Pensieve header: Khovanov Homology, Day 5.

Topics (in no particular order). Whatever you may suggest; whatever comes to my mind; ~~the Fibonacci numbers~~; ~~the Catalan numbers~~; ~~the Jones polynomial~~; ~~a more efficient Jones algorithm~~; ~~a riddle on spheres~~; Khovanov homology; Γ-calculus; the Hopf fibration; Hilbert's 13th problem; non-commutative Gaussian elimination; free Lie algebras; the Baker-Campbell-Hausdorff formula; wacky numbers; ~~an order 4 torus~~; the Schwarz Lantern; knot colourings; the Temperley-Lieb pairing; the dodecahedral link; ~~sound experiments~~; barycentric subdivisions; ~~some Peano curves~~; braid closures and Vogel's algorithm; ~~the insolubility of the quintic~~; phase portraits; the Mandelbrot set; shadows of the Cantor square; quilt plots; some image transformations; De Bruijn graphs; the Riemann series theorem; finite type invariants and the Willerton fish; ~~the Towers of Hanoi~~; ~~Hochschild homology of (some) coalgebras~~; ~~convolutions and image improvements~~; ~~the 8-5-3 milk jug problem~~; ~~a cow problem,~~ a permutations package.

One further class meeting on Thursday at 10 at Bahen 6180!

Outstanding Challenges

The last day to submit projects for marks will be the last day of the UofT examination period, December 20 2017 at midnight.

- Update the NCGE program to contain "backtracking information". Use it to find how to turn the lower face of a Rubik's cube by turning all but the lower face of that cube.
- Complete the package perm, with documentation and all. For Perm[5,2,3,1,4], etc, your package should know $\sigma\circ\tau$, σ^{-1}, $\sigma[\![i]\!]$, Pivot[σ], PermutationQ[σ], IdentityPermutation[n], it should interact well with Cycles, and its internals should be hidden. It should live in a file "Perm.m".
- Write a sophisticated Mathematica package that implements QuiltPlot and DoubleQuiltPlot with many bells and whistles, as explained on November 20th.
- Draw approximations of the Cantor square C^2. Then rotate C^2 by an angle θ and project the rotation vertically, draw that projection (while explaining why it is what you think it is), and compute its measure. Perhaps also use Manipulate[..., $\{\theta, 0, \pi/2\}$]?
- Everybody knows that the length of a smooth curve is the supremum of the lengths of its polygonal approximations. Everybody should know that the same is not true for smooth surfaces. Can you draw a smooth surface in $\mathbb{R}^3$ along with a triangulated approximation thereof whose area is vastly more? An appropriate animation may be even more convincing.
- Make the best picture of the Hopf fibration the world has even seen.

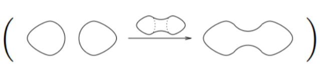
Khovanov Homology

Challenge: Implement Khovanov Homology as it is defined in the handout. Your implementation should fit in one page and should be good enough to show that $\mathrm{Kh}(5_1) \neq \mathrm{Kh}(10_{132})$.

Khovanov: $K(L)$ is a chain complex of graded $\mathbb{Z}$-modules;
$V = \mathrm{span}\langle v_+, v_-\rangle$; $\deg v_\pm = \pm 1$; $q\dim V = q + q^{-1}$;

$$K(\bigcirc^k) = V^{\otimes k};\quad K(\times) = \mathrm{Flatten}\left(0 \to \underset{\text{height } 0}{K(\rangle\langle)\{1\}} \to \underset{\text{height } 1}{K(\asymp)\{2\}} \to 0\right);$$

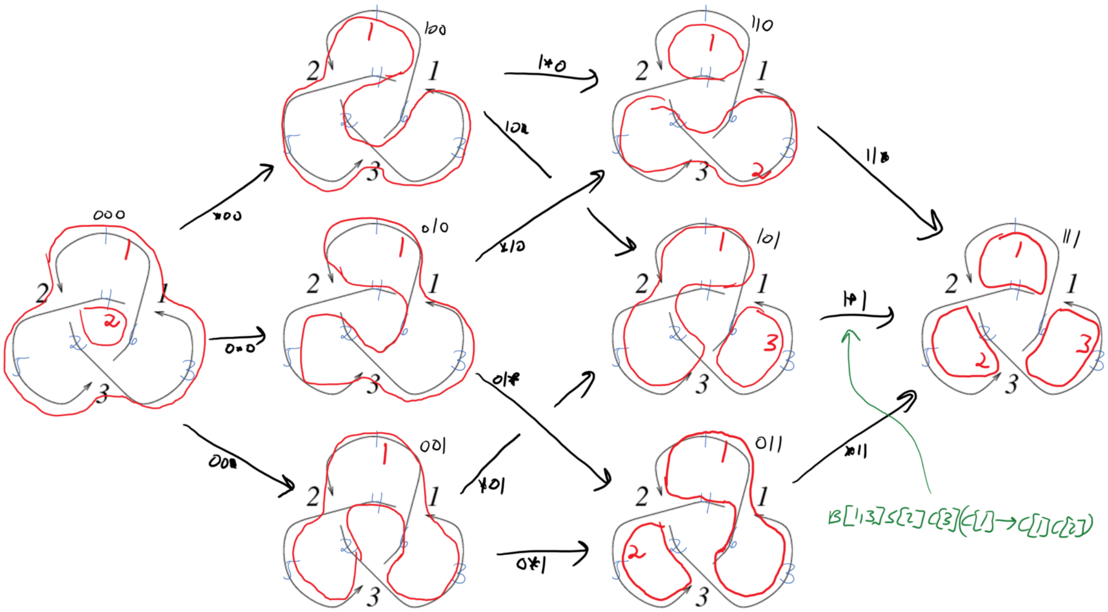
$$K(\times) = \mathrm{Flatten}\left(0 \to \underset{\text{height } -1}{K(\asymp)\{-2\}} \to \underset{\text{height } 0}{K(\rangle\langle)\{-1\}} \to 0\right);$$



$\longrightarrow (V\otimes V \xrightarrow{m} V)$ $\quad m: \begin{cases} v_+\otimes v_- \mapsto v_- & v_+\otimes v_+ \mapsto v_+ \\ v_-\otimes v_+ \mapsto v_- & v_-\otimes v_- \mapsto 0 \end{cases}$

$\longrightarrow (V \xrightarrow{\Delta} V\otimes V)$ $\quad \Delta: \begin{cases} v_+ \mapsto v_+\otimes v_- + v_-\otimes v_+ \\ v_- \mapsto v_-\otimes v_- \end{cases}$

```
K = K31 = PD[X[3, 1, 4, 6], X[1, 5, 2, 4], X[5, 3, 6, 2]];
K51 = PD[X[1, 6, 2, 7], X[3, 8, 4, 9], X[5, 10, 6, 1], X[7, 2, 8, 3], X[9, 4, 10, 5]];
K10132 = PD[X[4, 2, 5, 1], X[8, 4, 9, 3], X[5, 12, 6, 13], X[15, 18, 16, 19], X[9, 16, 10, 17],
   X[17, 10, 18, 11], X[13, 20, 14, 1], X[19, 14, 20, 15], X[11, 6, 12, 7], X[2, 8, 3, 7]];
```




```
SetAttributes[{B, P}, Orderless]; t = 0;
VerticesAndCycles = Expand[
    Times @@ (K /. X[i_, j_, k_, l_] :> (++t; P[i, j] P[k, l] + B[t] P[i, l] P[j, k]))
   ] //. B[i__] B[j__] :> B[i, j] /. P[a_, b_] :> P[a, b][Min[a, b]] //.
  P[a_, b_][m1_] P[b_, c_][m2_] :> P[a, c][Min[m1, m2]] /. {P[i_, i_][m_] :> c[m], P[_, _][m_]^2 :> c[m]}
```

```
B[1] c[1] + B[2] c[1] + B[3] c[1] + c[1] c[2] + B[1, 2] c[1] c[2] +
 B[2, 3] c[1] c[2] + B[1, 3] c[1] c[3] + B[1, 2, 3] c[1] c[2] c[3]
```

```
K
```

```
PD[X[3, 1, 4, 6], X[1, 5, 2, 4], X[5, 3, 6, 2]]
```

```
Times @@ K
```

```
X[1, 5, 2, 4] X[3, 1, 4, 6] X[5, 3, 6, 2]
```

```
FullBasis = List @@ Expand[VerticesAndCycles /. c[m_] :> vp[m] + vm[m]]
```

```
{B[1] vm[1], B[2] vm[1], B[3] vm[1], vm[1] vm[2], B[1, 2] vm[1] vm[2], B[2, 3] vm[1] vm[2],
 B[1, 3] vm[1] vm[3], B[1, 2, 3] vm[1] vm[2] vm[3], B[1] vp[1], B[2] vp[1], B[3] vp[1], vm[2] vp[1],
 B[1, 2] vm[2] vp[1], B[2, 3] vm[2] vp[1], B[1, 3] vm[3] vp[1], B[1, 2, 3] vm[2] vm[3] vp[1], vm[1] vp[2],
 B[1, 2] vm[1] vp[2], B[2, 3] vm[1] vp[2], B[1, 2, 3] vm[1] vm[3] vp[2], vp[1] vp[2], B[1, 2] vp[1] vp[2],
 B[2, 3] vp[1] vp[2], B[1, 2, 3] vm[3] vp[1] vp[2], B[1, 3] vm[1] vp[3], B[1, 2, 3] vm[1] vm[2] vp[3],
 B[1, 3] vp[1] vp[3], B[1, 2, 3] vm[2] vp[1] vp[3], B[1, 2, 3] vm[1] vp[2] vp[3], B[1, 2, 3] vp[1] vp[2] vp[3]}
```

Next Task. Produce a “generating function for the edges”.

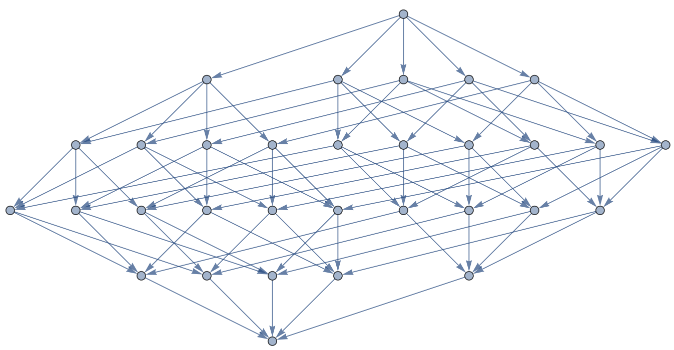
```
? Coefficient
```

> Coefficient[*expr*, *form*] gives the coefficient of *form* in the polynomial *expr*.
> Coefficient[*expr*, *form*, *n*] gives the coefficient of *form*^*n* in *expr*. >>

```
n = 2;
G = Expand[Product[a_i + b_i + ϵ e_i, {i, n}]];
{VV = List @@ Coefficient[G, ϵ, 0], EE = List @@ Coefficient[G, ϵ, 1]}
```

$\{\{a_1 a_2, a_2 b_1, a_1 b_2, b_1 b_2\}, \{a_2 e_1, b_2 e_1, a_1 e_2, b_1 e_2\}\}$

```
n = 5;
G = Expand[Product[a_i + b_i + ϵ e_i, {i, n}]];
{VV = List @@ Coefficient[G, ϵ, 0], EE = List @@ Coefficient[G, ϵ, 1]};
Graph[VV, Table[(ξ /. e → a) → (ξ /. e → b), {ξ, EE}]]
```



```
SetAttributes[{B, P}, Orderless]; t = 0;
ϵ /: ϵ^p_ /; p > 1 := 0;
VerticesAndCycles = Expand[
     Times @@ (K /. X[i_, j_, k_, l_] :→ (++t; P[i, j] P[k, l] + B[t] P[i, l] P[j, k]))
    ] //. B[i__] B[j__] :→ B[i, j] /. P[a_, b_] :→ P[a, b][Min[a, b]] //.
  P[a_, b_][m1_] P[b_, c_][m2_] :→ P[a, c][Min[m1, m2]] /. {P[i_, i_][m_] :→ c[m], P[_, _][m_]^2 :→ c[m]}
```

```
B[1] c[1] + B[2] c[1] + B[3] c[1] + c[1] c[2] + B[1, 2] c[1] c[2] +
 B[2, 3] c[1] c[2] + B[1, 3] c[1] c[3] + B[1, 2, 3] c[1] c[2] c[3]
```

```
SetAttributes[{B, P}, Orderless]; t = 0; ϵ /: ϵ^p_ /; p > 1 := 0;
Expand[
    Times @@ (K /. X[i_, j_, k_, l_] :→ (++t; P[i, j] P[k, l] + B[t] P[i, l] P[j, k] + ϵ S[t] X[i, j, k, l]))
   ] //. B[i__] B[j__] :→ B[i, j] /. P[a_, b_] :→ P[a, b][Min[a, b]] //.
 P[a_, b_][m1_] P[b_, c_][m2_] :→ P[a, c][Min[m1, m2]] /. {P[i_, i_][m_] :→ c[m], P[_, _][m_]^2 :→ c[m]}
```

```
B[1] c[1] + B[2] c[1] + B[3] c[1] + c[1] c[2] + B[1, 2] c[1] c[2] +
 B[2, 3] c[1] c[2] + B[1, 3] c[1] c[3] + B[1, 2, 3] c[1] c[2] c[3] +
 ϵ S[2] X[1, 5, 2, 4] P[1, 5][1] P[2, 4][2] + ϵ B[1] S[2] X[1, 5, 2, 4] P[1, 4][1] P[2, 5][2] +
 ϵ B[3] S[2] X[1, 5, 2, 4] P[1, 4][1] P[2, 5][2] + ϵ B[1, 3] c[3] S[2] X[1, 5, 2, 4] P[1, 4][1] P[2, 5][2] +
 ϵ S[3] X[5, 3, 6, 2] P[2, 6][2] P[3, 5][1] + ϵ B[2] S[3] X[5, 3, 6, 2] P[2, 5][2] P[3, 6][1] +
 ϵ B[2] S[1] X[3, 1, 4, 6] P[1, 4][1] P[3, 6][2] + ϵ B[3] S[1] X[3, 1, 4, 6] P[1, 4][1] P[3, 6][3] +
 ϵ B[2, 3] c[2] S[1] X[3, 1, 4, 6] P[1, 4][1] P[3, 6][3] + ϵ B[1] S[3] X[5, 3, 6, 2] P[2, 5][1] P[3, 6][3] +
 ϵ B[1, 2] c[1] S[3] X[5, 3, 6, 2] P[2, 5][2] P[3, 6][3] + ϵ S[1] X[3, 1, 4, 6] P[1, 3][1] P[4, 6][2]
```

```
ϵ B[1, 3] c[3] S[2] X[1, 5, 2, 4] P[1, 4][1] P[2, 5][2] /. X[i_, j_, k_, l_] P[i, j][m1_] P[k, l][m2_] :→ (...)
```