```
SetDirectory["C:\\Users\\T15Roland\\Wiskunde\\Bn\\Theta"];
Once[<< KnotTheory`]
```

Out[∘]=

```
C:\Users\T15Roland\Wiskunde\Bn\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[∘]:=

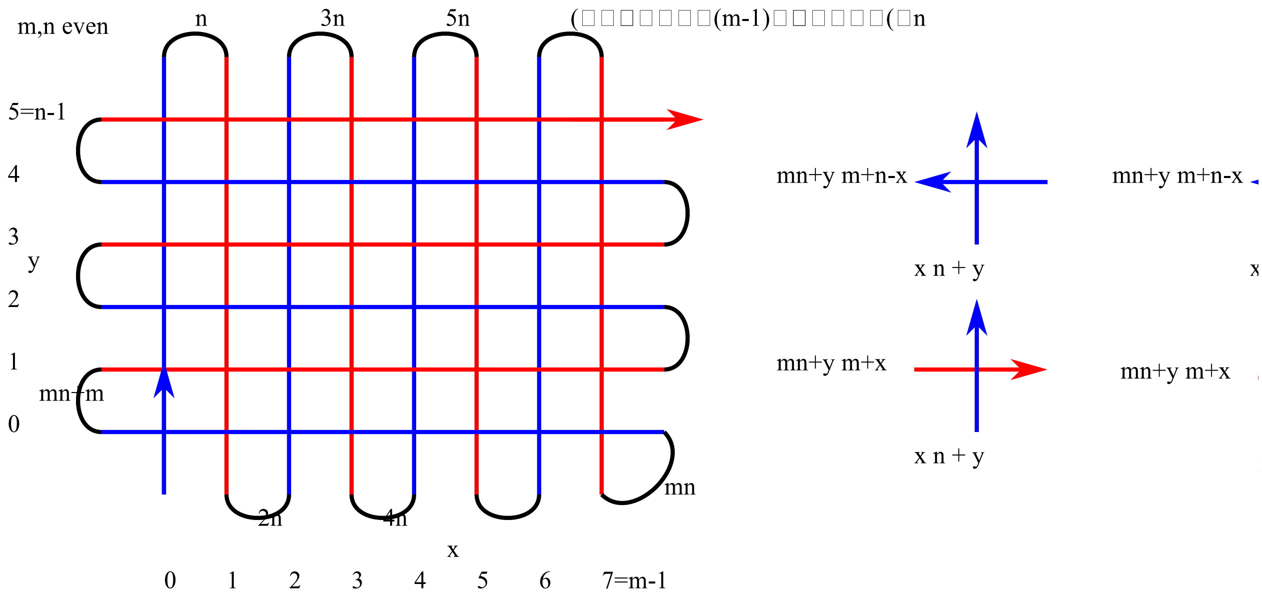
```
Γ1[φ_, k_] = -φ / 2 + φ g3kk;
```

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}} ))];
  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[Sum_{k1=1}^{n} Sum_{k2=1}^{n} θ[Cs[[k1]], Cs[[k2]]]];
  z += gEval[Sum_{k=1}^{n} R_1 @@ Cs[[k]]];
  z += gEval[Sum_{k=1}^{2 n} Γ_1[φ[[k]], k]];
  {Δ, (Δ /. T → T_1) (Δ /. T → T_2) (Δ /. T → T_3) 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[T_1^(m1=-Exponent[θ,T_1,Min]) T_2^(m2=-Exponent[θ,T_2,Min]) θ, {T_1, T_2}];
     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]];
```

A mat-shaped knot based on an m by n rectangular grid. The integers m,n should be even and the

crossings can be chosen arbitrarily.
Probably all knots can be brought into this form.




```
(*m,n, even, by default all xings positive. The
 list negxs flips the sign of those crossings.*)
Mat[m_, n_, negxs_ : {}] := Module[{pd}, pd = (Flatten@Table[
        Switch[
         {EvenQ[x], EvenQ[y]},
         {True, True},
         X[m n + y m + m - x - 1, n x + y + 1, m n + y m + m - x, n x + y],
         {True, False},
         X[n x + y, m n + y m + x + 1, n x + y + 1, m n + y m + x],
         {False, True},
         X[n x + n - y - 1, m n + y m + m - x, n x + n - y, m n + y m + m - x - 1],
         {False, False},
         X[m n + y m + x, n x + n - y, m n + y m + x + 1, n x + n - y - 1]
        ]
        , {x, 0, m - 1}, {y, 0, n - 1}]
     ) /. {X[a_, b_, c_, d_] :> X[a + 1, b + 1, c + 1, d + 1]};
  Do[pd[[i]] = (pd[[i]] /. {X[a_, b_, c_, d_] :> X[d, a, b, c]}), {i, negxs}];
  PD @@ pd
 ]
(*Randomly assign signs to all crossings*)
RandMat[m_, n_] := Mat[m, n, RandomSample[Range[m n], RandomInteger[{0, m n}]]]
```

In[∘]:=
```
(*The trefoil and the unknot*)
{Mat[2, 2],
 Mat[2, 2, {1}]}
θ /@ %
```

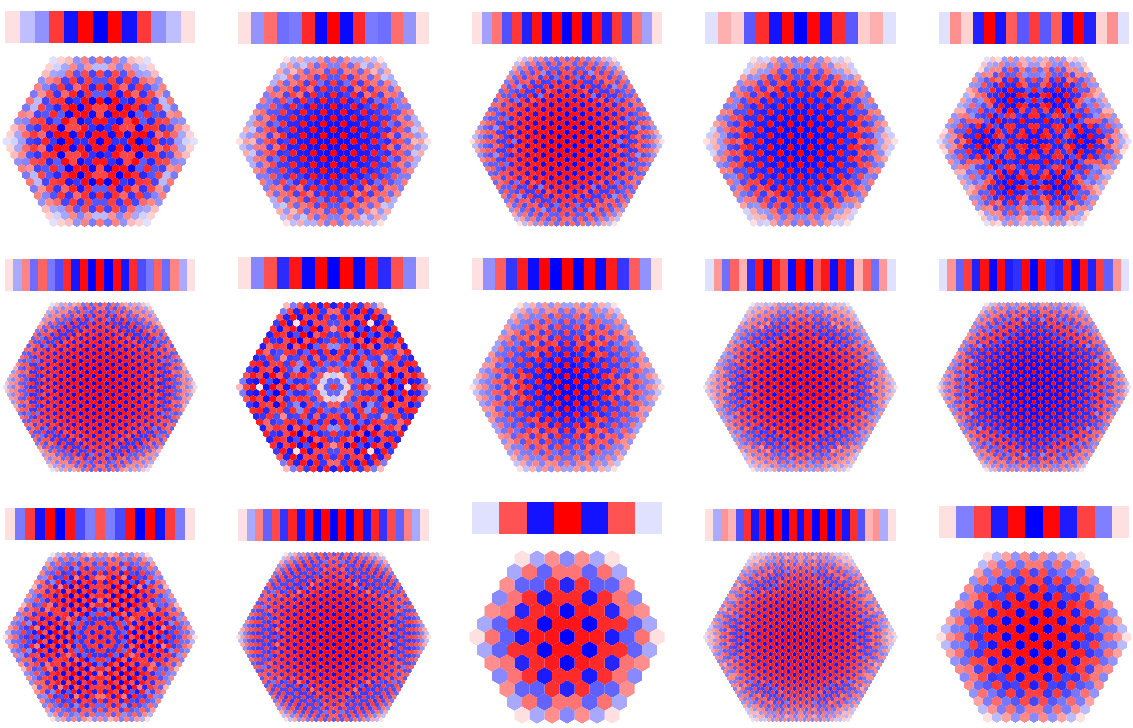
Out[∘]=
```
{PD[X[6, 2, 7, 1], X[2, 8, 3, 7], X[4, 6, 5, 5], X[8, 4, 9, 3]],
 PD[X[1, 6, 2, 7], X[2, 8, 3, 7], X[4, 6, 5, 5], X[8, 4, 9, 3]]}
```

Out[∘]=

$$\left\{\left\{\frac{1-T+T^2}{T},\ \frac{1-T_1+T_1^2-T_2-T_1^3\,T_2+T_2^2+T_1^4\,T_2^2-T_1\,T_2^3-T_1^4\,T_2^3+T_1^2\,T_2^4-T_1^3\,T_2^4+T_1^4\,T_2^4}{T_1^2\,T_2^2}\right\},\ \{1,0\}\right\}$$

In[∘]:=
```
GraphicsGrid[Table[PolyPlot@θ@RandMat[6, 8], {i, 3}, {j, 5}]]
```

Out[∘]=



In[ ]:=

```
GraphicsGrid[Table[PolyPlot@θ@RandMat[10, 8], {i, 4}, {j, 7}]]
```

Out[ ]=