In[•]:=
```
Once[<< KnotTheory`];
```

In[•]:=
```
RibbonKnots = {Knot[6, 1], Knot[8, 8], Knot[8, 9], Knot[8, 20], Knot[9, 27],
  Knot[9, 41], Knot[9, 46], Knot[10, 3], Knot[10, 22], Knot[10, 35], Knot[10, 42],
  Knot[10, 48], Knot[10, 75], Knot[10, 87], Knot[10, 99], Knot[10, 123],
  Knot[10, 129], Knot[10, 137], Knot[10, 140], Knot[10, 153], Knot[10, 155]}
```

Out[•]=
```
{Knot[6, 1], Knot[8, 8], Knot[8, 9], Knot[8, 20], Knot[9, 27], Knot[9, 41],
 Knot[9, 46], Knot[10, 3], Knot[10, 22], Knot[10, 35], Knot[10, 42],
 Knot[10, 48], Knot[10, 75], Knot[10, 87], Knot[10, 99], Knot[10, 123],
 Knot[10, 129], Knot[10, 137], Knot[10, 140], Knot[10, 153], Knot[10, 155]}
```

In[•]:=
```
ThinPosition[K_] := Module[{todo, done, pd, c},
  todo = List @@ PD@K; done = {}; pd = PD[];
  While[todo =!= {},
   AppendTo[pd, c = RandomChoice@MaximalBy[todo, Length[done ⋂ List @@ #] &]];
   todo = DeleteCases[todo, c];
   done = done ⋃ List @@ c];
  pd ]
```

In[•]:=
```
PD[GST48] = PD[X[1, 15, 2, 14], X[29, 2, 30, 3], X[40, 4, 41, 3],
   X[4, 44, 5, 43], X[5, 26, 6, 27], X[95, 7, 96, 6], X[7, 1, 8, 96], X[8, 14, 9, 13],
   X[28, 9, 29, 10], X[41, 11, 42, 10], X[11, 43, 12, 42], X[12, 27, 13, 28],
   X[15, 31, 16, 30], X[61, 16, 62, 17], X[72, 17, 73, 18], X[83, 18, 84, 19],
   X[34, 20, 35, 19], X[20, 89, 21, 90], X[92, 21, 93, 22], X[22, 79, 23, 80],
   X[23, 68, 24, 69], X[24, 57, 25, 58], X[56, 25, 57, 26], X[31, 63, 32, 62],
   X[32, 74, 33, 73], X[33, 85, 34, 84], X[35, 50, 36, 51], X[81, 37, 82, 36],
   X[70, 38, 71, 37], X[59, 39, 60, 38], X[54, 39, 55, 40], X[55, 45, 56, 44],
   X[45, 59, 46, 58], X[46, 70, 47, 69], X[47, 81, 48, 80], X[91, 49, 92, 48],
   X[49, 91, 50, 90], X[82, 52, 83, 51], X[71, 53, 72, 52], X[60, 54, 61, 53],
   X[74, 63, 75, 64], X[85, 64, 86, 65], X[65, 76, 66, 77], X[66, 87, 67, 88],
   X[94, 67, 95, 68], X[86, 75, 87, 76], X[77, 88, 78, 89], X[93, 78, 94, 79]];
```

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 = {1}, 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 = 1, k ≤ 2 n, ++k,
    If[FreeQ[front, -k],
     front = Flatten@Replace[front, k → (xs /. {
             Xp[k, l_] | Xm[l_, k] :> {l + 1, k + 1, -l},
             Xp[l_, k] | Xm[k, l_] :> (++rots[[l]]; {-l, k + 1, l + 1}),
             _Xp | _Xm :> {}
            }), {1}],
     Cases[front, k | -k] /. {k, -k} :> --rots[[k]];
    ]
   ];
   RVK[xs, rots] ];
RVK[K_] := RVK[PD[K]];
```

$$\text{rot}_{i\_}[n\_] := (*\text{rot}_i[n]=*)\begin{cases} \eta_i & n == 0 \\ C_\$\,\text{rot}_i[n-1] \;//\; m_{i,\$\to i} & n > 0 \\ \overline{C}_\$\,\text{rot}_i[n+1] \;//\; m_{i,\$\to i} & n < 0 \end{cases}$$

In[•]:=
```
Z[K_] := Z[RVK@K];
Z[rvk_RVK] := Module[{ζ, done, st, c, χ, i, j, k},
  ζ = 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}}];
   ζ *= χ;
   Do[
    If[MemberQ[done, k + 1], ζ = ζ // m_{k,k+1→k}; st = st /. k + 1 → k];
    If[MemberQ[done, k - 1], ζ = ζ // 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@ζ
 ]
```

In[•]:=
```
ZF[K_] := Z@ThinPosition@K;
```