0-co one step middle c program.

The format is E = {Const,Linear,Quad} which means $O(\text{Const}\, e^{\text{Linear+Quad}})$ we take b=Log[t]

```
(*Stitching operation*)
m[i_, j_, k_][E_] := Block[
  {α, β, γ, Γ, δ, C, L, Q, QRest, μ },
  C = E[[1]]; (*The (inverse of the) constant in front, aka ω*)
  L = E[[2]]; (*The linear term of the exponential, unscaled*)
  Q = E[[3]]; (*The quadratic term (unscaled).*)

  α  = D[Q, w_i] /. a_j → 0;
  β  = D[Q, a_j] /. w_i → 0;
  γ  = D[L, c_i]; (*includes b*)
  Γ  = D[L, c_j];
  δ  = D[Q, a_j, w_i];
  QRest = Q /. {a_j → 0, w_i → 0};
  μ  = 1 - (1 - t_k) δ;

  {μ C, L, QRest + μ^-1 ((1 - t_k) α β + Exp[Γ] α w_i + Exp[γ] β a_j + Exp[γ + Γ] δ a_j w_i )} /.
    {c_i|j → c_k, a_i|j → a_k, w_i|j → w_k, t_i|j → t_k} // Together
 ]
```

```
EE = {2 - t_1, Log[t_2] c_1 + 2 Log[t_1] c_2, 2 r a_3 w_1 + 3 s a_2 w_1 + 2 r a_2 w_3};
```

```
(*Meta-associativity*)
(EE // m[1, 2, 1] // m[1, 3, 1] // Expand) -
 (EE // m[2, 3, 2] // m[1, 2, 1] // Expand)
```

```
{0, 0, 0}
```

```
(*Disjoint union of two tangles, multiplies the constant terms, adds the exponents*)
DisjUnion[L_] := {Times @@ L[[All, 1]], Plus @@ L[[All, 2]], Plus @@ L[[All, 3]]}
```

```
(*R-matrix*)
Rp[i_, j_] := {1, Log[t_i] c_j, a_i w_j}
Rm[i_, j_] := {1, -Log[t_i] c_j, -t_i^-1 a_i w_j}
```

```
(*Cuaps*)
nr[i_] := {t_i^(1/2), 0, 0}
nl[i_] := {1, 0, 0}
ur[i_] := {t_i^(-1/2), 0, 0}
ul[i_] := {1, 0, 0}
```

```
(*Reidemeister 2*)
DisjUnion@{Rp[1, 2], Rm[3, 4]} // m[1, 3, x] // m[2, 4, y]
```

{1, 0, 0}

```
(*Reidemeister 3*)
(DisjUnion@{Rp[1, 2], Rp[3, 4], Rp[5, 6]} // m[1, 3, x] // m[2, 5, y] // m[4, 6, z]) -
 (DisjUnion@{Rp[1, 2], Rp[3, 4], Rp[5, 6]} // m[3, 5, x] // m[1, 6, y] // m[2, 4, z])
```

{0, 0, 0}

```
(*Kinks*)
```

$\text{Kp}[i\_] := \left\{\sqrt{t_i}, \text{Log}[t_i]\, c_i, a_i w_i\right\}$

$\text{Km}[i\_] := \left\{\frac{1}{\sqrt{t_i}}, -\text{Log}[t_i]\, c_i, -\frac{a_i w_i}{t_i}\right\}$

```
DisjUnion[{Rp[1, 4], nr[2], ul[3]}] // m[1, 2, 1] // m[1, 3, 1] // m[1, 4, 1]
DisjUnion[{Rm[4, 1], nr[2], ul[3]}] // m[1, 2, 1] // m[1, 3, 1] // m[1, 4, 1];
DisjUnion[{Rp[4, 1], nl[2], ur[3]}] // m[1, 2, 1] // m[1, 3, 1] // m[1, 4, 1];
DisjUnion[{Rm[1, 4], nl[2], ur[3]}] // m[1, 2, 1] // m[1, 3, 1] // m[1, 4, 1];
```

$\left\{\sqrt{t_1}, \text{Log}[t_1]\, c_1, a_1 w_1\right\}$

```
(*3_1 0-Framed*)
Z = DisjUnion@{Rp[5, 1], Rp[2, 6], Rp[7, 3], ur[4], Km[8], Km[9], Km[10]};
For[i = 2, i ≤ 10, i++, Z = Z // m[1, i, 1]];
Z // Expand
```

$\left\{-1 + \frac{1}{t_1} + t_1, 0, 0\right\}$

```
(*4_1 0-Framed*)
Z = DisjUnion@{Rm[8, 1], Rp[5, 9], Rm[2, 6], Rp[10, 3], nr[7], ur[4]};
For[i = 2, i ≤ 10, i++, Z = Z // m[1, i, 1]];
Z // Expand
```

$\left\{3 - \frac{1}{t_1} - t_1, 0, 0\right\}$

```
(*5_1 0-framed*)
Z = DisjUnion@{Rp[7, 1], Rp[2, 8], Rp[9, 3],
    Rp[4, 10], Rp[11, 5], ur[6], Km[12], Km[13], Km[14], Km[15], Km[16]};
For[i = 2, i ≤ 16, i++, Z = Z // m[1, i, 1]];
Z // Expand
```

$\left\{1 + \frac{1}{t_1^2} - \frac{1}{t_1} - t_1 + t_1^2, 0, 0\right\}$

```
ReverseStrand[i_][E_] :=
 Block[(*Strand reversal means taking the q-antipode S at strand i. This changes
    the order acw to the opposite order wca so we need to commute it back.*)
  {α, β, γ, Γ, δ, C, L, Q, QRest, μ },
  (*First apply the antipode to the terms*)
  C = E[[1]] /. {a_i → -t_i^-1 a_i, t_i → t_i^-1, c_i → -c_i, w_i → -w_i};
  L = E[[2]] /. {a_i → -t_i^-1 a_i, t_i → t_i^-1, c_i → -c_i, w_i → -w_i};
  Q = E[[3]] /. {a_i → -t_i^-1 a_i, t_i → t_i^-1, c_i → -c_i, w_i → -w_i};

  α  = D[Q, w_i] /. a_i → 0;
  β  = D[Q, a_i] /. w_i → 0;
  γ  = D[L, c_i]; (*includes b*)
  δ  = D[Q, a_i, w_i];
  QRest = Q /. {a_i → 0, w_i → 0};
  β = β Exp[γ]; δ = δ Exp[γ]; (*first commute a and c in wca*)
  μ  = 1 - (1 - t_i) δ;
  (*Print[α,β,γ,δ];*)
  {μ C, L, QRest + μ^-1 ((1 - t_i) α β  + Exp[γ] α w_i + β a_i + Exp[γ] δ a_i w_i )}(*//Together*)
 ]
```

```
DoubleStrand[i_, j_, k_][E_] := (*Just the q-coproduct*)
 E /. {a_i → t_k a_j + a_k, c_i → c_j + c_k, w_i → w_j + w_k, t_i → t_j t_k}
```

```
Simplify[Log[a b] - Log[a] - Log[b], Assumptions → {a > 0, b > 0}]
```

```
0
```

```
(*Double positive crossing*)
Simplify[(DoubleStrand[1, x, y]@Rp[1, 2] // Expand) -
   (DisjUnion[{Rp[x, 2], Rp[y, z]}] // m[2, z, 2]), Assumptions → {t_x > 0}]
(DoubleStrand[2, x, y]@Rp[1, 2] // Expand) -
 (DisjUnion[{Rp[1, y], Rp[z, x]}] // m[1, z, 1])
```

```
{0, 0, 0}
```

```
{0, 0, 0}
```

```
(*Reverse positive crossing*)
Simplify[ReverseStrand[1]@Rp[1, 2] -
  Rm[1, 2], Assumptions → {t_1 > 0}]
ReverseStrand[2]@Rp[1, 2] -
  Rm[1, 2] /. b_i_ → Log[t_i]
```

```
{0, 0, 0}
```

```
{0, 0, 0}
```

```
(*Double negative Kink*)
Simplify[(DoubleStrand[1, 2, 3]@Km[1] // Expand) -
  (DisjUnion@{Rm[2, 8], Rm[4, 9], Rm[3, 6], Rm[5, 7]} // m[2, 4, 2] // m[2, 6, 2] // m[2, 8,
      2] // m[3, 5, 3] // m[3, 7, 3] // m[3, 9, 3]) // Together, Assumptions → {t2 > 0}]
```

$\left\{-1+\frac{1}{\sqrt{t_2 t_3}}, 0, 0\right\}$

```
Kp[1]
Km[1]
```

$\left\{\sqrt{t_1}, b_1 c_1, a_1 w_1\right\}$

$\left\{\frac{1}{\sqrt{t_1}}, -b_1 c_1, -\frac{a_1 w_1}{t_1}\right\}$

```
Rp[1, 2] // m[1, 2, 1]
```

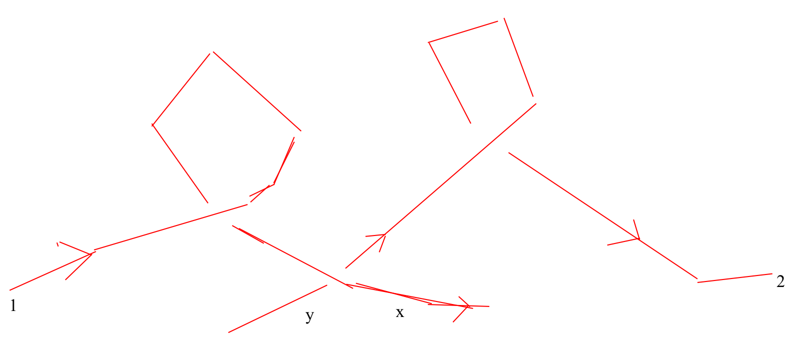
$\{1, \text{Log}[t_1]\, c_1, a_1 w_1\}$

```
(*Reverse Kinks*)
Simplify[ReverseStrand[1]@Kp[1] - Kp[1], Assumptions → {t1 > 0}]
Simplify[ReverseStrand[1]@Km[1] - Km[1], Assumptions → {t1 > 0}]
```

$\{0, 0, 0\}$

$\{0, 0, 0\}$

Now let's build a trefoil by strand doubling and then reversing the left hand strand, starting with this tangle:



```
(*It's just two negative curls crossing over in a positive crossing.*)
Start = DisjUnion@{Km[1], Km[2], Rp[x, y]} // m[1, x, 1] // m[y, 2, 2]
```

$\left\{\frac{1}{\sqrt{t_1}\,\sqrt{t_2}},\ -\text{Log}[t_1]\,c_1+\text{Log}[t_1]\,c_2-\text{Log}[t_2]\,c_2,\ \frac{1}{t_1 t_2}\left(-a_1 t_2 w_1-a_2 t_1^2 w_2+a_1 t_1 t_2 w_2\right)\right\}$

```
Simplify[(DoubleStrand[2, -2, 2]@DoubleStrand[1, -1, 1]@Start // Expand // Together) -
  ((DisjUnion@{Rp[2, 5], Rp[1, 6], Rp[3, 8], Rp[4, 7], DoubleStrand[1, x, y]@Km[1],
            DoubleStrand[1, z, j]@Km[1]} // m[5, 6, 5] // m[7, 8, 7] // m[1, 3, 1] //
          m[2, 4, 2] // m[y, 1, y] // m[x, 2, x] // m[5, j, j] // m[7, z, z]) /.
    {j → 2, z → -2, y → 1, x → -1}), Assumptions → {t1 > 0, t2 >
    0}]
```

{0, 0, 0}

```
ReverseStrand[-2]@
    ReverseStrand[-1]@DoubleStrand[2, -2, 2]@DoubleStrand[1, -1, 1]@Start //
  m[1, -2, 1] // m[1, -1, 1] // m[1, 2, 1]
```

$\{1-t_1+t_1^2, 0, 0\}$

```
ReverseStrand[-2]@ReverseStrand[-1]@DoubleStrand[2, -2, 2]@DoubleStrand[1, -1, 1]@
      (DisjUnion@{Km[1], Km[2], Rp[x, y]} // m[1, x, 1] // m[y, 2, 2]) //
  m[1, -2, 1] // m[1, -1, 1] // m[1, 2, 1]
(*We could also do a negative crossing instead of a positive one
 and the result would still be the trefoil.*)
ReverseStrand[-2]@ReverseStrand[-1]@DoubleStrand[2, -2, 2]@DoubleStrand[1, -1, 1]@
      (DisjUnion@{Km[1], Km[2], Rm[x, y]} // m[1, y, 1] // m[x, 2, 2]) //
  m[1, -2, 1] // m[1, -1, 1] // m[1, 2, 1]
```

$\{1-t_1+t_1^2, 0, 0\}$

$\{1-t_1+t_1^2, 0, 0\}$