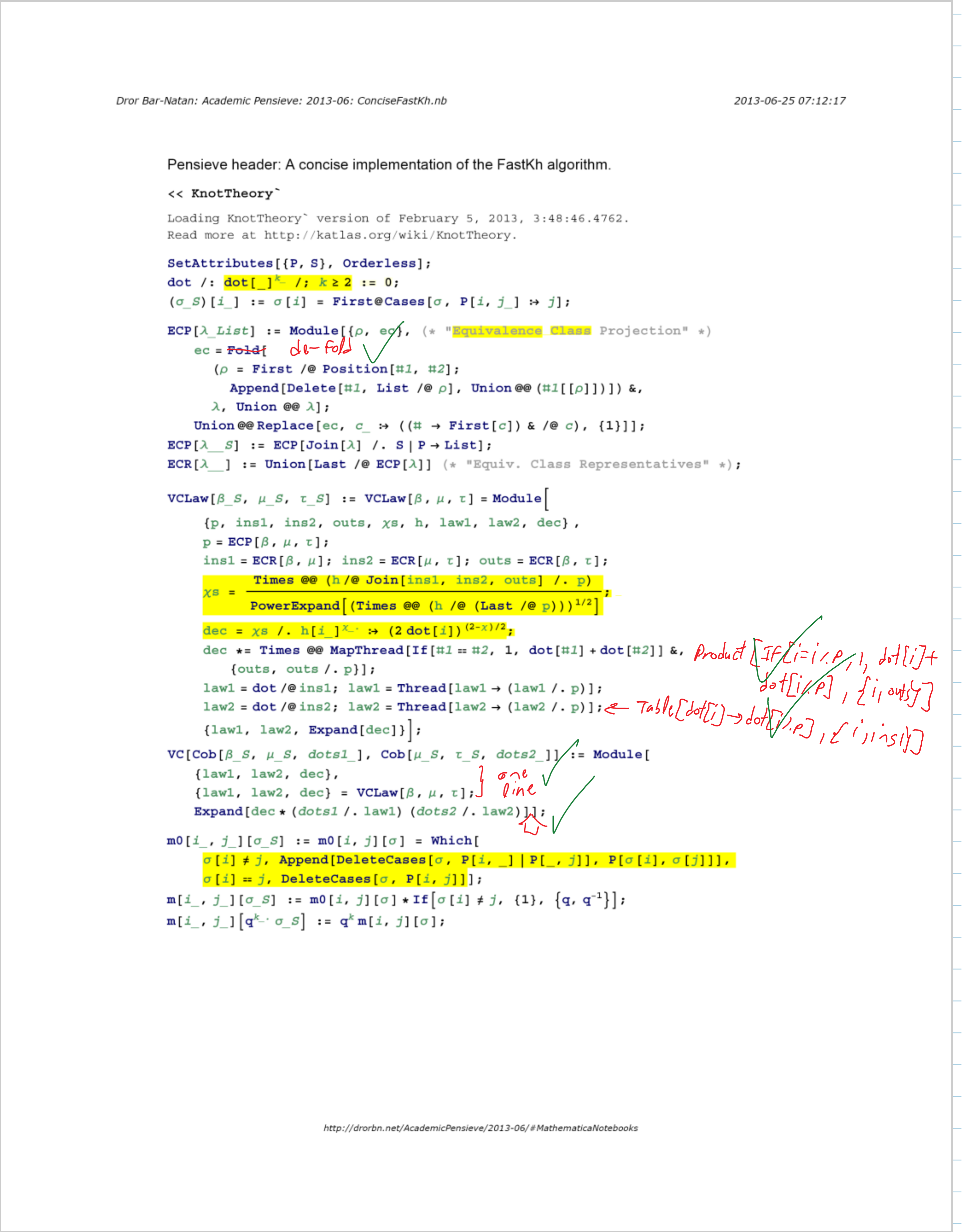
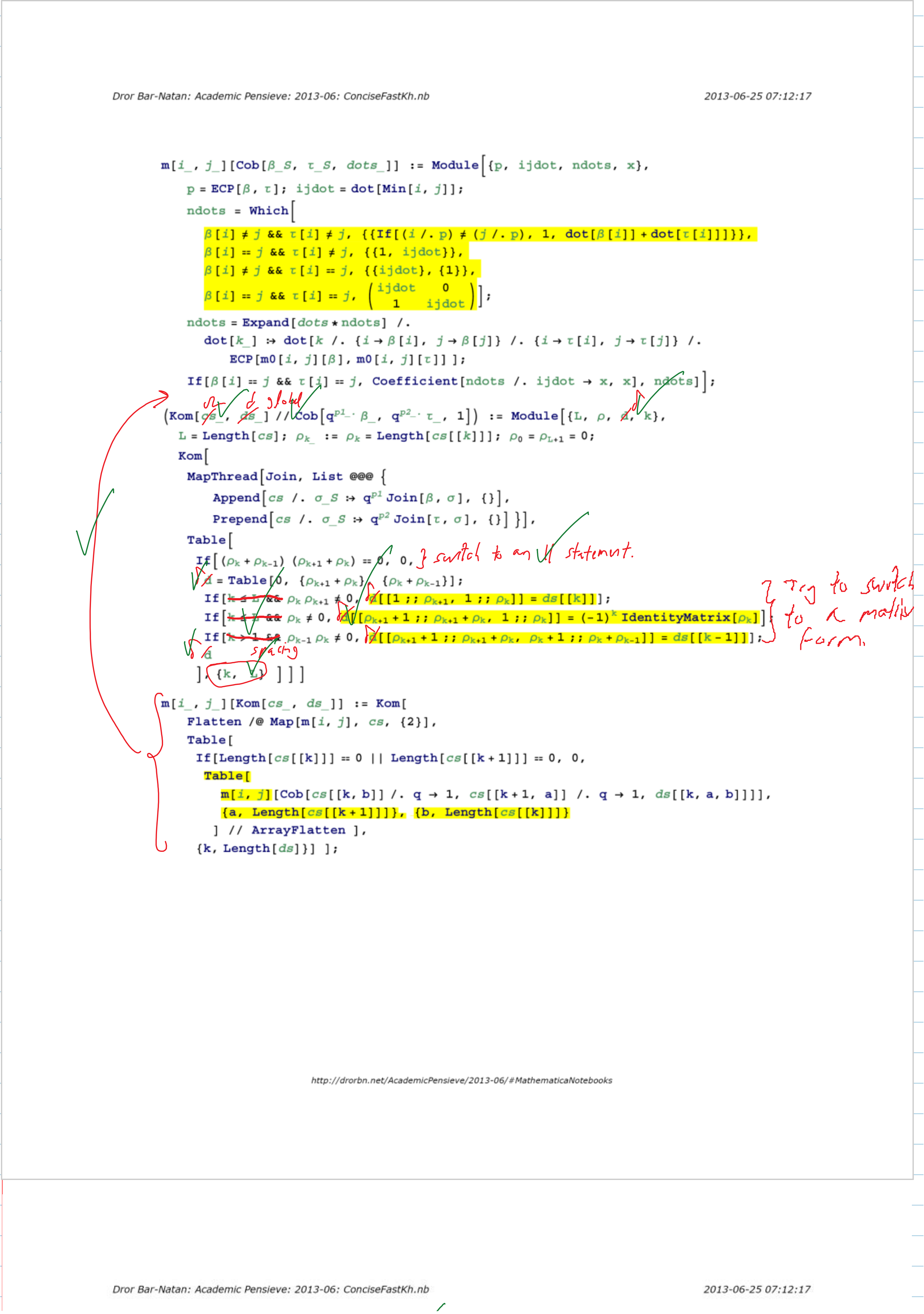
ConciseFastKh on June 25, 2013

June-25-13
4:02 PM



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

ECP[λ_List] := Module[{ρ, ec}, (* "Equivalence Class Projection" *)
   ec = ~~Fold~~[        de-fold
     (ρ = First /@ Position[#1, #2];
       Append[Delete[#1, List /@ ρ], Union @@ (#1[[ρ]])]) &,
     λ, Union @@ λ];
   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];
          Times @@ (h /@ Join[ins1, ins2, outs] /. p)
    χs = ------------------------------------------------ ;
         PowerExpand[(Times @@ (h /@ (Last /@ p)))^(1/2)]
    dec = χs /. h[i_]^χ_. ⧴ (2 dot[i])^((2-χ)/2);
    dec *= Times @@ MapThread[If[#1 == #2, 1, dot[#1] + dot[#2]] &,     Product[If[i==i/.p, 1, dot[i]+
       {outs, outs /. p}];                                                 dot[i/.p]], {i, outs}]
    law1 = dot /@ ins1; law1 = Thread[law1 → (law1 /. p)];
    law2 = dot /@ ins2; law2 = Thread[law2 → (law2 /. p)]; ← Table[dot[i] → dot[i/.p], {i, ins1}]
    {law1, law2, Expand[dec]}];

VC[Cob[β_S, μ_S, dots1_], Cob[μ_S, τ_S, dots2_]] := Module[
   {law1, law2, dec},                  } one
   {law1, law2, dec} = VCLaw[β, μ, τ];  } line
   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]]];
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[{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]];

(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]}] ];
```

σs d global

d

switch to an If statement.

Try to switch to a matrix form.

spacing

highlighted stuff

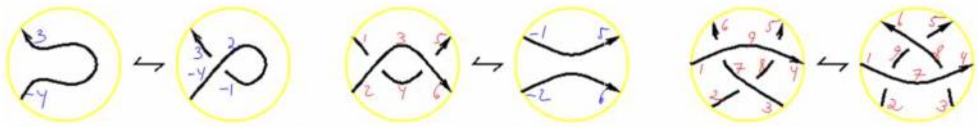
→ ∫ₖ

ds[[k]] → d[[k]]

cs[[k]] → Ω[[k]]

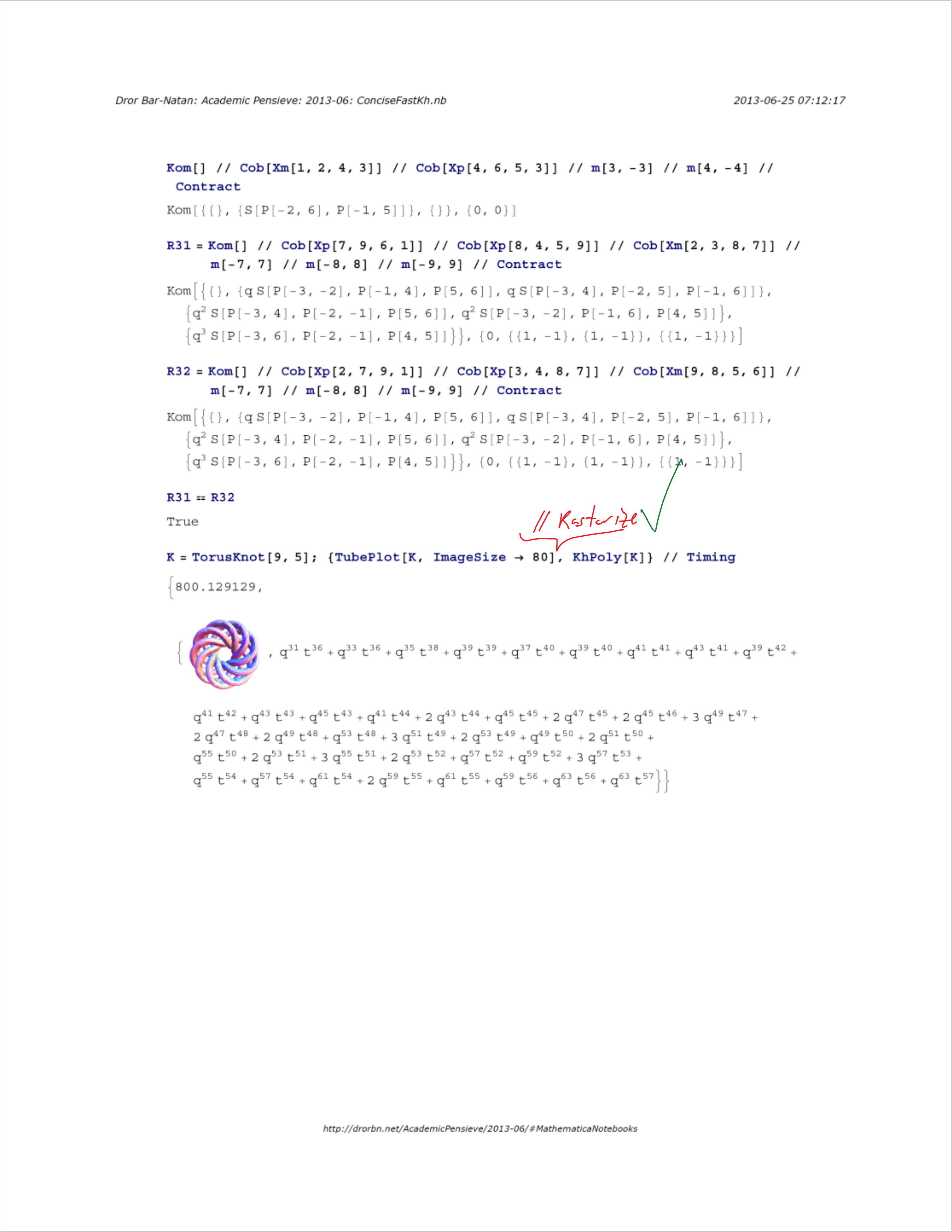
```
                                       Ω     d
Contract[kom_Kom] := Module[{~~cs~~, ~~ds~~, L, k, done, a, b, ϕ, γδ},
    {cs, ds} = List @@ kom; L = Length[ds];
    For[k = 1, k ≤ L, ++k,
     done = False; While[! done, done = True;
      For[a = 1, a ≤ Length[cs[[k + 1]]], ++a, For[b = 1, b ≤ 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, ds[[k]] = 0,
          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}]] ];
         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]];
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]
```

Rephrase 2 lines.



```
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
```

// Rasterize ✓

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

{800.129129,
```

$$\{\ ,\ q^{31}t^{36}+q^{33}t^{36}+q^{35}t^{38}+q^{39}t^{39}+q^{37}t^{40}+q^{39}t^{40}+q^{41}t^{41}+q^{43}t^{41}+q^{39}t^{42}+$$
$$q^{41}t^{42}+q^{43}t^{43}+q^{45}t^{43}+q^{41}t^{44}+2q^{43}t^{44}+q^{45}t^{45}+2q^{47}t^{45}+2q^{45}t^{46}+3q^{49}t^{47}+$$
$$2q^{47}t^{48}+2q^{49}t^{48}+q^{53}t^{48}+3q^{51}t^{49}+2q^{53}t^{49}+q^{49}t^{50}+2q^{51}t^{50}+$$
$$q^{55}t^{50}+2q^{53}t^{51}+3q^{55}t^{51}+2q^{53}t^{52}+q^{57}t^{52}+q^{59}t^{52}+3q^{57}t^{53}+$$
$$q^{55}t^{54}+q^{57}t^{54}+q^{61}t^{54}+2q^{59}t^{55}+q^{61}t^{55}+q^{59}t^{56}+q^{63}t^{56}+q^{63}t^{57}\}\}$$

Consider standardizing smoothing labels.

Consider dot[i] ⟶ ◉;