Pensieve header: Khovanov Homology, Day 4.

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.

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[..., {θ, 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.

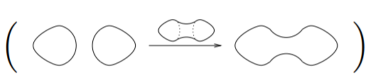
Khovanov Homology

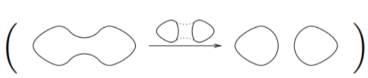
Challenge: Implement Khovanov Homology as it is defined in the handout. Your implementation should fit in one page and should be good enough to show that $\text{Kh}(5_1) \neq \text{Kh}(10_{132})$.

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

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

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


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


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

```
SystemOpen[
 "C:\\drorbn\\AcademicPensieve\\Classes\\17-1750-ShamelessMathematica\\171127-111330.png"]
```

```
K = PD[X[1, 4, 2, 3], X[2, 4, 1, 3]]
```

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

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

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

```
SetAttributes[P, Orderless];
Expand[Times @@ K /. X[i_, j_, k_, l_] :> A P[i, j] P[k, l] + B P[i, l] P[j, k]] //.
  P[a_, b_] P[b_, c_] :> P[a, c] /. {P[i_, i_] → d, P[_, _]^2 → d}
```

$3 A^2 B d + A^3 d^2 + 3 A B^2 d^2 + B^3 d^3$

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

```
A^3 c[3] c[4] + A^2 B c[3] c[4] + A B^2 c[3] c[4] + A^2 B c[2] c[5] +
 A B^2 c[2] c[6] + A^2 B c[3] c[5] c[6] + A B^2 c[3] c[5] c[6] + B^3 c[4] c[5] c[6]
```

```
Expand[Times @@ K /. X[i_, j_, k_, l_] :> A P[i, j] P[k, l] + B P[i, l] P[j, k]] /.
 P[a_, b_] :> P[a, Min[a, b], b]
```

```
A^2 B P[1, 1, 4] P[1, 1, 5] P[2, 2, 4] P[2, 2, 6] P[3, 3, 5] P[3, 3, 6] +
 A B^2 P[1, 1, 4]^2 P[2, 2, 5] P[2, 2, 6] P[3, 3, 5] P[3, 3, 6] +
 A B^2 P[1, 1, 4] P[1, 1, 5] P[2, 2, 4] P[2, 2, 5] P[3, 3, 6]^2 + B^3 P[1, 1, 4]^2 P[2, 2, 5]^2 P[3, 3, 6]^2 +
 A^3 P[1, 1, 3] P[1, 1, 5] P[2, 2, 4] P[2, 2, 6] P[3, 3, 5] P[4, 4, 6] +
 A^2 B P[1, 1, 3] P[1, 1, 4] P[2, 2, 5] P[2, 2, 6] P[3, 3, 5] P[4, 4, 6] +
 A^2 B P[1, 1, 3] P[1, 1, 5] P[2, 2, 4] P[2, 2, 5] P[3, 3, 6] P[4, 4, 6] +
 A B^2 P[1, 1, 3] P[1, 1, 4] P[2, 2, 5]^2 P[3, 3, 6] P[4, 4, 6]
```

```
Expand[Times @@ K /. X[i_, j_, k_, l_] :> A P[i, j] P[k, l] + B P[i, l] P[j, k]] /.
  P[a_, b_] :> P[a, Min[a, b], b] //. P[a_, m1_, b_] P[b_, m2_, c_] :> P[a, Min[m1, m2], c]
```

```
A B^2 P[1, 1, 4]^2 P[2, 2, 3] + A^3 P[1, 1, 3] P[2, 2, 4] + A^2 B P[1, 1, 3] P[2, 2, 4] +
 A^2 B P[1, 1, 2] P[3, 3, 5] + A B^2 P[1, 1, 2] P[3, 3, 6]^2 + B^3 P[1, 1, 4]^2 P[2, 2, 5]^2 P[3, 3, 6]^2 +
 A^2 B P[1, 1, 3] P[2, 2, 5] P[4, 4, 6] + A B^2 P[1, 1, 3] P[2, 2, 5]^2 P[4, 4, 6]
```

```
SetAttributes[P, Orderless];
SetAttributes[B, 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]
```

```
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]}
```

```
FullBasis // Length
```

```
30
```