Knot Scape and knot tables

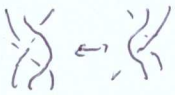
Review (+)

knot smooth $S^1 \hookrightarrow S^3$

$\{\text{knots}\} = \dfrac{\text{diagrams}}{\text{Reidemeister moves}}$

I)—→)

II)(→)(

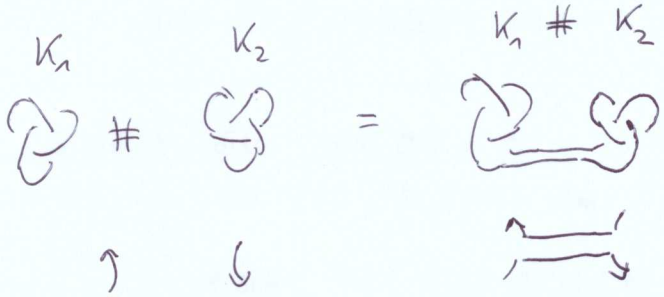
III 

local moves

orientation

mirror image

connected sum



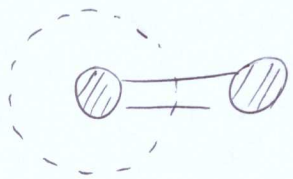
$K_1 \# K_2 = K_2 \# K_1$

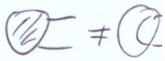
$K \# O = K$

Def: K prime if $\nexists K_1, K_2 \neq O$ $K = K_1 \# K_2$

every knot is a csum in a unique way up to permuting factors

Def: diagram is prime if not otherwise composite



and 

reducible crossing

here P & Q can be trivial arc 

diagram is reduced = no reducible crossing

prime ⟹ reduced (converse is false)

crossing number

$c(K) = \min \{ c(D) : D \text{ diagram of } K \}$

if $c(D) = c(K)$ then D is <u>minimal</u> diagram

reducible diagrams are never minimal

<u>conjectured</u> $c(K_1 \# K_2) = c(K_1) + c(K_2)$

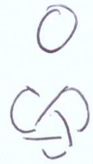
<u>knot table</u>: list all prime knots of given crossing number up to orientation and mirror image by exactly one minimal diagram each

\# of crsgs $\rightarrow 5_2 \leftarrow$ order in the table

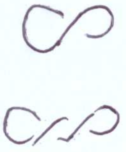
history Tait ~1880 Little, Kirkman, .. Conway, .. Caudron
Hoste - Thistlethwaite, ...
~199...

Flint Rankin - Sherman

<u>alternating</u> diagram



when you walk along knot pass crsgs over-under-over-under



alternating
reducible diagrams of O

other diagrams are non-alternating

<u>Def</u> K altern. :⟺ K has al. diagram otherwise non-altern.

for alternating links many problems are resolved:

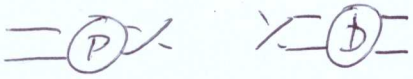
Th (Kauffman-Lin-Th) D reduced alternating diagram ⇒ D minimal

Th (Menasco) K alternating K prime ⇔ reduced alternating diag D is prime

Th (Menasco-Th.)

D_1, D_2 reduced alternating.

D_1, D_2 represent the same knot ⇔ $D_1 - D_2$ by a seq of flypes

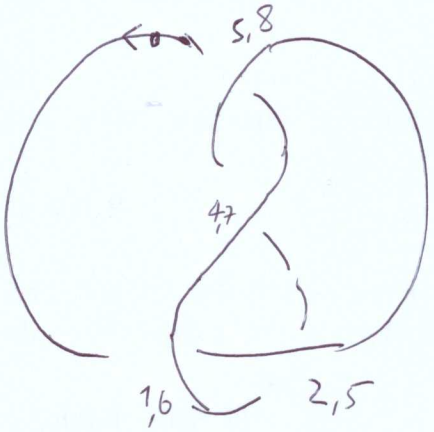


Knot Scape

- developed ~199.. by Rosh-Thistlethwaite
- version 1999 available on Th's homepage
- provides access to tables up to 16 crossings and allows to calculate some invariants
- requires tcl/tk under Linux
- tcl/tk main script involves C/C++ binaries (which can be exchanged)
- multiple (undocumented) bugs with binaries
- I have an own partly fixed and heavily altered version

"Help" help section with many explanations

DT notation used for diagrams



2u 4d 6u 8d
5d 7u 1d 3u

u = upper sheet
d = lower sheet

if even # u +
d −

one DT notation of diagram

1	3	5	7
6	−8	2	−4

depends on starting point and direction

DT notation determines prime diagram uniquely up to mirroring and orientation

⇓
reduced

mirroring can also change signs 6 −8 2 −4 → −6 8 −2 4

convention to fix mirror sign of first # = sign of crossing in Knotscape

↗↘ + ↖↙ −

(at always consistent)

↳ Lichtenstein / hand input?

in above example −↑→ ⇝ −6 8 −2 4
1,6
−

Note D alternating ⟺ all signs are + (or all are −)

≤49 limitation

DT diagram entry <cross> <knot id> <DT code>
#2 1 4 6

historically alternating knots (and diagrams)
in a table are listed before non-alternating ones

"Browse" functions allows to load or append
single knot or range of knots in knot table
to text window
my hacks: getting of non-alternating knots, getting of ≤10 crossing knots

"File" menu

Open / Close / Save / Save as operate on text in
text window
(usually a list of diagrams)

"Knot Sculptor"

facility for drawing knot / link diagram
with mouse

knot diagrams can be sent to text field

"Edit Braid"

input diagram of knot as closed braid $\sigma_1\ \sigma_2\ \sigma_3\ \sigma_1^{-1}\ \sigma_2^{-1}\ \sigma_3^{-1}$
$a\ b\ c\ A\ B\ C$

$1 \ldots \overset{\sigma_i}{\gtrless} 1 \ldots 1$

(warning! bugs: can run into ∞ loops ...)

"Actions"

locate in table

goal: if diagram is input, find to which
knot in table it belongs
(or which crossing thereof)

works (so far wlt bugs) if initial diagram has ≤ 16 crossings

(original version uses "a"/"n" for (non-) alternating in output window, but I've hacked this away)

if input diagram has ≥ 17 crossings and cannot identify knot in table, outputs a "best reduction available" diagram

(warning: has bugs! may alter knot)

how it works:

knotfind.c

apply moves composed of Reid moves but more complex and not raising crossing number

Project part 1: put out which moves

writhe := #↗ − #↖

- the flype
- the Perko move (changes #↗, ↖ wlt changes #crossings)
- the (k,ℓ)-pass move (for some particular (k,ℓ))

assume $k \geq \ell$

$k+\ell$ odd

but even

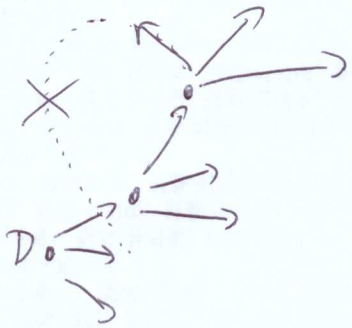
Hype and (4,6)-pass do at all unlike
but Reho's move does

- double pass ... etc. (see HTW article)

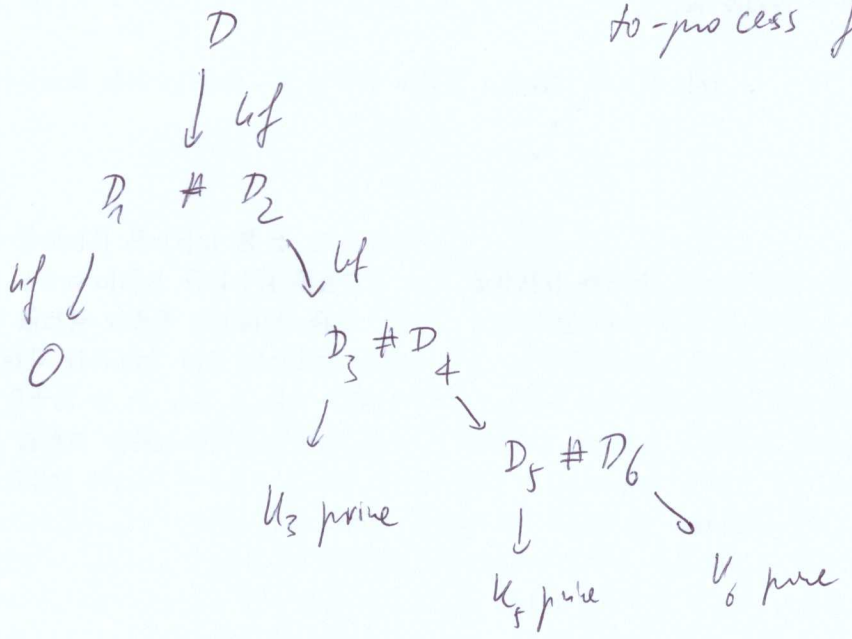
- apply all moves you can get
but make sure to avoid
cycles

maintain a dbase of
reached diagrams

code diagram by canon # and
for given canon # by DT notations
(canonical ht for each diagram)

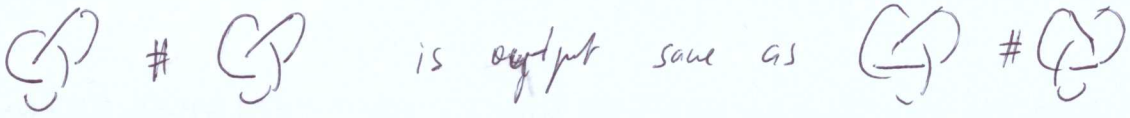


if you reach composite diagram
output factors and call hat put recursively
(maintain a stack of
to-process factor diagrams)



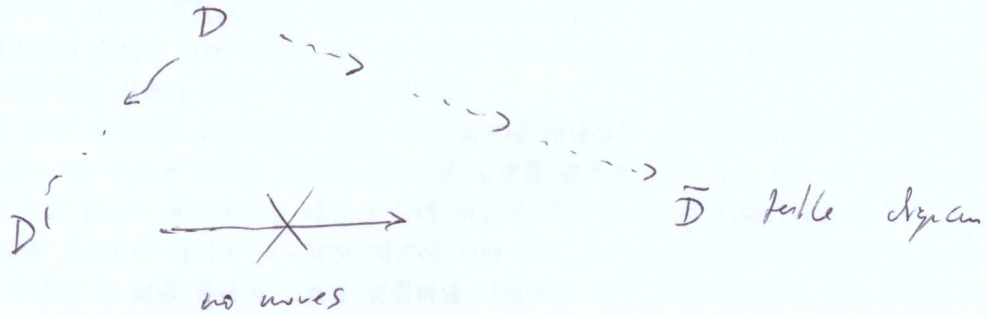
output: ~~0~~ # U_3 # U_5 # U_6

factors are output up to mirroring



but they are not the same even up to mirroring

problem: moves in knotfind.c 's repertoire do not always suffice to transform D into a diagram D' from the table even if $c(D') \leq 16$



sometimes $c(\bar{D}) < c(D')$

to solve this there is an extra list of duplications (for ≤16 crossing diagrams)

swap 4315 D'_1, D'_2, ... $\tilde{D}$ } change all this to this

bug: (for >16 crs at least)
some move changes knot type

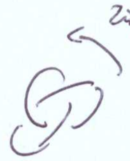
$D \rightsquigarrow D'$ D, D' should belong to the same knot but sometimes they don't!

Project: fix bugs in knotfind.c
or better write a correct new one

Then: I have tools for the flype and general (k,l)-pass

apparently (k,l)-pass for general k,l is not used in knotfind.c

"Draw Knot" uses some circle packing algorithm to draw knot from diagram



"symmetry group" group of motions of complement that leave knot invariant up to orientations

effect on universe and ori

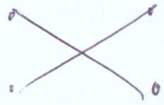
K   −K   some may coincide

!K   −!K = !−K

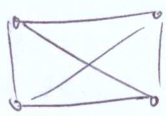
"symmetry type"


chiral 9_{32}


invertible / reversible 


−amphicheiral 8_{17}

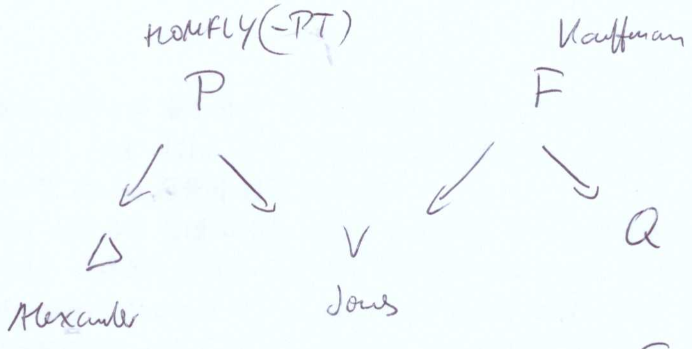

+amphicheiral


fully amphicheiral  figure-8-knot 4_1

(works if the knot is "hyperbolic")

hyperbolic routines (so far) bug-free

"Compute Polynomials"



substitution

$l^{-1}P(\ldots) + mP(\ldots) + lP(\ldots) = 0$ P

$i=\sqrt{-1}$ $l = it$ $m = i(t^{1/2}-t^{-1/2})$ } V

Millett (HOMFLY's M) - Ewing ~ 1987-90

write very fast program for P, F (using skein method)

KnotScape (in downloadable version)

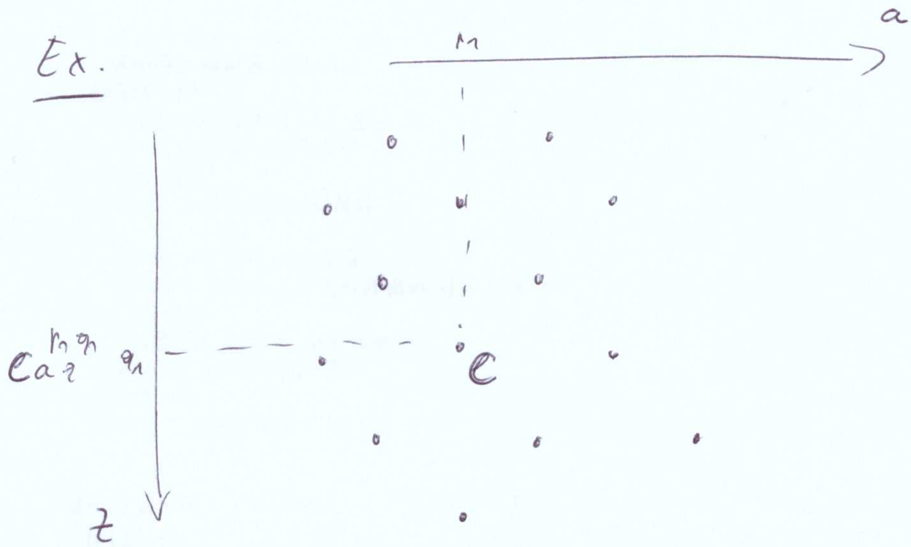
uses Millett-Ewing + substitution routines for Δ, V

Millett-Ewing has enormous bugs

· crash, ∞ loop, visibly wrong output

hiddenly —"—

Ex.

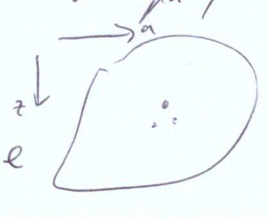


Kauffman poly $F(a,z)$

p_1+q_1 always even

visibly — sometimes output monomials with p_1+q_1 odd

hiddenly — sometimes 1 coeff wrong in middle



1,2 coeffs wrong by ±1

I was able to get from SnapPy
a version of SnapPy prog which I made (so far) bug free
but Kauffman is hopeless
on Ubuntu 18/20 crashes already at 12 crossings

↑ but these are in KnotScape this is bugged

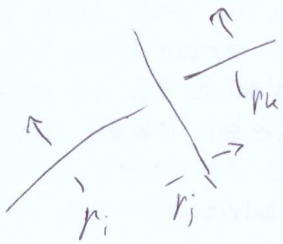
have now my own programs (SnapPy & Kauffman)
which are ~2× slower, but bug free
(so far)

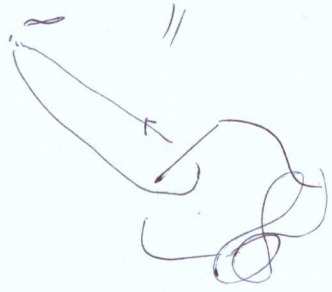
small project: integrate them into KnotScape
(properly)

"find homomorphisms"

$$\pi(S^3 \setminus K) \longrightarrow S_5$$

20 gens





Wirtinger presentation

$$\langle p_i \mid p_j p_k p_j^{-1} = p_i \rangle = \pi_1(S^3 \setminus K)$$

p_i meridians are all conjugate

so $s(p_i)$ will have same cycle type = (conj class in S_5)

count homomorphisms up to conjugacy

try to find few p_i greedily & test relations

(seems to work well, haven't tried much)