

Pensieve header: A concise implementation of the FastKh algorithm; continues pensieve://2013-06/;
annotated version in pensieve://2017-08/.

Full sources at <http://drorbn.net/AcademicPensieve/2013-07/>.

<< KnotTheory`

Loading KnotTheory` version of September 6, 2014, 13:37:37.2841.

Read more at <http://katlas.org/wiki/KnotTheory>.

```
In[*]:= SetAttributes[{P, S}, Orderless]; dot /: dot[_]k /; k ≥ 2 := 0;
(σ_S)[i_] := σ[i] = First@Cases[σ, P[i, j_] → j];
```

```
In[*]:= EC[λ_List] := Module[{ρ, ec = λ}, (* "Finding Equivalence Classes" *)
  Do[ρ = First /@ Position[ec, i];
    ec = Append[Delete[ec, List /@ ρ], Union @@ (ec[[ρ])],
      {i, Union @@ λ}]; ec];
EC[λ_S] := EC[Join[λ] /. S | P → List];
ECP[λ_] := Union @@ Replace[EC[λ], c_ → ((# → First[c]) & /@ c), {1}];
```

```
In[*]:= VC[β_S, μ_S, τ_S] := VC[β, μ, τ] = Module[{ins, outs, p, χs, h, dec, dots, law},
  ins = First /@ Join[EC[β, μ], EC[μ, τ]];
  outs = First /@ EC[β, τ]; p = ECP[β, μ, τ];
  χs = Times @@ (h /@ Join[ins, outs] /. p);
  χs *= PowerExpand[(Times @@ (h /@ (Last /@ p)))-1/2];
  dec = χs /. h[i_] → (2 dot[i])(2-x)/2;
  dec *= Product[If[i == (i /. p), 1, dot[i] + dot[i /. p]], {i, outs}];
  Expand[dots * # /. law] & /.
    {dots → Expand[dec], law → Table[dot[i] → dot[i /. p], {i, Union[ins]}}];
```

```
In[*]:= m0[i_, j_][σ_S] := m0[i, j][σ] = If[σ[i] == j, DeleteCases[σ, P[i, j]],
  Append[DeleteCases[σ, P[i, _] | P[_ , j]], P[σ[i], σ[j]]]];
m[i_, j_][σ_S] := m0[i, j][σ] * If[σ[i] == j, {q, q-1}, {1}];
m[i_, j_][qk σ_S] := qk m[i, j][σ];
```

```
In[*]:= m[i_, j_][Cob[β_S, τ_S, dots_]] := Module[{p, ijdots, np, ndots, x},
  p = ECP[β, τ]; ijdots = dot@Min[i, j]; np = ECP[m0[i, j][β], m0[i, j][τ]];
  ndots = Which[β[i] == j ∧ τ[i] == j,  $\begin{pmatrix} \text{ijdots} & 0 \\ 1 & \text{ijdots} \end{pmatrix}$ ,
    β[i] == j ∧ τ[i] ≠ j, {{1, ijdots}},
    β[i] ≠ j ∧ τ[i] == j, {{ijdots}, {1}}, β[i] ≠ j ∧ τ[i] ≠ j,
    {{If[(i /. p) ≠ (j /. p), 1, dot[β[i]] + dot[τ[i]]]}];
  ndots = Expand[dots * ndots] /.
    dot[k_] → dot[k /. {i → β[i], j → β[j]} /. {i → τ[i], j → τ[j]} /. np];
  If[β[i] == j ∧ τ[i] == j, Coefficient[ndots /. ijdots → x, x], ndots];
```

```

In[*]:= m[i_, j_][Kom[Ω_, d_]] := Kom[
  Flatten /@ Map[m[i, j], Ω, {2}],
  Table[If[Length@Ω[[k]] == 0 ∨ Length@Ω[[k+1]] == 0, 0,
    Table[m[i, j][Cob[Ω[[k], b], Ω[[k+1], a], d[[k, a, b]] /. q → 1],
      {a, Length@Ω[[k+1]]}, {b, Length@Ω[[k]]}
    ] // ArrayFlatten ],
  {k, Length@d} ]];

```

```

In[*]:= (Kom[Ω_, d_] // Cob[qp1· β_, qp2· τ_, 1]) := Module[{L, ρ, δ, k},
  L = Length[Ω]; ρ[k_] := ρ[k] = Length[Ω[[k]]]; ρ[0] = ρ[L+1] = 0;
  Kom[
    MapThread[Join, List @@@ {
      Append[Ω /. σ_S ⇒ qp1 Join[β, σ], {}],
      Prepend[Ω /. σ_S ⇒ qp2 Join[τ, σ], {}] }],
    Table[
      If[ρ[k] + ρ[k-1] == 0 ∨ ρ[k+1] + ρ[k] == 0, 0,
        δ = Table[0, {ρ[k+1] + ρ[k]}, {ρ[k] + ρ[k-1]}];
        If[ρ[k] ρ[k+1] ≠ 0, δ[[1;; ρ[k+1], 1;; ρ[k]]] = d[[k]];
        If[ρ[k] ≠ 0,
          δ[[ρ[k+1] + 1;; ρ[k+1] + ρ[k], 1;; ρ[k]]] = (-1)k IdentityMatrix[ρ[k]];
        If[ρ[k-1] ρ[k] ≠ 0,
          δ[[ρ[k+1] + 1;; ρ[k+1] + ρ[k], ρ[k] + 1;; ρ[k] + ρ[k-1]]] = d[[k-1]];
        δ
      ], {k, L} ] ] ]

```

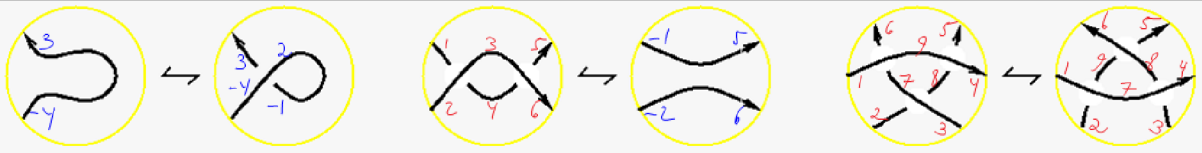
```

In[*]:= Contract[kom_Kom] := Module[{Ω, d, L, ρ, k, done, a, b, φ, γδ},
  {Ω, d} = List @@ kom; L = Length@d; ρ[k_] := Length@Ω[[k]];
  For[k = 1, k ≤ L, ++k,
    done = False; While[! done, done = True;
      For[a = 1, a ≤ ρ[k+1], ++a, For[b = 1, b ≤ ρ[k], ++b,
        If[NumberQ[φ = d[[k, a, b]]] ∧ φ ≠ 0 ∧ Ω[[k+1, a]] == Ω[[k, b]],
          done = False;
          If[ρ[k] ≤ 1 ∨ ρ[k+1] ≤ 1, d[[k]] = 0,
            γδ = Table[VC[Ω[[k, n]] /. q → 1, Ω[[k+1, a]] /. q → 1, Ω[[k+1, m]] /. q → 1] [
              d[[k, a, n] d[[k, m, b]]], {m, ρ[k+1]}, {n, ρ[k]} ];
            d[[k]] = Expand@Drop[d[[k]] - φ-1 γδ, {a}, {b}]]];
            Ω[[k]] = Drop[Ω[[k]], {b}]; Ω[[k+1]] = Drop[Ω[[k+1]], {a}];
            If[k > 1 ∧ ρ[k-1] > 0, d[[k-1]] = Drop[d[[k-1]], {b}]];
            If[k < L ∧ ρ[k+2] > 0, d[[k+1]] = Drop[d[[k+1]], {}, {a}]];
            If[a ≤ ρ[k+1], --a]; b = ρ[k] ] ] ] ];
  Kom[Ω, d] ];

```

```
In[ ]:= Kom[] = Kom[{{S[]}}, {}];
Cob@Xp[i_, j_, k_, L_] := Cob[q S[P[-i, j], P[k, -L]], q^2 S[P[-i, -L], P[j, k]], 1];
Cob@Xm[i_, j_, k_, L_] := Cob[q^-2 S[P[-i, -j], P[k, L]], q^-1 S[P[-i, L], P[-j, k]], 1];
Cob[x_X] := Cob[If[PositiveQ[x], Xp@@x, Xm@@x]];
```

```
In[ ]:= KhComplex[L_] := Module[
  {pd = PD[L], kom = Kom[], inside = {}, pos},
  While[Length[pd] > 0,
    pos = Last[Ordering[(Length[(List @@ #) ∩ inside] & /@ pd]]];
    kom = kom // Cob[pd[[pos]]];
    (kom = Contract[kom // m[#, -#]]) & /@ ((List @@ pd[[pos]]) ∩ inside);
    inside = inside ∪ (List @@ pd[[pos]]); pd = Drop[pd, {pos}];
  ];
  kom];
KhPoly[L_] := Expand[t^(-Length@Select[PD@L, NegativeQ] + Range[0, Crossings[L]]) .
  (List @@ Plus @@@ First @ KhComplex[L]) /. S[] -> 1]
```



```
In[ ]:= Kom[] // Cob[q S[P[-1, 2], P[3, -4]], q^2 S[P[-1, -4], P[2, 3]], 1] // m[-1, 2] // Contract
Out[ ]:= Kom[{{S[P[-4, 3]]}, {}}, {0}]
```

```
In[ ]:= Kom[] // Cob[Xm[1, 2, 4, 3]] // Cob[Xp[4, 6, 5, 3]] // m[3, -3] // m[4, -4] // Contract
Out[ ]:= Kom[{{}}, {S[P[-2, 6], P[-1, 5]]}, {}}, {0, 0}]
```

```
In[ ]:= R31 =
  Kom[] // Cob[Xp[7, 9, 6, 1]] // Cob[Xp[8, 4, 5, 9]] // Cob[Xm[2, 3, 8, 7]] // m[-7, 7] //
  m[-8, 8] // m[-9, 9] // Contract
```

```
Out[ ]:= Kom[{{}}, {q S[P[-3, -2], P[-1, 4], P[5, 6]], q S[P[-3, 4], P[-2, 5], P[-1, 6]]},
  {q^2 S[P[-3, 4], P[-2, -1], P[5, 6]], q^2 S[P[-3, -2], P[-1, 6], P[4, 5]]},
  {q^3 S[P[-3, 6], P[-2, -1], P[4, 5]]}, {0, {{1, -1}, {1, -1}}, {{1, -1}}}]
```

```
In[ ]:= R32 =
  Kom[] // Cob[Xp[2, 7, 9, 1]] // Cob[Xp[3, 4, 8, 7]] // Cob[Xm[9, 8, 5, 6]] // m[-7, 7] //
  m[-8, 8] // m[-9, 9] // Contract
```

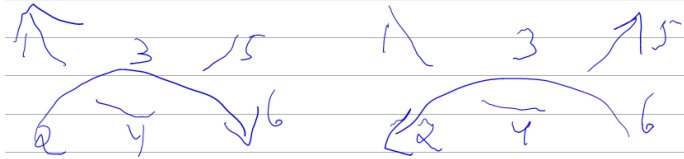
```
Out[ ]:= Kom[{{}}, {q S[P[-3, -2], P[-1, 4], P[5, 6]], q S[P[-3, 4], P[-2, 5], P[-1, 6]]},
  {q^2 S[P[-3, 4], P[-2, -1], P[5, 6]], q^2 S[P[-3, -2], P[-1, 6], P[4, 5]]},
  {q^3 S[P[-3, 6], P[-2, -1], P[4, 5]]}, {0, {{1, -1}, {1, -1}}, {{1, -1}}}]
```

In[]:= **R31 == R32**

Out[]:=

True

Added 250812: We also need to check R2c:



In[]:= **Kom[] // Cob[Xp[4, 3, 1, 2]] // Cob[Xm[5, 3, 4, 6]] // m[3, -3] // m[4, -4] // Contract**

Out[]:=

Kom[{ {}, {S[P[-5, 1], P[-2, 6]]}, {}}, {0, 0}]

In[]:= **Kom[] // Cob[Xm[1, 2, 4, 3]] // Cob[Xp[4, 6, 5, 3]] // m[3, -3] // m[4, -4] // Contract**

Out[]:=

Kom[{ {}, {S[P[-2, 6], P[-1, 5]]}, {}}, {0, 0}]

In[]:= **K = TorusKnot[9, 5]; {TubePlot[K, ImageSize -> 80] // Rasterize, KhPoly[K]} // Timing**

Out[]:=

{428.078,



{ $q^{31} + q^{33} + q^{35} t^2 + q^{39} t^3 + q^{37} t^4 + q^{39} t^4 + q^{41} t^5 + q^{43} t^5 + q^{39} t^6 + q^{41} t^6 + q^{43} t^7 + q^{45} t^7 +$

$q^{41} t^8 + 2 q^{43} t^8 + q^{45} t^9 + 2 q^{47} t^9 + 2 q^{45} t^{10} + 3 q^{49} t^{11} + 2 q^{47} t^{12} + 2 q^{49} t^{12} + q^{53} t^{12} + 3 q^{51} t^{13} +$
 $2 q^{53} t^{13} + q^{49} t^{14} + 2 q^{51} t^{14} + q^{55} t^{14} + 2 q^{53} t^{15} + 3 q^{55} t^{15} + 2 q^{53} t^{16} + q^{57} t^{16} + q^{59} t^{16} +$
 $3 q^{57} t^{17} + q^{55} t^{18} + q^{57} t^{18} + q^{61} t^{18} + 2 q^{59} t^{19} + q^{61} t^{19} + q^{59} t^{20} + q^{63} t^{20} + q^{63} t^{21}\}$