Banana Seifert

In[•]:=
```
SetDirectory["C:\\Users\\T15Roland\\Wiskunde\\Bn\\Seifert"]
Once[<< KnotTheory`]
```

Out[•]=
```
C:\Users\T15Roland\Wiskunde\Bn\Seifert
```

ParentDirectory: Argument File should be a positive machine–size integer, a nonempty string, or a File specification.

ParentDirectory: Argument File should be a positive machine–size integer, a nonempty string, or a File specification.

ToFileName: String or list of strings expected at position 1 in ToFileName[{File, WikiLink, mathematica}].

ToFileName: String or list of strings expected at position 1 in ToFileName[{File, QuantumGroups}].

```
Loading KnotTheory` version of September 6, 2014, 13:37:37.2841.
Read more at http://katlas.org/wiki/KnotTheory.
```

SetDelayed: Tag Diff in Diff[K_PD, rut_, ag_, n_, m_] is Protected.

In[•]:=
```
SetAttributes[P, Orderless]
ProcessSigns[pd_] :=
 pd /. X[a_, b_, c_, d_] :> If[PositiveQ[X[a, b, c, d]], Xp[a, b, c, d], Xm[a, b, c, d]]
SeifertCycles[pd_] := (Times @@ pd) /.
    {Xp[a_, b_, c_, d_] :> P[a, b] P[c, d], Xm[a_, b_, c_, d_] :> P[a, d] P[c, b]} //.
  {P[a___, b_] P[b_, c___] :> P[a, b, c], x_^2 -> x}
SeifertBycles[pd_] := (Times @@ pd) /.
    {Xp[a_, b_, c_, d_] :> Q[a] Q[b] Q[d, c], Xm[a_, b_, c_, d_] :> Q[a, d] Q[c] Q[b]} //.
  {Q[a___, b_] Q[b_, c___] :> Q[a, b, c]}
CutList[pd_] := First /@ List @@ (SeifertCycles[pd])
IsIncoming_z_[XX_] :=
 If[(z == 1) || (Head[XX] === Xp && (z == 4)) || (Head[XX] === Xm && (z == 2)), True, False]
CutPD[pd_] := Module[{pd2 = pd, pos},
  Do[
   pos = Position[pd, c];
   If[IsIncoming_pos[[1,2]][pd[[pos[[1, 1]]]]],
    Part[pd2, Sequence @@ pos[[1]]] = $[c];
    Part[pd2, Sequence @@ pos[[2]]] = $$[c],
    Part[pd2, Sequence @@ pos[[2]]] = $[c];
    Part[pd2, Sequence @@ pos[[1]]] = $$[c]];

   , {c, CutList[pd]}];
  pd2]
SeifertBGraph[K_] :=
 Module[{pd2 = CutPD[ProcessSigns@PD[K]], Edges, Signs = <||>, XV, OV, V},
  XV =
   (List @@ pd2) /. {Xp[a_, b_, c_, d_] :> {-a, b, -d}, Xm[a_, b_, c_, d_] :> {-a, -b, c}};
```

```
  Edges = XV // Flatten;
  (List @@ pd2) /. {Xp[a_, b_, c_, d_] :> (Signs = Append[Signs, {-a -> -1, b -> 1, -d -> -1}]),
    Xm[a_, b_, c_, d_] :> (Signs = Append[Signs, {-a -> 1, c -> 1, -b -> -1}])};
  OV = If[Length[#] == 1, Switch[Head[#[[1]]], Integer, {-#[[1]], #[[1]]}, $, -#, $$, #],
      Join[If[IntegerQ[#[[1]]], {#[[1]]}, {}],
       -Most[#],
       If[IntegerQ[#[[-1]]], {-#[[-1]]}, {}]
      ]

     ] & /@ (List @@ (SeifertBycles[pd2] /. x_^2 -> x) /. Q -> List);
  {V = Join[OV, XV], Signs}
 ]


DrawGraph[sb_] := Module[{i, neww, newedges = {},
   edges = Gather[Table[Labeled[UndirectedEdge @@ (First /@ Position[sb[[1]], e]),
       sb[[2]][e]], {e, sb[[2]] // Keys}], First[#1] === First[#2] &]},
  Do[If[Length[w] == 1, AppendTo[newedges , w[[1]]],
    neww = Table[{Labeled[UndirectedEdge[w[[i, 1, 1]], {w, i}], w[[i, 2]],
       Labeled[UndirectedEdge[w[[i, 1, 2]], {w, i}], ""], If[i > 1,
        Style[Labeled[UndirectedEdge[{w, i - 1}, {w, i}], ""], White], Nothing]},
      {i, Length[w]}];
    newedges = Join[newedges, neww];


   ]
   , {w, edges}];
  PlanarGraph[Flatten[newedges, 1],
   GraphLayout -> "TutteEmbedding", VertexSize -> {_List -> Tiny}, ImageSize -> Tiny]

 ]


(*Get rid of leaves and degree 2 vertices*)
Prune[sb_] := Module[{sbnew = sb, Leaves = Flatten@Select[sb[[1]], Length[#] == 1 &]},
  Do[
   sbnew[[1]] = DeleteCases[sbnew[[1]], lv, {2}];
   sbnew[[2]] = Delete[sbnew[[2]], Key[lv]]
   , {lv, Leaves}];
  sbnew /. {{} -> Nothing}]
Contract[sb_] :=
 Module[{sbnew = sb, SmallV, cand = Select[sb[[1]], Length[#] == 2 &], asso},
  If[Length[cand] === 0, sbnew,
   SmallV = First@cand;
   asso = sbnew[[2]];
   asso [First@SmallV] = asso[First@SmallV] + asso[Last@SmallV];
   sbnew[[2]] = Delete[asso, Key[Last@SmallV]];
```

```
    sbnew[[1]] = DeleteCases[sbnew[[1]], SmallV];
    sbnew[[1]] = sbnew[[1]] /. {Last@SmallV → First@SmallV};
    sbnew]
  ]
MergeVs_{x_,y_}[X_, Y_] := Join[Most@RotateLeft[X, x], Most@RotateLeft[Y, y]]
Del0[sb_] := Module[{sbnew = sb, e, v1, v2, cand = Select[Keys[sb[[2]]], sb[[2]][#] == 0 &]},
  If[Length[cand] === 0, sbnew,
   e = First@cand;
   {v1, v2} = Position[sb[[1]], e];
   sbnew[[1]] = Delete[sbnew[[1]], {{v1[[1]]}, {v2[[1]]}}];
   sbnew[[1]] = Append[sbnew[[1]], MergeVs_{v1[[2]],v2[[2]]}[sb[[1, v1[[1]]]], sb[[1, v2[[1]]]]]];
   sbnew[[2]] = Delete[sbnew[[2]], Key[e]];
   sbnew
   ]
  ]
(*Repeated simplification of the graph*)
SimpGr[G_] := FixedPoint[Contract[Del0[#]] &, Prune[G]]
```

In[•]:=
```
PD@Knot[4, 1]
```

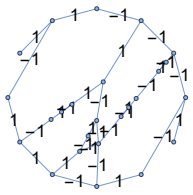
Out[•]=
```
PD[X[4, 2, 5, 1], X[8, 6, 1, 5], X[6, 3, 7, 4], X[2, 7, 3, 8]]
```

In[•]:=
```
SeifertBGraph[Knot[4, 1]]
```

Out[•]=
```
{{{-7, 7}, {-$[3]}, {$$[2]}, {$$[3]}, {6, -6, -4}, {-$[2], -8}, {-$[1], -5},
   {-4, $$[2], -$[1]}, {-8, 6, -5}, {-6, -$[3], 7}, {-$[2], -7, $$[3]}},
  ‹|-4 → -1, $$[2] → 1, -$[1] → -1, -8 → -1, 6 → 1, -5 → -1, -6 → 1,
   7 → 1, -$[3] → -1, -$[2] → 1, $$[3] → 1, -7 → -1|›}
```

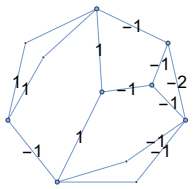
In[•]:=
```
DrawGraph@SeifertBGraph[Knot[10, 117]]
```

Out[•]=



In[•]:=
```
DrawGraph@SimpGr@SeifertBGraph[Knot[10, 117]]
```

Out[•]=



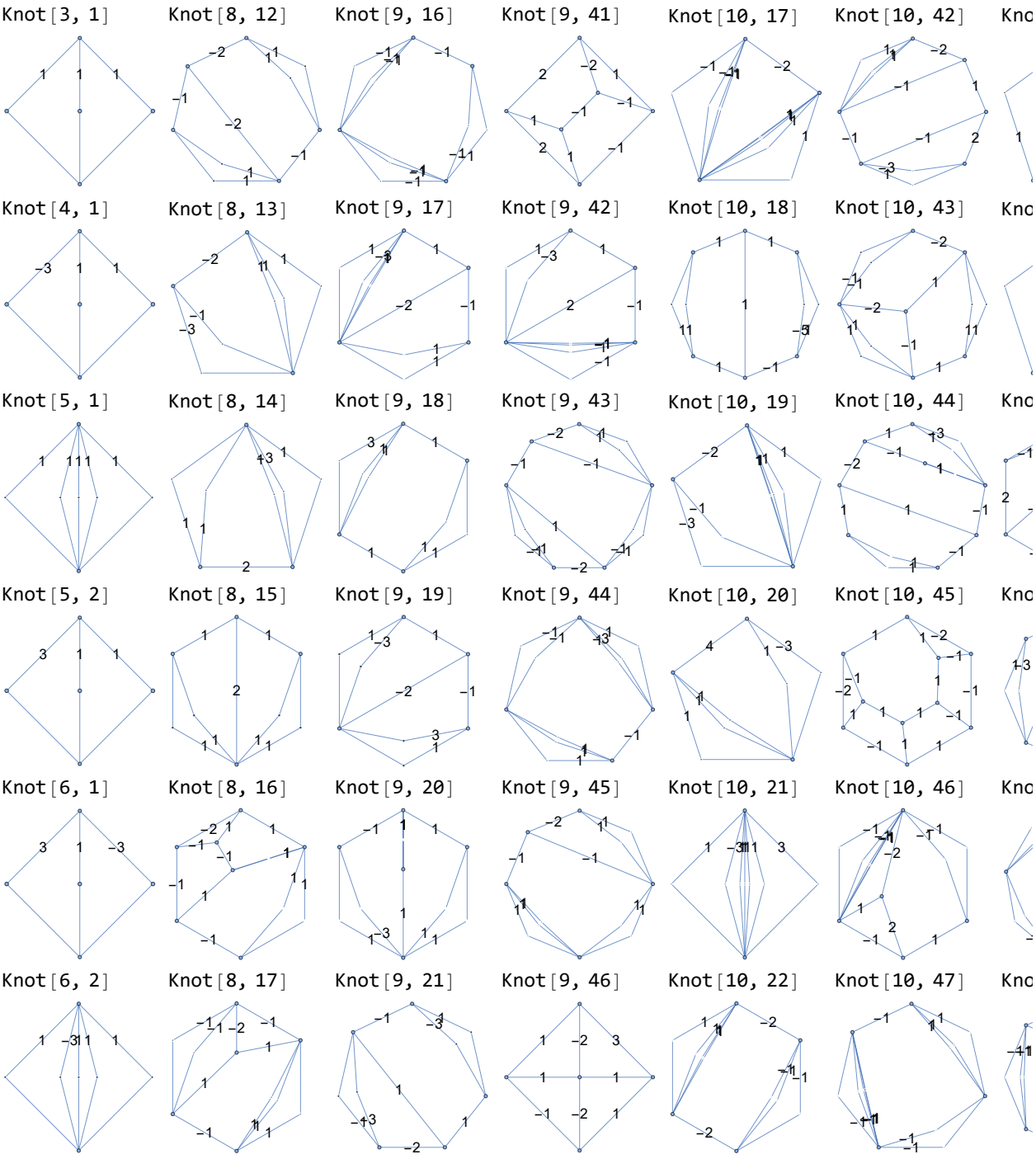
```
In[•]:= SimpGr@SeifertBGraph[Knot[4, 1]]
```

```
Out[•]=
{{{6, -6, -4}, {6, -4, -6}}, ⟨|-4 → -3, 6 → 1, -6 → 1|⟩}
```

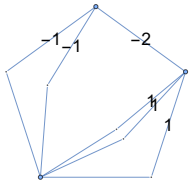
Seifert surfaces for the Rolfsen table

```
In[•]:= Multicolumn[
 TableForm@{#, (DrawGraph@SimpGr@SeifertBGraph[#])} & /@ AllKnots[{3, 10}], 10]
```

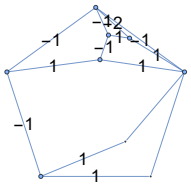
```
Out[•]=
```



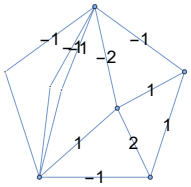
Knot[6, 3]



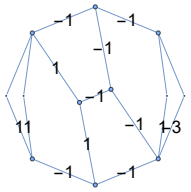
Knot[8, 18]



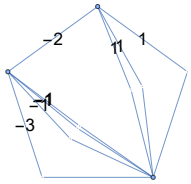
Knot[9, 22]



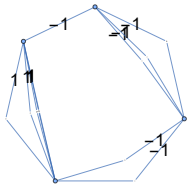
Knot[9, 47]



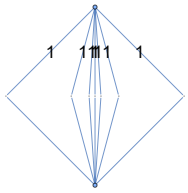
Knot[10, 23]



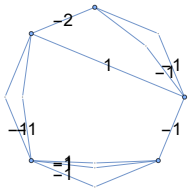
Knot[10, 48]



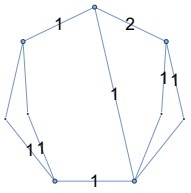
Knot[7, 1]



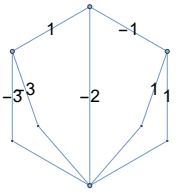
Knot[8, 19]



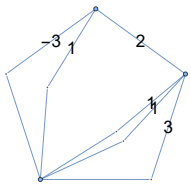
Knot[9, 23]



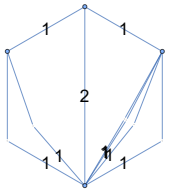
Knot[9, 48]



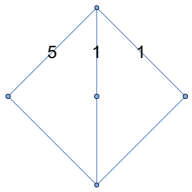
Knot[10, 24]



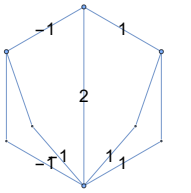
Knot[10, 49]



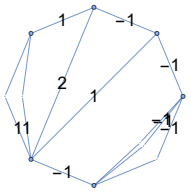
Knot[7, 2]



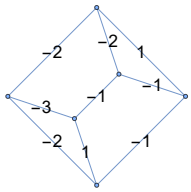
Knot[8, 20]



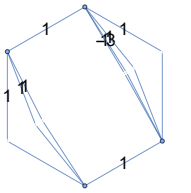
Knot[9, 24]



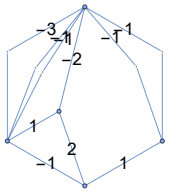
Knot[9, 49]



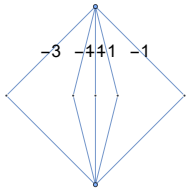
Knot[10, 25]



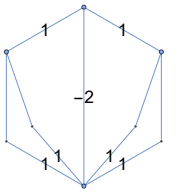
Knot[10, 50]



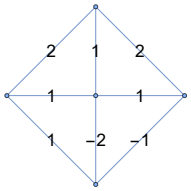
Knot[7, 3]



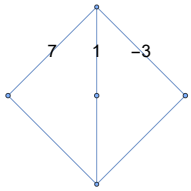
Knot[8, 21]



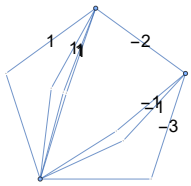
Knot[9, 25]



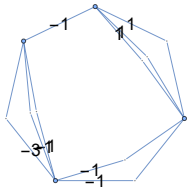
Knot[10, 1]



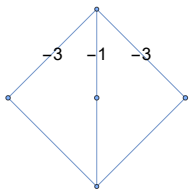
Knot[10, 26]



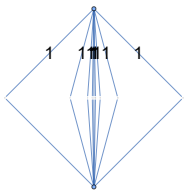
Knot[10, 51]



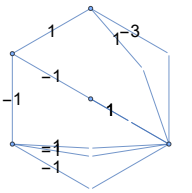
Knot[7, 4]



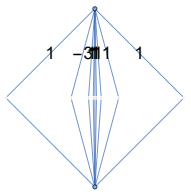
Knot[9, 1]



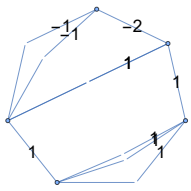
Knot[9, 26]



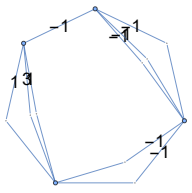
Knot[10, 2]



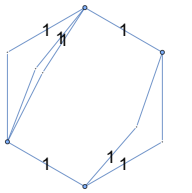
Knot[10, 27]



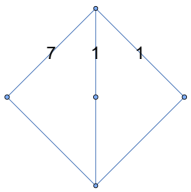
Knot[10, 52]



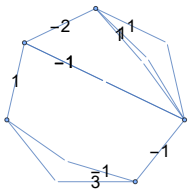
Knot[7, 5]



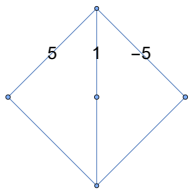
Knot[9, 2]



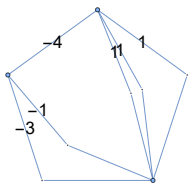
Knot[9, 27]



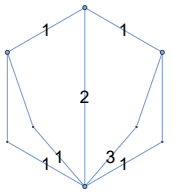
Knot[10, 3]



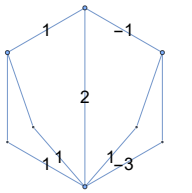
Knot[10, 28]



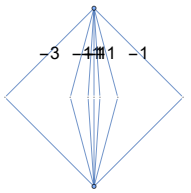
Knot[10, 53]



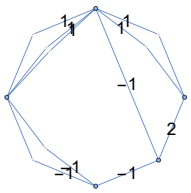
Knot[7, 6]



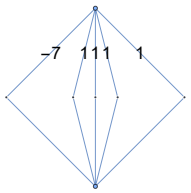
Knot[9, 3]



Knot[9, 28]



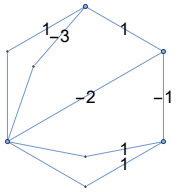
Knot[10, 4]



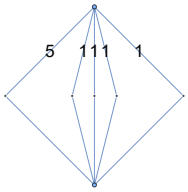
Knot[10, 29]

Knot[10, 54]

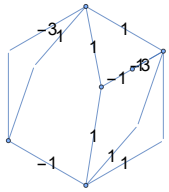
Knot[7, 7]



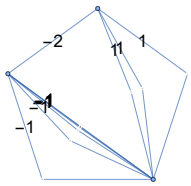
Knot[9, 4]



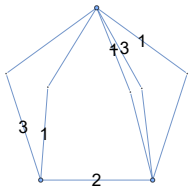
Knot[9, 29]



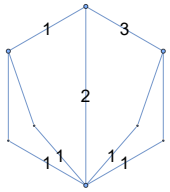
Knot[10, 5]



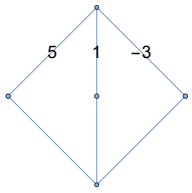
Knot[10, 30]



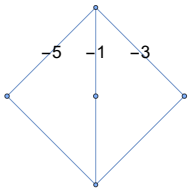
Knot[10, 55]



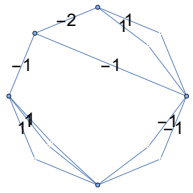
Knot[8, 1]



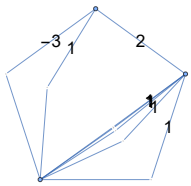
Knot[9, 5]



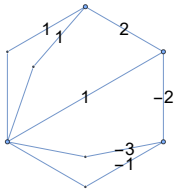
Knot[9, 30]



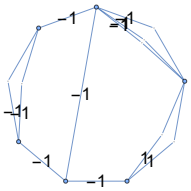
Knot[10, 6]



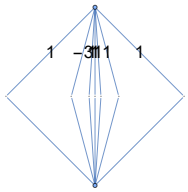
Knot[10, 31]



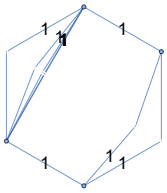
Knot[10, 56]



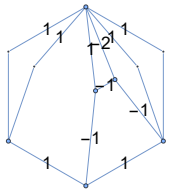
Knot[8, 2]



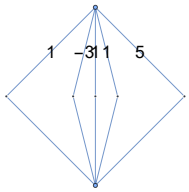
Knot[9, 6]



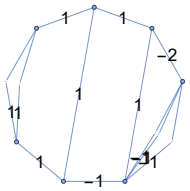
Knot[9, 31]



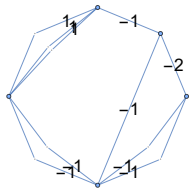
Knot[10, 7]



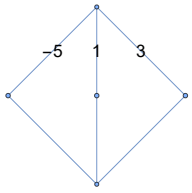
Knot[10, 32]



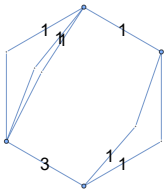
Knot[10, 57]



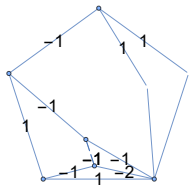
Knot[8, 3]



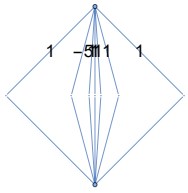
Knot[9, 7]



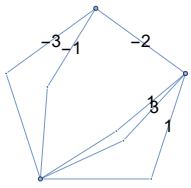
Knot[9, 32]



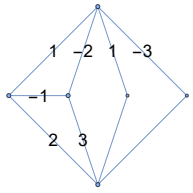
Knot[10, 8]



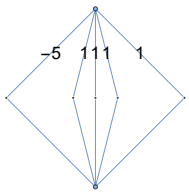
Knot[10, 33]



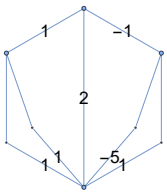
Knot[10, 58]



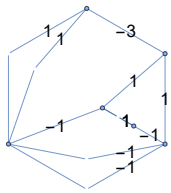
Knot[8, 4]



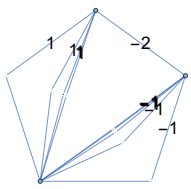
Knot[9, 8]



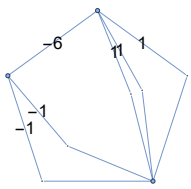
Knot[9, 33]



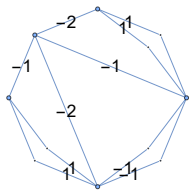
Knot[10, 9]



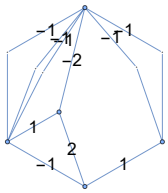
Knot[10, 34]



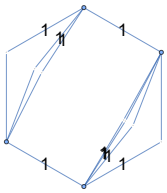
Knot[10, 59]



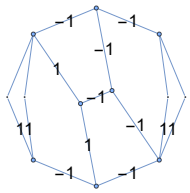
Knot[8, 5]



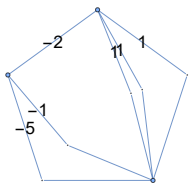
Knot[9, 9]



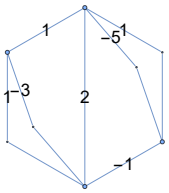
Knot[9, 34]



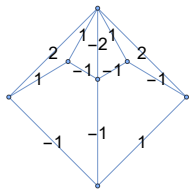
Knot[10, 10]



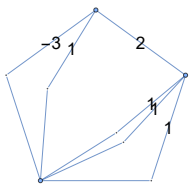
Knot[10, 35]



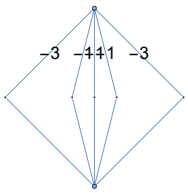
Knot[10, 60]



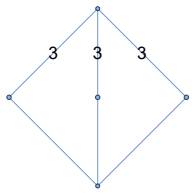
Knot[8, 6]



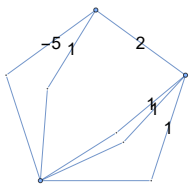
Knot[9, 10]



Knot[9, 35]



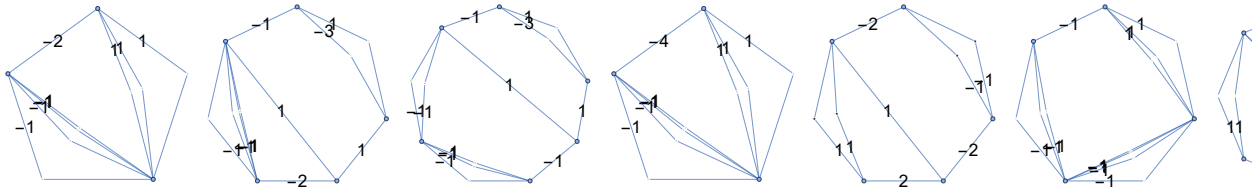
Knot[10, 11]



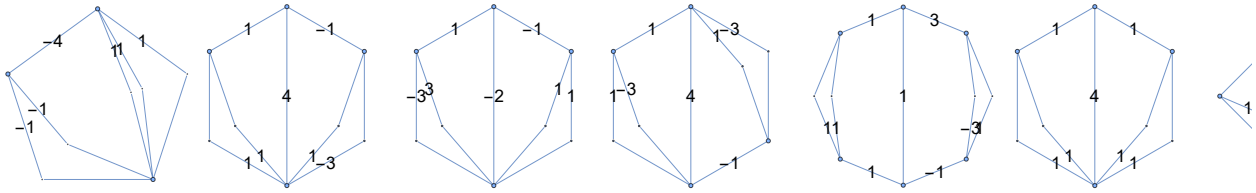
Knot[10, 36]

Knot[10, 61]

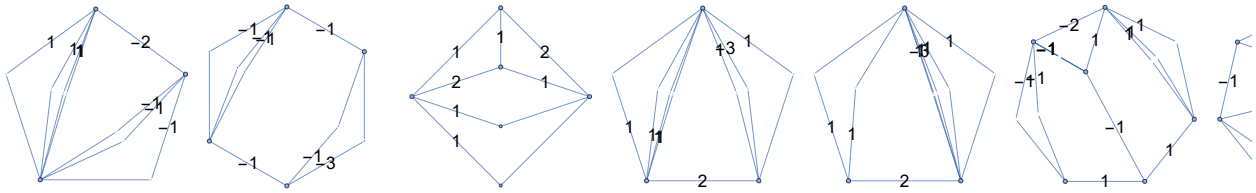
Knot[8, 7] Knot[9, 11] Knot[9, 36] Knot[10, 12] Knot[10, 37] Knot[10, 62] Kno



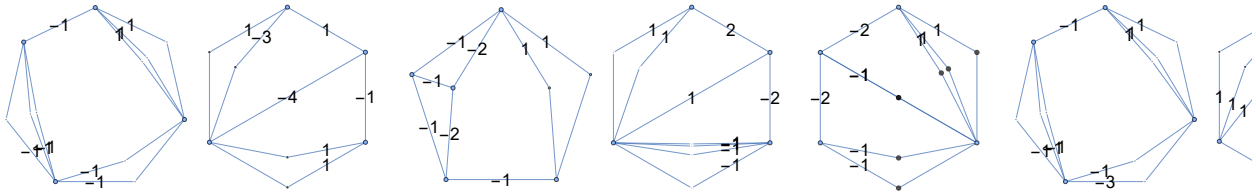
Knot[8, 8] Knot[9, 12] Knot[9, 37] Knot[10, 13] Knot[10, 38] Knot[10, 63] Kno



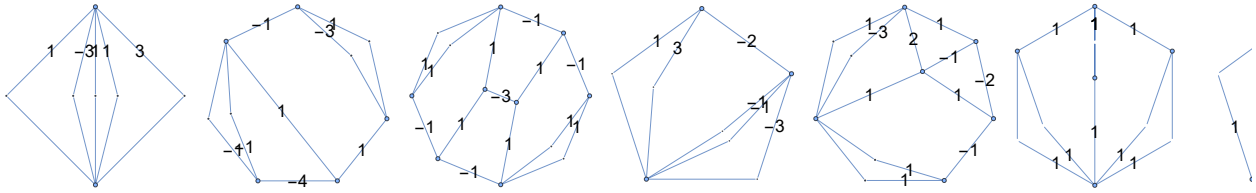
Knot[8, 9] Knot[9, 13] Knot[9, 38] Knot[10, 14] Knot[10, 39] Knot[10, 64] Kno



Knot[8, 10] Knot[9, 14] Knot[9, 39] Knot[10, 15] Knot[10, 40] Knot[10, 65] Kno



Knot[8, 11] Knot[9, 15] Knot[9, 40] Knot[10, 16] Knot[10, 41] Knot[10, 66] Kno



Converting back to PD code

In[•]:=

```
Walk[sb_] := Module[{now, next, vs, lab, sel, path},
  {vs, lab} = sb;
  now = {1, vs[[1, 1]], 1};
  sel = First@Select[Position[vs, now[[2]]], #[[1]] ≠ 1 &];
  next = {sel[[1]], vs[[Sequence @@ sel]], If[OddQ@lab[now[[2]]], now[[3]], -now[[3]]]};
  path = {now, next};
  Do[
   now =
    {next[[1]], vs[[next[[1]], Mod[sel[[2]] + next[[3]], Length[vs[[next[[1]]]]], 1]]], -next[[3]]};
   AppendTo[path, now];
   sel = First@Select[Position[vs, now[[2]]], #[[1]] ≠ now[[1]] &];
```

```
    next = {sel[[1]], vs[[Sequence @@ sel]], If[OddQ@lab[now[[2]]], now[[3]], -now[[3]]]};
    AppendTo[path, next];
    , {i, 1, 2 Length[Keys[lab]] - 1}];
   path


 ]
MakeTwistRegion[a_, b_, s_, n_, l_] :=
 Switch[{s, l},
  {-1, -1}, Table[
   If[OddQ[k], X[b - k, a + k, b - k + 1, a + k - 1], X[a + k - 1, b - k + 1, a + k, b - k]], {k, n}],
  {1, -1}, Table[
   If[OddQ[k], X[a + k - 1, b - k, a + k, b - k + 1], X[b - k, a + k - 1, b - k + 1, a + k]], {k, n}],
  {-1, 1}, Table[
   If[OddQ[k], X[a + k - 1, b - k + 1, a + k, b - k], X[b - k, a + k, b - k + 1, a + k - 1]], {k, n}],
  {1, 1}, Table[
   If[OddQ[k], X[b - k, a + k - 1, b - k + 1, a + k], X[a + k - 1, b - k, a + k, b - k + 1]], {k, n}]
 ]

Gr2PD[sb_] := Module[{vs, lab, path, pos, pd = {}, a, b, s, m},
  {vs, lab} = sb;
  path = Walk[sb];
  Do[
   pos = Select[Position[path, e], #[[2]] == 2 &];

   a = 1 + 1/2 Sum[Abs[lab[path[[p, 2]]]], {p, pos[[1, 1]] - 1}];
   b = a + 1/2 Sum[Abs[lab[path[[p, 2]]]], {p, pos[[1, 1]], pos[[3, 1]] - 1}] +
     Abs[lab[path[[pos[[3, 1]], 2]]]];
   (*{e,pos,a,b}//Echo;*)
   pd = Join[pd, MakeTwistRegion[a, b, Sign[lab[e]], Abs[lab[e]], path[[pos[[1, 1]]]][[3]]]]


   , {e, Keys[lab]}];
  m = Max[pd /. X[u_, v_, w_, x_] :> Sequence[u, v, w, x]];
  PD @@ (pd /. {m -> 1})
 ]
```

In[•]:=
```
SimpGr@SeifertBGraph[Knot[8, 5]]
```

Out[•]=
```
{{{4, -12, -14}, {12, -12, -6},
  {-6, -7, -3, -9, -15, -11}, {4, -3, -7}, {12, -11, -15, -9, -14}},
 <|-6 -> -2, 4 -> 1, -3 -> -1, -7 -> -1, -14 -> -1, -9 -> -1, -12 -> 2, 12 -> 1, -11 -> -1, -15 -> -1|>}
```

```
In[•]:= Gr2PD[SimpGr@SeifertBGraph[Knot[8, 5]]]
```

```
Out[•]=
PD[X[7, 23, 8, 22], X[21, 9, 22, 8], X[4, 1, 5, 2], X[19, 3, 20, 2],
 X[3, 21, 4, 20], X[5, 13, 6, 12], X[11, 19, 12, 18], X[24, 13, 1, 14],
 X[14, 23, 15, 24], X[6, 15, 7, 16], X[9, 17, 10, 16], X[17, 11, 18, 10]]
```

```
In[•]:= SimpGr@SeifertBGraph[Knot[8, 17]]
```

```
Out[•]=
{{{12, -12, -6}, {8, -8, -4, -10},
  {-6, -7, -15, -11}, {12, -11, -15, -10}, {8, -7, -12, -4, -8}},
 ‹|-6 → -2, 8 → 1, -7 → -1, -8 → 1, -12 → 1, -4 → 1, 12 → 1, -11 → -1, -10 → -1, -15 → -1|›}
```

Verification usig Theta

```
In[•]:= Rot[pd_PD] := Module[{n, xs, x, rots, Xp, Xm, 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]];
    ]
   ];
   {xs /. {Xp[i_, j_] :> {+1, i, j}, Xm[i_, j_] :> {-1, i, j}}, rots}];
Rot[K_] := Rot[PD[K]];
```

```
In[•]:= CF[ℰ_] := Module[{vs = Union@Cases[ℰ, g__, ∞], ps, c},
   Total[CoefficientRules[Expand[ℰ], vs] /. (ps_ → c_) :> Factor[c] (Times @@ vs^ps)]];
```

```
In[•]:= T3 = T1 T2;
```

```
In[•]:= R1[s_, i_, j_] =
  CF[s (1/2 - g3ii + T2^s g1ii g2ji - g1ii g2jj - (T2^s - 1) g2ji g3ii + 2 g2jj g3ii - (1 - T3^s) g2ji g3ji -
      g2ii g3jj - T2^s g2ji g3jj + g1ii g3jj + ((T1^s - 1) g1ji (T2^2 s g2ji - T2^s g2jj + T2^s g3jj) +
        (T3^s - 1) g3ji (1 - T2^s g1ii - (T1^s - 1) (T2^s + 1) g1ji + (T2^s - 2) g2jj + g2ij)) / (T2^s - 1))];
```

```
In[•]:= Θ[{s0_, i0_, j0_}, {s1_, i1_, j1_}] := CF[
  s1 (T1^s0 - 1) (T2^s1 - 1)^-1 (T3^s1 - 1) g1,j1,i0 g3,j0,i1 ((T2^s0 g2,i1,i0 - g2,i1,j0) - (T2^s0 g2,j1,i0 - g2,j1,j0))]
```

In[•]:=
```
Γ1[φ_, k_] = -φ / 2 + φ g3kk;
```

In[•]:=
```
Θ[K_] := Module[{Cs, φ, n, A, s, i, j, k, Δ, G, ν, α, β, gEval, c, z},
  {Cs, φ} = Rot[K]; n = Length[Cs];
  A = IdentityMatrix[2 n + 1];
  Cases[Cs, {s_, i_, j_} :> (A[[{i, j}, {i + 1, j + 1}]] += ({{-T^s, T^s - 1}, {0, -1}}))];
  (*A//Echo;*)
  Δ = T^((-Total[φ] - Total[Cs[[All, 1]]])/2) Det[A];
  G = Inverse[A];
  gEval[ℰ_] := Factor[ℰ /. g_ν_,α_,β_ :> (G[[α, β]] /. T -> T_ν)];
  z = gEval[∑_{k1=1}^{n} ∑_{k2=1}^{n} Θ[Cs[[k1]], Cs[[k2]]]];
  z += gEval[∑_{k=1}^{n} R1 @@ Cs[[k]]];
  z += gEval[∑_{k=1}^{2 n} Γ1[φ[[k]], k]];
  {Δ, (Δ /. T -> T1) (Δ /. T -> T2) (Δ /. T -> T3) z} // Factor];
```

```
In[•]:= PolyPlot[{Δ_, Θ_}] := Module[{crs, m, m1, m2, maxc, minc, s, , rect, hex}, GraphicsColumn[{
          rect = {{0, 0}, {1, 0}, {1, 1}, {0, 1}};
          hex = Table[{Cos[α], Sin[α]} / Cos[2 π / 12] / 2, {α, 2 π / 12, 2 π, 2 π / 6}];
          If[Expand[Δ] === 0, Graphics[],
           crs = CoefficientRules[T^(m=-Exponent[Δ,T,Min]) Δ, {T}];
           maxc = N@Log@Max@Abs[Last /@ crs];
           minc = N@Log@Min@Select[Abs[Last /@ crs], # > 0 &];
           If[minc == maxc, s[_] = 0, s[c_] := s[c] = (maxc - Log@c) / (maxc - minc)];
           Graphics[crs /. ({x_} → c_) :> {
               Lighter[Which[c == 0, White, c > 0, Red, c < 0, Blue], 0.88 s[Abs@c]],
               Polygon[({x + m - 1 / 2, 0} + #) & /@ rect] }, AspectRatio → 1 / 5 ]
          ],
          If[Expand[Θ] === 0, Graphics[{White, Disk[]}],
           crs = CoefficientRules[T1^(m1=-Exponent[Θ,T1,Min]) T2^(m2=-Exponent[Θ,T2,Min]) Θ, {T1, T2}];
           maxc = N@Log@Max@Abs[Last /@ crs];
           minc = N@Log@Min@Select[Abs[Last /@ crs], # > 0 &];
           If[minc == maxc, s[_] = 0, s[c_] := s[c] = (maxc - Log@c) / (maxc - minc)];
           Graphics[{{{White, Disk[{0, 0}, 1 + Cos[2 π / 12] Norm[{m1, m2}] / √2 ]},
             crs /. ({x1_, x2_} → c_) :> {
                 Lighter[Which[c == 0, White, c > 0, Red, c < 0, Blue], 0.88 s[Abs@c]],
                 Polygon[(({{1, -1 / 2}, {0, √3 / 2}}).{x1 - m1, x2 - m2} + #) & /@ hex] }
           }]
          ]
         }, Spacings → 0]];

In[•]:= VerifyInverse[K_] := Θ[K] -
          Θ[Gr2PD[SimpGr@SeifertBGraph[K]]]

        Print[{#, VerifyInverse[#]}] & /@ AllKnots[{3, 10}];
```

```
{Knot[3, 1], {0, 0}}
```

Out[∘]=

```
{Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null,
 Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null, Null}
```

In[∘]:=

```
Gr2PD[SimpGr@SeifertBGraph[Knot[8, 17]]]
```

Out[∘]=

```
PD[X[9, 19, 10, 18], X[17, 11, 18, 10], X[14, 7, 15, 8],
 X[21, 9, 22, 8], X[6, 13, 7, 14], X[22, 11, 1, 12], X[12, 5, 13, 6],
 X[16, 1, 17, 2], X[19, 3, 20, 2], X[15, 5, 16, 4], X[3, 21, 4, 20]]
```

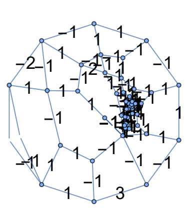
In[∘]:=

```
PDGST48 = 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[∘]:=

```
DrawGraph@SimpGr@SeifertBGraph[PDGST48]
```

Out[∘]=



In[•]:=
```
VerifyInverse[PDGST48]
```

Out[•]=
```
{0, 0}
```

In[•]:=
```
PolyPlot[θ[PDGST48]]
```

Out[•]=