Pensieve header: The Kerler Algebra and the Alexander polynomial. Closely related to arXiv://math/0008204.

In[∘]:=
```
SetDirectory["C:\\drorbn\\AcademicPensieve\\Classes\\21-1350-KnotTheory"];
<< Common.m
```

In[∘]:=
```
MT = (
        a    b    c    d    ka   kb   kc   kd
   a    a    b    0    0    ka   kb   0    0
   b    0    0    a    b    0    0    -ka  -kb
   c    c    d    0    0    -kc  -kd  0    0
   d    0    0    c    d    0    0    kc   kd
   ka   ka   kb   0    0    a    b    0    0
   kb   0    0    ka   kb   0    0    -a   -b
   kc   kc   kd   0    0    -c   -d   0    0
   kd   0    0    kc   kd   0    0    c    d
);

ℰ_ // m_{i_,j_→k_} := Expand[ℰ] /. Flatten@Table[MT⟦α, 1⟧_i MT⟦1, β⟧_j →
      (MT⟦α, β⟧ /. v : (a | b | c | d | ka | kb | kc | kd) ⧴ v_k), {α, 2, 9}, {β, 2, 9}];
η_{i_} :=
  a_i +
   d_i;
```

In[∘]:=
```
Flatten@Table[MT⟦α, 1⟧_i MT⟦1, β⟧_j →
   (MT⟦α, β⟧ /. v : (a | b | c | d | ka | kb | kc | kd) ⧴ v_k), {α, 2, 9}, {β, 2, 9}]
```

Out[∘]=
```
{a_i a_j → a_k, a_i b_j → b_k, a_i c_j → 0, a_i d_j → 0, a_i ka_j → ka_k, a_i kb_j → kb_k, a_i kc_j → 0,
 a_i kd_j → 0, a_j b_i → 0, b_i b_j → 0, b_i c_j → a_k, b_i d_j → b_k, b_i ka_j → 0, b_i kb_j → 0, b_i kc_j → -ka_k,
 b_i kd_j → -kb_k, a_j c_i → c_k, b_j c_i → d_k, c_i c_j → 0, c_i d_j → 0, c_i ka_j → -kc_k, c_i kb_j → -kd_k,
 c_i kc_j → 0, c_i kd_j → 0, a_j d_i → 0, b_j d_i → 0, c_j d_i → c_k, d_i d_j → d_k, d_i ka_j → 0, d_i kb_j → 0,
 d_i kc_j → kc_k, d_i kd_j → kd_k, a_j ka_i → ka_k, b_j ka_i → kb_k, c_j ka_i → 0, d_j ka_i → 0, ka_i ka_j → a_k,
 ka_i kb_j → b_k, ka_i kc_j → 0, ka_i kd_j → 0, a_j kb_i → 0, b_j kb_i → 0, c_j kb_i → ka_k, d_j kb_i → kb_k,
 ka_j kb_i → 0, kb_i kb_j → 0, kb_i kc_j → -a_k, kb_i kd_j → -b_k, a_j kc_i → kc_k, b_j kc_i → kd_k,
 c_j kc_i → 0, d_j kc_i → 0, ka_j kc_i → -c_k, kb_j kc_i → -d_k, kc_i kc_j → 0, kc_i kd_j → 0, a_j kd_i → 0,
 b_j kd_i → 0, c_j kd_i → kc_k, d_j kd_i → kd_k, ka_j kd_i → 0, kb_j kd_i → 0, kc_j kd_i → c_k, kd_i kd_j → d_k}
```

In[∘]:=
```
KBasis[i_] := {a_i, b_i, c_i, d_i, ka_i, kb_i, kc_i, kd_i};
KBasis[i_, is__] := Flatten@Outer[Times, KBasis[i], KBasis[is]]
```

In[∘]:=
```
KBasis[1]
```

Out[∘]=
```
{a_1, b_1, c_1, d_1, ka_1, kb_1, kc_1, kd_1}
```

```
In[]:= KBasis[5, 6]
```

Out[]= $\{a_5 a_6, a_5 b_6, a_5 c_6, a_5 d_6, a_5 ka_6, a_5 kb_6, a_5 kc_6, a_5 kd_6, a_6 b_5, b_5 b_6, b_5 c_6, b_5 d_6, b_5 ka_6, b_5 kb_6, b_5 kc_6, b_5 kd_6, a_6 c_5, b_6 c_5, c_5 c_6, c_5 d_6, c_5 ka_6, c_5 kb_6, c_5 kc_6, c_5 kd_6, a_6 d_5, b_6 d_5, c_6 d_5, d_5 d_6, d_5 ka_6, d_5 kb_6, d_5 kc_6, d_5 kd_6, a_6 ka_5, b_6 ka_5, c_6 ka_5, d_6 ka_5, ka_5 ka_6, ka_5 kb_6, ka_5 kc_6, ka_5 kd_6, a_6 kb_5, b_6 kb_5, c_6 kb_5, d_6 kb_5, ka_6 kb_5, kb_5 kb_6, kb_5 kc_6, kb_5 kd_6, a_6 kc_5, b_6 kc_5, c_6 kc_5, d_6 kc_5, ka_6 kc_5, kb_6 kc_5, kc_5 kc_6, kc_5 kd_6, a_6 kd_5, b_6 kd_5, c_6 kd_5, d_6 kd_5, ka_6 kd_5, kb_6 kd_5, kc_6 kd_5, kd_5 kd_6\}$

Associativity of *m*:

```
In[]:= Short[lhs = KBasis[1, 2, 3] // m1,2→1 // m1,3→1]
rhs = KBasis[1, 2, 3] // m2,3→2 // m1,2→1;
lhs == rhs
```

Out[]//Short= $\{a_1, b_1, 0, 0, ka_1, \ll 502 \gg, d_1, 0, 0, kc_1, kd_1\}$

Out[]= True

Units:

```
In[]:= {(η1 KBasis[2] // m1,2→1) == KBasis[1], (η1 KBasis[2] // m2,1→1) == KBasis[1]}
```

Out[]= {True, True}

```
In[]:= R_{i_,j_} := T^{-1/2} (a_i ka_j - T a_i kd_j + (T - 1) b_j kc_i + d_i ka_j + T d_i kd_j);
       R̄_{i_,j_} := T^{1/2} (a_i ka_j - T^{-1} a_i kd_j + (T^{-1} - 1) b_j kc_i + d_i ka_j + T^{-1} d_i kd_j);
       C_{i_} := T^{-1/2} (ka_i + kd_i); C̄_{i_} := T^{1/2} (ka_i + kd_i);
```

Reidemeister 3:

```
In[]:= Short[lhs = R1,2 R4,3 R5,6 // m1,4→1 // m2,5→2 // m3,6→3]
rhs = R2,3 R1,4 R5,6 // m1,5→1 // m2,6→2 // m3,4→3;
lhs == rhs
```

Out[]//Short= $\frac{a_1 a_3 ka_2}{T^{3/2}} + \ll 27 \gg$

Out[]= True

Reidemeister 2b:

```
In[]:= Short[lhs = R1,2 R̄3,4 // m1,3→1 // m2,4→2]
rhs = η1 η2 // Expand;
lhs == rhs
```

Out[]//Short= $a_1 a_2 + a_2 d_1 + a_1 d_2 + d_1 d_2$

Out[]= True

Naive Reidemeister 2c:

In[∘]:= Short[lhs = $R_{1,2}\,\overline{R}_{3,4}$ // $m_{1,3\to 1}$ // $m_{4,2\to 2}$]
rhs = $\eta_1\,\eta_2$ // Expand;
Simplify[lhs - rhs]

Out[∘]//Short= $a_1\,a_2 + a_2\,d_1 + \ll 1\gg + \ll 1\gg - 2\,kb_2\,kc_1 + 2\,T\,kb_2\,kc_1$

Out[∘]= $2\times(-1+T)\,kb_2\,kc_1$

Corrected Reidemeister 2c:

In[∘]:= lhs = $R_{1,4}\,\overline{R}_{5,2}\,\overline{C}_3$ // $m_{2,4\to 2}$ // $m_{1,3\to 1}$ // $m_{1,5\to 1}$
rhs = $\overline{C}_1\,\eta_2$ // Expand;
lhs == rhs

Out[∘]= $\sqrt{T}\,a_2\,ka_1 + \sqrt{T}\,d_2\,ka_1 + \sqrt{T}\,a_2\,kd_1 + \sqrt{T}\,d_2\,kd_1$

Out[∘]= True

$C\,\overline{C}$:

In[∘]:= $C_1\,\overline{C}_2$ // $m_{1,2\to 1}$

Out[∘]= $a_1 + d_1$

Reidemeister 1s:

In[∘]:= {($\overline{C}_2\,R_{1,3}$ // $m_{1,2\to 1}$ // $m_{1,3\to 1}$) == η_1, ($\overline{C}_2\,\overline{R}_{3,1}$ // $m_{1,2\to 1}$ // $m_{1,3\to 1}$) == η_1,
($C_2\,\overline{R}_{1,3}$ // $m_{1,2\to 1}$ // $m_{1,3\to 1}$) == η_1, ($C_2\,R_{3,1}$ // $m_{1,2\to 1}$ // $m_{1,3\to 1}$) == η_1}

Out[∘]= {True, True, True, True}

The whirl:

In[∘]:= Expand[($\overline{C}_1\,\overline{C}_2\,R_{3,4}\,C_5\,C_6$ // $m_{1,3\to 1}$ // $m_{1,5\to 1}$ // $m_{2,4\to 2}$ // $m_{2,6\to 2}$) == $R_{1,2}$]

Out[∘]= True

RVK and Z

In[∘]:= Knot[8, 17] // PD

KnotTheory: Loading precomputed data in PD4Knots`.

```
Out[∘]= PD[X[6, 2, 7, 1], X[14, 8, 15, 7], X[8, 3, 9, 4], X[2, 13, 3, 14],
  X[12, 5, 13, 6], X[4, 9, 5, 10], X[16, 12, 1, 11], X[10, 16, 11, 15]]
```

RVK, rot, Z modified from 2016-09/OneSmidgen.nb. See also in AP/Projects/SL2Invariant/.

Some details of the code below are at http://drorbn.net/bbs/show?shot=Dror-160920-151350.jpg.

```
In[∘]:= RVK::usage =
  "RVK[xs, rots] represents a Rotational Virtual Knot with a list of n Xp/Xm crossings
    xs and a length 2n list of rotation numbers rots. Crossing
    sites are indexed 1 through 2n, and rots⟦k⟧ is the rotation
    between site k-1 and site k. RVK is also a casting operator
    converting to the RVK presentation from other knot presentations.";
```

```
In[•]:= RVK[pd_PD] := Module[{n, xs, x, rots, front = {0}, k},
    n = Length@pd; rots = Table[0, {2 n}];
    xs = Cases[pd, x_X :> { Xp[x[[4]], x[[1]]]  PositiveQ@x
                            Xm[x[[2]], x[[1]]]  True       ];
    For[k = 0, k < 2 n, ++k, If[k == 0 ∨ FreeQ[front, -k],
      front = Flatten@Replace[front, k → (xs /. {
            Xp[k + 1, l_] | Xm[l_, k + 1] :> {l, k + 1, 1 - l},
            Xp[l_, k + 1] | Xm[k + 1, l_] :> (++rots[[l]]; {1 - l, k + 1, l}),
            _Xp | _Xm :> {}
           }), {1}],
      Cases[front, k | -k] /. {k, -k} :> --rots[[k + 1]];
     ]];
    RVK[xs, rots] ];
RVK[K_] := RVK[PD[K]];

                                    η_i                   n == 0
In[•]:= rot_i_[n_] := rot_i[n] = {  C_$ rot_i[n - 1] // m_{i,$→i}   n > 0
                                    C̄_$ rot_i[n + 1] // m_{i,$→i}   n < 0

In[•]:= Z[K_] := Z[RVK@K];
Z[rvk_RVK] := Module[{ξ, done, st, c, χ, i, j, k},
  ξ = 1; done = {}; st = Range[2 Length[rvk[[1]]]];
  Do[
   {i, j} = List @@ c;
   χ = c /. {_Xp :> R_{i,j}, _Xm :> R̄_{i,j}};
   Do[χ = (rot_0[rvk[[2, k]]] χ) // m_{0,k→k}, {k, {i, j}}];
   ξ *= χ;
   Do[
    If[MemberQ[done, k + 1], ξ = ξ // m_{k,k+1→k}; st = st /. k + 1 → k];
    If[MemberQ[done, k - 1], ξ = ξ // m_{st[[k-1]],k→st[[k-1]]}; st = st /. k → st[[k - 1]]],
    {k, {i, j}}];
   done = done ⋃ {i, j},
   {c, rvk[[1]]}
  ];
  Factor@ξ
 ]

In[•]:= K = Knot[8, 17]; Factor@Alexander[K][T]
z = Z[K]

In[•]:= Timing@Union@Table[Simplify[Alexander[K][T] η_1 == Z[K]], {K, AllKnots[{3, 10}]}]

In[•]:= ZF[K_] := Z@ThinPosition@K;

In[•]:= Timing@Union@Table[Simplify[Alexander[K][T] η_1 == ZF[K]], {K, AllKnots[{3, 10}]}]

In[•]:= Timing[ZF[GST48]]
```