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_. := 0;
(σ_S)[i_] := σ[i] = First@Cases[σ, P[i, j_] ↦ j];

ECP[λ_List] := Module[{ρ, ec}, (* "Equivalence Class Projection" *)
   ec = Fold[
      (ρ = First /@ Position[#1, #2];
       Append[Delete[#1, List /@ ρ], Union @@ (#1[[ρ]])]) &,
      λ, Union @@ λ
     ][[1]] // SortBy[#, First] &;
   Union @@ Replace[ec, c_ ↦ ((# → First[c]) & /@ c), {1}]];
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)];
    dec = χs /. h[i_]^k_. ↦ (2 dot[i])^((2-χ)/2);
    dec *= Times @@ MapThread[If[#1 == #2, 1, dot[#1] + dot[#2]] &,
       {outs, outs /. p}];
    law1 = dot /@ ins1; law1 = Thread[law1 → (law1 /. p)];
    law2 = dot /@ ins2; law2 = Thread[law2 → (law2 /. p)];
    {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] := Which[
   σ[i] ≠ j, Append[DeleteCases[σ, P[i, _] | P[_, j]], P[σ[i], σ[j]]],
   σ[i] == j, DeleteCases[σ, P[i, j]]];
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];
```

```
m[i_, j_][Cob[β_S, τ_S, dots_]] := Module[
   {β, τ, p, ijdot, ndots, x},
   {β, τ} = m0[i, j] /@ {β, τ};
   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[nβ, nτ]];
   If[β[i] == j && τ[i] == j, Coefficient[ndots /. ijdot → x, x], ndots]];

Kom /: Kom[cs_, ds_] * Cob[q^p1_. β_, q^p2_. τ_, 1] := Module[{L, ρ, d, k},
  L = Length[cs]; ρ_k_ := ρ_k = Length[cs[[k]]]; ρ_0 = ρ_L+1 = 0;
  Kom[
   MapThread[Join, List @@@ {
      Append[cs /. σ_S :> q^p1 Join[β, σ], {}],
      Prepend[cs /. σ_S :> q^p2 Join[τ, σ], {}] }],
   Table[
    If[(ρ_k + ρ_k-1) (ρ_k+1 + ρ_k) == 0, 0,
     d = Table[0, {ρ_k+1 + ρ_k}, {ρ_k + ρ_k-1}];
     If[k ≤ L && ρ_k ρ_k+1 ≠ 0, d[[1 ;; ρ_k+1, 1 ;; ρ_k]] = ds[[k]]];
     If[k ≤ L && ρ_k ≠ 0, d[[ρ_k+1 + 1 ;; ρ_k+1 + ρ_k, 1 ;; ρ_k]] = (-1)^k IdentityMatrix[ρ_k]];
     If[k > 1 && ρ_k-1 ρ_k ≠ 0, d[[ρ_k+1 + 1 ;; ρ_k+1 + ρ_k, ρ_k + 1 ;; ρ_k + ρ_k-1]] = ds[[k - 1]]];
     d
    ], {k, L} ] ] ]

m[i_, j_][Kom[cs_, ds_]] := Kom[
   Flatten /@ Map[m[i, j], cs, {2}],
   Table[
    If[Length[cs[[k]]] == 0 || Length[cs[[k + 1]]] == 0, 0,
     Table[
       m[i, j][Cob[cs[[k, b]] /. q → 1, cs[[k + 1, a]] /. q → 1, ds[[k, a, b]]]],
       {a, Length[cs[[k + 1]]]}, {b, Length[cs[[k]]]}
     ] // ArrayFlatten ],
    {k, Length[ds]}] ];
```

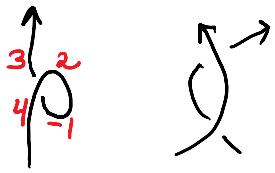
```
Contract[kom_Kom] := Module[{cs, ds, L, k, done, a, b, ϕ, γδ},
  {cs, ds} = List @@ kom; L = Length[ds];
  For[k = 1, k ≤ L, ++k,
   done = False; While[! done, done = True;
    For[a = 1, a ≤ Length[cs[[k + 1]]], ++a, For[b = 1, b ≤ Length[cs[[k]]], ++b,
     If[NumberQ[ϕ = ds[[k, a, b]]] && ϕ ≠ 0 && cs[[k + 1, a]] == cs[[k, b]],
      done = False;
      If[Length[cs[[k]]] > 1 && Length[cs[[k + 1]]] > 1,
       γδ = Table[
         VC[Cob[cs[[k, d]], cs[[k + 1, a]], ds[[k, a, d]]] /. q → 1,
          Cob[cs[[k, b]], cs[[k + 1, c]], ds[[k, c, b]]] /. q → 1],
         {c, Length[cs[[k + 1]]]}, {d, Length[cs[[k]]]}];
       ds[[k]] = Expand[Drop[ds[[k]] - ϕ^-1 γδ, {a}, {b}]],
       (* else *) ds[[k]] = 0];
      cs[[k]] = Drop[cs[[k]], {b}]; cs[[k + 1]] = Drop[cs[[k + 1]], {a}];
      If[k > 1, ds[[k - 1]] = If[ds[[k - 1]] === 0, 0, Drop[ds[[k - 1]], {b}]]];
      If[k < L,
       ds[[k + 1]] = If[ds[[k + 1]] === 0, 0, Drop[ds[[k + 1]], {}, {a}]]];
      If[a ≤ Length[cs[[k + 1]]], --a]; b = Length[cs[[k]]]; ]]]];
  Kom[cs, ds]];

CFKh[L_] := Module[
  {pd = PD[L], kom = Kom[{{S[]}}, {}], inside = {}, tp = 0, pos},
  While[Length[pd] > 0,
   pos = Last[Ordering[(Length[(List @@ #) ⋂ inside]) & /@ pd]];
   kom = kom // (pd[[pos]] /. {
        X[i_, j_, k_, l_] /; (j - l == 1 || l - j > 1) ↦
         Cob[q S[P[-i, j], P[k, -l]], q^2 S[P[-i, -l], P[j, k]], 1],
        X[i_, j_, k_, l_] /; (l - j == 1 || j - l > 1) ↦
         (--tp; Cob[q^-2 S[P[-i, -j], P[k, l]], q^-1 S[P[-i, l], P[-j, k]], 1])
       });
   (kom = Contract[kom // m[#, -#]]) & /@ ((List @@ pd[[pos]]) ⋂ inside);
   inside = inside ⋃ (List @@ pd[[pos]]); pd = Drop[pd, {pos}]];
  Expand[t^tp-1+Range[Length[First[kom]]].(List @@ Plus @@@ First @ kom) /. S[] → 1]]
```

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

```
{1083.848148,
```

```
, 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}}
```


Add proofs of R123



consider standardizing smoothing labels.

consider dot[i] ⟶ •;