Pensieve header: A concise implementation of the FastKh algorithm.

```
<< KnotTheory`
```

```
Loading KnotTheory` version of February 5, 2013, 3:48:46.4762.
Read more at http://katlas.org/wiki/KnotTheory.
```

```
SetAttributes[{P, S}, Orderless];
dot /: dot[_]^k_ /; k ≥ 2 := 0;
```

one line

```
(σ_S)[i_] := σ[i] = First@Cases[σ, P[i, j_] :> j];

ECP[λ_List] := Module[{ρ, ec = λ}, (* "Equivalence Class Projection" *)
  Do[ ρ = First /@ Position[ec, i];
   ec = Append[Delete[ec, List /@ ρ], Union @@ (ec[[ρ]])],
   {i, Union @@ λ} ];
  Union @@ Replace[ec, c_ :> ((# → First[c]) & /@ c), {1}] ];
```

split here.

```
ECP[λ__S] := ECP[Join[λ] /. S | P → List];
ECR[λ__] := Union[Last /@ ECP[λ]] (* "Equiv. Class Representatives" *);

VCLaw[β_S, μ_S, τ_S] := VCLaw[β, μ, τ] = Module[
   {p, ins1, ins2, outs, χs, h, law1, law2, dec} ,
   p = ECP[β, μ, τ];
   ins1 = ECR[β, μ]; ins2 = ECR[μ, τ]; outs = ECR[β, τ];
   χs = (Times @@ (h /@ Join[ins1, ins2, outs] /. p)) / PowerExpand[(Times @@ (h /@ (Last /@ p)))^(1/2)];
```

split to 2 lines?

```
   dec = χs /. h[i_]^χ_. :> (2 dot[i])^((2-χ)/2);
   dec *= Product[If[i == (i /. p), 1, dot[i] + dot[i /. p]], {i, outs}];
   law1 = Table[dot[i] → dot[i /. p], {i, ins1}];
   law2 = Table[dot[i] → dot[i /. p], {i, ins2}];
```

one law is enough! call it dotlaw.

```
   {law1, law2, Expand[dec]}];

VC[Cob[β_S, μ_S, dots1_], Cob[μ_S, τ_S, dots2_]] := Module[
  {law1, law2, dec}, {law1, law2, dec} = VCLaw[β, μ, τ];
  Expand[dec * (dots1 /. law1) (dots2 /. law2)] ];

m0[i_, j_][σ_S] := m0[i, j][σ] = Which[
   σ[i] ≠ j, Append[DeleteCases[σ, P[i, _] | P[_, j]], P[σ[i], σ[j]]],
   σ[i] == j, DeleteCases[σ, P[i, j]]];
```

2-line "If"?

```
m[i_, j_][σ_S] := m0[i, j][σ] * If[σ[i] ≠ j, {1}, {q, q^-1}];
m[i_, j_][q^k_. σ_S] := q^k m[i, j][σ];
```

reverse order.

np

np=

```
m[i_, j_][Cob[β_S, τ_S, dots_]] := Module[{p, ijdot, ndots, x},
  p = ECP[β, τ]; ijdot = dot[Min[i, j]];
  ndots = Which[
    β[i] ≠ j && τ[i] ≠ j, {{If[(i /. p) ≠ (j /. p), 1, dot[β[i]] + dot[τ[i]]]}},
    β[i] == j && τ[i] ≠ j, {{1, ijdot}},
    β[i] ≠ j && τ[i] == j, {{ijdot}, {1}},
    β[i] == j && τ[i] == j, (ijdot   0
                             1    ijdot)];
  ndots = Expand[dots * ndots] /.
    dot[k_] :> dot[k /. {i → β[i], j → β[j]} /. {i → τ[i], j → τ[j]} /.
      ECP[m0[i, j][β], m0[i, j][τ]]];
  If[β[i] == j && τ[i] == j, Coefficient[ndots /. ijdot → x, x], ndots]];
```

np

Re-evaluated each time?

```
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]] /. q → 1, Ω[[k + 1, a]] /. q → 1, d[[k, a, b]]]],
      {a, Length[Ω[[k + 1]]]}, {b, Length[Ω[[k]]]}
     ] // ArrayFlatten ],
   {k, Length[d]}] ];
```

one line

```
(Kom[Ω_, d_] // Cob[q^p1_ · β_, q^p2_ · τ_, 1]) := Module[{L, ρ, δ, k},
 L = Length[Ω]; ρk_ := ρk = Length[Ω[[k]]]; ρ0 = ρL+1 = 0;
 Kom[
  MapThread[Join, List @@@ {
     Append[Ω /. σ_S :> q^p1 Join[β, σ], {}],
     Prepend[Ω /. σ_S :> q^p2 Join[τ, σ], {}] }],
  Table[
   If[ρk + ρk-1 == 0 || ρk+1 + ρk == 0, 0,
    δ = Table[0, {ρk+1 + ρk}, {ρk + ρk-1}];
    If[ρk ρk+1 ≠ 0, δ[[1 ;; ρk+1, 1 ;; ρk]] = d[[k]]];
    If[ρk ≠ 0, δ[[ρk+1 + 1 ;; ρk+1 + ρk, 1 ;; ρk]] = (-1)^k IdentityMatrix[ρk]];
    If[ρk-1 ρk ≠ 0, δ[[ρk+1 + 1 ;; ρk+1 + ρk, ρk + 1 ;; ρk + ρk-1]] = d[[k - 1]]];
    δ
   ], {k, L} ]]]
```

change to Table Language

Try to re-write in matrix form.

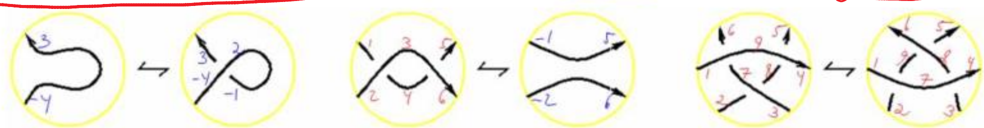
Dror Bar-Natan: Academic Pensieve: 2013-06: ConciseFastKh.nb 2013-06-26 09:39:18

```
Contract[kom_Kom] := Module[{Ω, d, L, ρ, k, done, a, b, ϕ, γδ},
  {Ω, d} = List @@ kom; L = Length[d]; ρk_ := Length[Ω[[k]]];
  For[k = 1, k ≤ L, ++k,
   done = False; While[! done, done = True;
    For[a = 1, a ≤ ρk+1, ++a, For[b = 1, b ≤ ρk, ++b,
      If[NumberQ[ϕ = d[[k, a, b]]] && ϕ ≠ 0 && Ω[[k + 1, a]] == Ω[[k, b]],
       done = False;
       If[ρk ≤ 1 || ρk+1 ≤ 1, d[[k]] = 0,
        γδ = Table[VC[
            Cob[Ω[[k, n]], Ω[[k + 1, a]], d[[k, a, n]]] /. q → 1,
            Cob[Ω[[k, b]], Ω[[k + 1, m]], d[[k, m, b]]] /. q → 1
           ], {m, ρk+1}, {n, ρk}];
        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 && d[[k - 1]] =!= 0, d[[k - 1]] = Drop[d[[k - 1]], {b}]];
       If[k < L && d[[k + 1]] =!= 0, d[[k + 1]] = Drop[d[[k + 1]], {}, {a}]];
       If[a ≤ ρk+1, --a]; b = ρk; ] ]] ] ];
  Kom[Ω, d] ];

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]];

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]
```

consider replacing the "while" with a recursive call to contract.

Rewrite in terms of ρ

Put dividing line



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

```
Kom[] // Cob[Xm[1, 2, 4, 3]] // Cob[Xp[4, 6, 5, 3]] // m[3, -3] // m[4, -4] //
  Contract
```

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

```
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
```

```
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}}}]
```

```
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
```

```
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}}}]
```

```
R31 == R32
```

```
True
```

```
K = TorusKnot[9, 5]; {TubePlot[K, ImageSize → 80] // Rasterize, KhPoly[K]} // Timing
```

{762.470488,

{, $q^{31} + q^{33} + q^{35} t^{2} + q^{39} t^{3} + q^{37} t^{4} + 2 q^{39} t^{4} + q^{41} t^{4} + 2 q^{41} t^{7} + 2 q^{43} t^{7} + 2 q^{41} t^{8} +$
$2 q^{43} t^{8} + 2 q^{45} t^{8} + q^{47} t^{8} + 4 q^{45} t^{9} + 4 q^{47} t^{9} + q^{49} t^{9} + 2 q^{45} t^{10} + 2 q^{47} t^{10} + q^{47} t^{11} +$
$2 q^{49} t^{11} + q^{51} t^{11} + 2 q^{47} t^{12} + 2 q^{49} t^{12} + q^{51} t^{12} + 2 q^{51} t^{13} + q^{53} t^{13} + q^{49} t^{14} + 4 q^{51} t^{14} +$
$4 q^{53} t^{14} + q^{55} t^{14} + 3 q^{51} t^{15} + 8 q^{53} t^{15} + 5 q^{55} t^{15} + 5 q^{53} t^{16} + 8 q^{55} t^{16} + 2 q^{57} t^{16} +$
$q^{53} t^{17} + 6 q^{55} t^{17} + 7 q^{57} t^{17} + 3 q^{59} t^{17} + q^{55} t^{18} + 6 q^{57} t^{18} + 4 q^{59} t^{18} + 2 q^{57} t^{19} +$
$4 q^{59} t^{19} + 4 q^{61} t^{19} + q^{63} t^{19} + 4 q^{59} t^{20} + 6 q^{61} t^{20} + 2 q^{63} t^{20} + q^{59} t^{21} + 2 q^{61} t^{21} + q^{63} t^{21}$}}

Consider standardizing smoothing labels.

Consider dot[i] ⟶ •;