine the mod 2 and the integral Khovanov homology of K shown below.

(Manturov type)



K

(c) Find the Doubled Khovanov homology of K above.

5. Use Marker States to compute ∇_E for  E.