
code

data (pre-computed)

```
In[ ]:= type = "E6";
(*Define basis elements as symbols*)
basisElements = {V0, V1, V2, V3, V4, V5, V6, V7, V8};
(*Define multiplication rules*)
multiplicationRules = {
  (V0 * V0) => V0,
  (V0 * V1) => V1,
  (V0 * V2) => V2,
  (V0 * V3) => V3,
  (V0 * V4) => V4,
  (V0 * V5) => V5,
  (V0 * V6) => V6,
  (V0 * V7) => V7,
  (V0 * V8) => V8,
  (V1 * V1) => V3 + V6 + V2,
  (V1 * V2) => V4 + V5 + V0,
  (V1 * V3) => V4,
  (V1 * V4) => V8 + V7 + V1,
  (V1 * V5) => V7 + V1,
  (V1 * V6) => V4 + V5,
  (V1 * V7) => V6 + V2,
  (V1 * V8) => V2,
  (V2 * V2) => V8 + V7 + V1,
  (V2 * V3) => V1,
  (V2 * V4) => V3 + V6 + V2,
  (V2 * V5) => V6 + V2,
  (V2 * V6) => V7 + V1,
  (V2 * V7) => V4 + V5,
  (V2 * V8) => V4,
  (V3 * V3) => V8,
  (V3 * V4) => V2,
  (V3 * V5) => V6,
  (V3 * V6) => V7,
  (V3 * V7) => V5,
  (V3 * V8) => V0,
  (V4 * V4) => V4 + V5 + V0,
  (V4 * V5) => V4 + V5,
  (V4 * V6) => V6 + V2,
  (V4 * V7) => V7 + V1,
  (V4 * V8) => V1,
  (V5 * V5) => V4 + V0,
  (V5 * V6) => V3 + V2,
  (V5 * V7) => V8 + V1,
  (V5 * V8) => V7,
  (V6 * V6) => V8 + V1,
```

```

(V6 * V7) => V4 + V0,
(V6 * V8) => V5,
(V7 * V7) => V3 + V2,
(V7 * V8) => V6,
(V8 * V8) => V3};
KRvariableToVerlindeImage = {
  x1 -> ringElement[V1],
  x2 -> ringElement[V3],
  x3 -> ringElement[V2 + V6],
  x4 -> ringElement[V8],
  x5 -> ringElement[V0 + V4 + 2 V5],
  x6 -> ringElement[V0],
  x7 -> ringElement[V0 + V5],
  x8 -> ringElement[V0],
  x9 -> ringElement[V1 + V7],
  x10 -> ringElement[V3],
  x11 -> ringElement[V2],
  x12 -> ringElement[V8],
  x26 -> ringElement[V1],
  x27 -> ringElement[V2],
  x28 -> ringElement[V2 + V6],
  x30 -> ringElement[V0 + V5],
  x31 -> ringElement[V0 + V4 + 2 V5],
  x46 -> ringElement[V1 + V7]
};

```

import exchange relations

```

In[ ]:= extractxindex[var_] := Module[{numberStr}, Which[
  ListQ[var], extractxindex /@ var,
  True, numberStr = StringDrop[SymbolName[var], 1];
  ToExpression[numberStr]]]
xSort[vars_List] := SortBy[vars, extractxindex]
swapIfNotOrdered[triple_List] := Module[{second = triple[[2]], third = triple[[3]]},
  If[OrderedQ[{second, third}], ReplacePart[triple, {2 -> third, 3 -> second}], triple]]
filePath = FileNameJoin[{NotebookDirectory[], "exchange_relations_" <> type <> ".txt"}];
fileLines = ReadList[filePath, String];
convertedLines = DeleteDuplicates[StringReplace[fileLines, {"[" -> "{", "]" -> "}"}]];
exchangerelations = ToExpression /@ convertedLines;
exchangerelations =
  Map[{xSort[#[[1]]], xSort[#[[2]]], xSort[#[[3]]]} &, exchangerelations];
exchangerelations = SortBy[exchangerelations, extractxindex[#[[1]]] &];
exchangerelations = Map[swapIfNotOrdered, exchangerelations];
exchangerelations = DeleteDuplicates[exchangerelations];
(* display sample exchange relations *)
exchangerelations[[1 ;; 10]]
(* display cluster variables *)
clustervariables = Flatten[exchangerelations] // DeleteDuplicates // xSort
Out[ ]:= {{x1, x17}, {x2, x11}, {x16}}, {{x1, x18}, {x2, x15}, {x16, x23}},
{{x1, x21}, {x2, x20}, {x16, x24}}, {{x1, x22}, {x2, x36}, {x24}},
{{x1, x26}, {x28}, {x2}}, {{x1, x34}, {x2, x3}, {x4}},
{{x1, x35}, {x2, x7}, {x23}}, {{x1, x37}, {x2, x7, x11}, {x15}},
{{x1, x38}, {x2, x7, x9}, {x39}}, {{x1, x40}, {x2, x9}, {x25}}

```

```
Out[ ]:= {x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, x14, x15, x16, x17,
          x18, x19, x20, x21, x22, x23, x24, x25, x26, x27, x28, x29, x30, x31, x32, x33,
          x34, x35, x36, x37, x38, x39, x40, x41, x42, x43, x44, x45, x46, x47, x48}
```

code for Verlinde ring

```
In[ ]:= rankofVerlinde = Length[basisElements];

simplifyRingElement[expr_] := Module[{result},
  result = Total[Coefficient[expr, #] * # & /@ basisElements];
  result]

(*Define a function for addition of ring elements*)
ringAdd[element1_, element2_] := element1 + element2
(*Define a function for multiplication of ring elements*)
ringMultiply[element1_, element2_] :=
  Expand[Expand[(element1 element2)] /. multiplicationRules]

(*Define the ringElement class*)
ringElement /: Plus[ringElement[e1_], ringElement[e2_]] :=
  ringElement[simplifyRingElement[ringAdd[e1, e2]]];
ringElement /: Times[ringElement[e1_], ringElement[e2_]] :=
  ringElement[simplifyRingElement[ringMultiply[e1, e2]]];
ringElement /: Times[scalar_, ringElement[e_]] /:
  FreeQ[scalar, Alternatives @@ basisElements] :=
  ringElement[simplifyRingElement[scalar * e]];
(*Subtraction rule using addition and scalar multiplication*)
ringElement /: Subtract[ringElement[e1_], ringElement[e2_]] :=
  ringElement[simplifyRingElement[ringAdd[e1, -1 * e2]]];

unknownindex = Complement[Range[1, Length[clustervariables]],
  Map[extractxindex, First /@ KRvariableToVerlindeImage]];

variableToVerlindeImage = Union[
  KRvariableToVerlindeImage,
  Table[ToExpression["x" <> ToString[index]] ->
    ringElement[Sum[c[index, k] * Evaluate[Symbol["V" <> ToString[k]]],
      {k, 0, rankofVerlinde - 1}]], {index, unknownindex}]];
unknowns = Flatten@Table[Table[c[index, k], {k, 0, rankofVerlinde - 1}],
  {index, unknownindex}];

(* convert expression in cluster variables to Verlinde element *)
convertToRingElements[expr_] := expr /. variableToVerlindeImage
(*Function to transform {{x1,x17},x2 x3+x4} to x1*x17-(x2*x3+x4)*)
exchangeVerlinde[{{a_, b_}, X_, Y_}] :=
  convertToRingElements[a] * convertToRingElements[b] -
  (Times @@ convertToRingElements[X]) - (Times @@ convertToRingElements[Y])
exponents[expr_] := Module[{terms, totalDegree},
  terms = If[Head[Expand[expr]] == Plus, List @@ Expand[expr], {Expand[expr]}];
  totalDegree = Map[Total[Cases[Factor[#], c[_] ^ n_ >= n, {0, 1}]] &, terms];
  totalDegree
]
extractCoefficients[ringElement[expr_]] := Coefficient[expr, #] & /@ basisElements
```

exemple of exchange relations and corresponding equations for Verlinde coefficients

```
In[ ]:= unknowns // Length
```

```
Out[ ]:= 270
```

```
In[ ]:= Map[# == 0 &, extractCoefficients[exchangeVerlinde[{{x1, x34}, {x2, x3}, {x4}}]]] //  
TableForm
```

```
Out[ ]//TableForm=
```

```
c[34, 2] == 0  
-1 + c[34, 0] + c[34, 4] + c[34, 5] == 0  
c[34, 1] + c[34, 7] + c[34, 8] == 0  
c[34, 1] == 0  
c[34, 2] + c[34, 3] + c[34, 6] == 0  
c[34, 2] + c[34, 6] == 0  
c[34, 1] + c[34, 7] == 0  
-1 + c[34, 4] + c[34, 5] == 0  
-1 + c[34, 4] == 0
```

```
In[ ]:= Map[# == 0 &, extractCoefficients[exchangeVerlinde[{{x1, x17}, {x2, x11}, {x16}}]]] //  
TableForm
```

```
Out[ ]//TableForm=
```

```
-c[16, 0] + c[17, 2] == 0  
-1 - c[16, 1] + c[17, 0] + c[17, 4] + c[17, 5] == 0  
-c[16, 2] + c[17, 1] + c[17, 7] + c[17, 8] == 0  
-c[16, 3] + c[17, 1] == 0  
-c[16, 4] + c[17, 2] + c[17, 3] + c[17, 6] == 0  
-c[16, 5] + c[17, 2] + c[17, 6] == 0  
-c[16, 6] + c[17, 1] + c[17, 7] == 0  
-c[16, 7] + c[17, 4] + c[17, 5] == 0  
-c[16, 8] + c[17, 4] == 0
```

Solving equations for Verlinde coefficients

set of all equations among coefficients $\{c[j,k]\}$

```
In[ ]:= coeffList = DeleteDuplicates@
  Flatten[Table[(extractCoefficients@exchangeVerlinde[exchangerelations[[index]]],
    {index, 1, Length[exchangerelations]}]];
coeffList = SortBy[coeffList, Max[exponents[#]] &];
coeffList[[1 ;; 50]]

Out[ ]:= {0, -c[13, 0], -c[13, 1], 1 - c[13, 2], -c[13, 3], -c[13, 4],
  -c[13, 5], c[13, 4] + c[13, 5], c[13, 0] + 2 c[13, 4] + c[13, 5],
  c[13, 0] + c[13, 4] + 2 c[13, 5], c[13, 0] + 2 c[13, 4] + 2 c[13, 5],
  2 c[13, 0] + 3 c[13, 4] + 2 c[13, 5], c[13, 0] + 4 c[13, 4] + 3 c[13, 5],
  -c[13, 6], -1 + c[13, 2] + c[13, 6], -2 + 2 c[13, 2] + c[13, 3] + c[13, 6],
  -1 + c[13, 2] + c[13, 3] + 2 c[13, 6], -2 + 2 c[13, 2] + c[13, 3] + 2 c[13, 6],
  -3 + 3 c[13, 2] + 2 c[13, 3] + 2 c[13, 6], -4 + 4 c[13, 2] + c[13, 3] + 3 c[13, 6],
  -c[13, 7], c[13, 1] + c[13, 7], -c[13, 8], 2 c[13, 1] + c[13, 7] + c[13, 8],
  c[13, 1] + 2 c[13, 7] + c[13, 8], 2 c[13, 1] + 2 c[13, 7] + c[13, 8],
  4 c[13, 1] + 3 c[13, 7] + c[13, 8], 3 c[13, 1] + 2 c[13, 7] + 2 c[13, 8], -c[14, 0],
  -c[14, 1], c[14, 1], -c[14, 2], c[14, 2], -c[14, 3], 1 - c[14, 4], -1 + c[14, 4],
  -c[14, 5], -1 + c[14, 4] + c[14, 5], -1 + c[14, 0] + c[14, 4] + c[14, 5],
  -2 + c[14, 0] + 2 c[14, 4] + c[14, 5], -2 + c[14, 0] + 2 c[14, 4] + 2 c[14, 5],
  -c[14, 6], c[14, 2] + c[14, 6], c[14, 2] + c[14, 3] + c[14, 6],
  2 c[14, 2] + c[14, 3] + c[14, 6], 2 c[14, 2] + c[14, 3] + 2 c[14, 6],
  -c[14, 7], c[14, 1] + c[14, 7], -c[14, 8], c[14, 1] + c[14, 7] + c[14, 8]}
```

first set of linear equations

```
In[ ]:= Block[{knownsols, newSols, numEquations, equations, solution, currentIndex, batchSize},
  knownsols = {};
  Print["number of known variables before:", Length@knownsols];
  equations = coeffList;
  equations = equations /. knownsols;
  equations = DeleteDuplicates@ Select[equations, Max[exponents[#]] == 1 &];
  Print["number of equations used:", Length[equations]];
  equations = Map[# == 0 &, equations];
  equations = Join[equations, knownsols /. Rule -> Equal];
  solution = Solve[equations, unknowns];
  If[Length[solution] == 1,
    newSols = Select[solution[[1]], NumericQ[#[[2]]] &];
    knownsols = Union[knownsols, newSols];
    Print["number of known variables after: ", {Length[#], #} &@knownsols];
  ]
]
```

number of known variables before:0

number of equations used:1146

number of known variables after:

```
{261, {c[13, 0] → 0, c[13, 1] → 0, c[13, 2] → 1, c[13, 3] → 0, c[13, 4] → 0, c[13, 5] → 0,
c[13, 6] → 0, c[13, 7] → 0, c[13, 8] → 0, c[14, 0] → 0, c[14, 1] → 0, c[14, 2] → 0, c[14, 3] → 0,
c[14, 4] → 1, c[14, 5] → 0, c[14, 6] → 0, c[14, 7] → 0, c[14, 8] → 0, c[15, 0] → 0, c[15, 1] → 0,
c[15, 2] → 0, c[15, 3] → 0, c[15, 4] → 0, c[15, 5] → 0, c[15, 6] → 0, c[15, 7] → 1, c[15, 8] → 1,
c[16, 0] → 0, c[16, 1] → 0, c[16, 2] → 0, c[16, 3] → 0, c[16, 4] → 0, c[16, 5] → 0, c[16, 6] → 0,
c[16, 7] → 1, c[16, 8] → 1, c[17, 0] → 0, c[17, 1] → 0, c[17, 2] → 0, c[17, 3] → 0, c[17, 4] → 1,
c[17, 5] → 0, c[17, 6] → 0, c[17, 7] → 0, c[17, 8] → 0, c[18, 0] → 0, c[18, 1] → 0, c[18, 2] → 1,
c[18, 3] → 0, c[18, 4] → 0, c[18, 5] → 0, c[18, 6] → 1, c[18, 7] → 0, c[18, 8] → 0, c[19, 0] → 0,
c[19, 1] → 0, c[19, 2] → 1, c[19, 3] → 0, c[19, 4] → 0, c[19, 5] → 0, c[19, 6] → 0, c[19, 7] → 0,
c[19, 8] → 0, c[20, 0] → 0, c[20, 1] → 0, c[20, 2] → 1, c[20, 3] → 0, c[20, 4] → 0, c[20, 5] → 0,
c[20, 6] → 1, c[20, 7] → 0, c[20, 8] → 0, c[21, 0] → 1, c[21, 1] → 0, c[21, 2] → 0, c[21, 3] → 0,
c[21, 4] → 1, c[21, 5] → 2, c[21, 6] → 0, c[21, 7] → 0, c[21, 8] → 0, c[22, 0] → 0, c[22, 1] → 0,
c[22, 2] → 0, c[22, 3] → 1, c[22, 4] → 0, c[22, 5] → 0, c[22, 6] → 1, c[22, 7] → 0, c[22, 8] → 0,
c[23, 0] → 0, c[23, 1] → 0, c[23, 2] → 1, c[23, 3] → 0, c[23, 4] → 0, c[23, 5] → 0, c[23, 6] → 0,
c[23, 7] → 0, c[23, 8] → 0, c[24, 0] → 0, c[24, 1] → 0, c[24, 2] → 0, c[24, 3] → 0, c[24, 4] → 1,
c[24, 5] → 1, c[24, 6] → 0, c[24, 7] → 0, c[24, 8] → 0, c[25, 0] → 0, c[25, 1] → 0, c[25, 2] → 0,
c[25, 3] → 0, c[25, 4] → 1, c[25, 5] → 0, c[25, 6] → 0, c[25, 7] → 0, c[25, 8] → 0, c[29, 0] → 0,
c[29, 1] → 1, c[29, 2] → 0, c[29, 3] → 0, c[29, 4] → 0, c[29, 5] → 0, c[29, 6] → 0, c[29, 7] → 0,
c[29, 8] → 0, c[32, 0] → 0, c[32, 1] → 0, c[32, 2] → 1, c[32, 3] → 0, c[32, 4] → 0, c[32, 5] → 0,
c[32, 6] → 1, c[32, 7] → 0, c[32, 8] → 0, c[33, 0] → 1, c[33, 1] → 0, c[33, 2] → 0, c[33, 3] → 0,
c[33, 4] → 0, c[33, 5] → 1, c[33, 6] → 0, c[33, 7] → 0, c[33, 8] → 0, c[34, 0] → 0, c[34, 1] → 0,
c[34, 2] → 0, c[34, 3] → 0, c[34, 4] → 1, c[34, 5] → 0, c[34, 6] → 0, c[34, 7] → 0, c[34, 8] → 0,
c[35, 0] → 0, c[35, 1] → 1, c[35, 2] → 0, c[35, 3] → 0, c[35, 4] → 0, c[35, 5] → 0, c[35, 6] → 0,
c[35, 7] → 0, c[35, 8] → 0, c[36, 0] → 0, c[36, 1] → 1, c[36, 2] → 0, c[36, 3] → 0, c[36, 4] → 0,
c[36, 5] → 0, c[36, 6] → 0, c[36, 7] → 0, c[36, 8] → 0, c[37, 0] → 0, c[37, 1] → 0, c[37, 2] → 0,
c[37, 3] → 0, c[37, 4] → 1, c[37, 5] → 1, c[37, 6] → 0, c[37, 7] → 0, c[37, 8] → 0, c[38, 0] → 0,
c[38, 1] → 0, c[38, 2] → 1, c[38, 3] → 1, c[38, 4] → 0, c[38, 5] → 0, c[38, 6] → 2, c[38, 7] → 0,
c[38, 8] → 0, c[39, 0] → 0, c[39, 1] → 0, c[39, 2] → 0, c[39, 3] → 0, c[39, 4] → 1, c[39, 5] → 1,
c[39, 6] → 0, c[39, 7] → 0, c[39, 8] → 0, c[40, 0] → 0, c[40, 1] → 0, c[40, 2] → 0, c[40, 3] → 1,
c[40, 4] → 0, c[40, 5] → 0, c[40, 6] → 1, c[40, 7] → 0, c[40, 8] → 0, c[41, 0] → 0, c[41, 1] → 0,
c[41, 2] → 0, c[41, 3] → 0, c[41, 4] → 1, c[41, 5] → 1, c[41, 6] → 0, c[41, 7] → 0, c[41, 8] → 0,
c[42, 0] → 0, c[42, 1] → 1, c[42, 2] → 0, c[42, 3] → 0, c[42, 4] → 0, c[42, 5] → 0, c[42, 6] → 0,
c[42, 7] → 2, c[42, 8] → 1, c[43, 0] → 0, c[43, 1] → 1, c[43, 2] → 0, c[43, 3] → 0, c[43, 4] → 0,
c[43, 5] → 0, c[43, 6] → 0, c[43, 7] → 1, c[43, 8] → 0, c[44, 0] → 1, c[44, 1] → 0, c[44, 2] → 0,
c[44, 3] → 0, c[44, 4] → 1, c[44, 5] → 2, c[44, 6] → 0, c[44, 7] → 0, c[44, 8] → 0,
c[45, 0] → 0, c[45, 1] → 1, c[45, 2] → 0, c[45, 3] → 0, c[45, 4] → 0, c[45, 5] → 0,
c[45, 6] → 0, c[45, 7] → 1, c[45, 8] → 0, c[47, 0] → 0, c[47, 1] → 1, c[47, 2] → 0,
c[47, 3] → 0, c[47, 4] → 0, c[47, 5] → 0, c[47, 6] → 0, c[47, 7] → 1, c[47, 8] → 0}}
```

second set of linear equations

```
In[ ]:= Block[{knownsols, newSols, numEquations,
equations, solution, currentIndex, batchSize},
knownsols = {c[13, 0] → 0, c[13, 1] → 0, c[13, 2] → 1, c[13, 3] → 0, c[13, 4] → 0,
c[13, 5] → 0, c[13, 6] → 0, c[13, 7] → 0, c[13, 8] → 0, c[14, 0] → 0,
c[14, 1] → 0, c[14, 2] → 0, c[14, 3] → 0, c[14, 4] → 1, c[14, 5] → 0, c[14, 6] → 0,
c[14, 7] → 0, c[14, 8] → 0, c[15, 0] → 0, c[15, 1] → 0, c[15, 2] → 0, c[15, 3] → 0,
c[15, 4] → 0, c[15, 5] → 0, c[15, 6] → 0, c[15, 7] → 1, c[15, 8] → 1, c[16, 0] → 0,
c[16, 1] → 0, c[16, 2] → 0, c[16, 3] → 0, c[16, 4] → 0, c[16, 5] → 0, c[16, 6] → 0,
c[16, 7] → 1, c[16, 8] → 1, c[17, 0] → 0, c[17, 1] → 0, c[17, 2] → 0, c[17, 3] → 0,
c[17, 4] → 1, c[17, 5] → 0, c[17, 6] → 0, c[17, 7] → 0, c[17, 8] → 0, c[18, 0] → 0,
```

```

c[18, 1] → 0, c[18, 2] → 1, c[18, 3] → 0, c[18, 4] → 0, c[18, 5] → 0, c[18, 6] → 1,
c[18, 7] → 0, c[18, 8] → 0, c[19, 0] → 0, c[19, 1] → 0, c[19, 2] → 1, c[19, 3] → 0,
c[19, 4] → 0, c[19, 5] → 0, c[19, 6] → 0, c[19, 7] → 0, c[19, 8] → 0, c[20, 0] → 0,
c[20, 1] → 0, c[20, 2] → 1, c[20, 3] → 0, c[20, 4] → 0, c[20, 5] → 0, c[20, 6] → 1,
c[20, 7] → 0, c[20, 8] → 0, c[21, 0] → 1, c[21, 1] → 0, c[21, 2] → 0, c[21, 3] → 0,
c[21, 4] → 1, c[21, 5] → 2, c[21, 6] → 0, c[21, 7] → 0, c[21, 8] → 0, c[22, 0] → 0,
c[22, 1] → 0, c[22, 2] → 0, c[22, 3] → 1, c[22, 4] → 0, c[22, 5] → 0, c[22, 6] → 1,
c[22, 7] → 0, c[22, 8] → 0, c[23, 0] → 0, c[23, 1] → 0, c[23, 2] → 1, c[23, 3] → 0,
c[23, 4] → 0, c[23, 5] → 0, c[23, 6] → 0, c[23, 7] → 0, c[23, 8] → 0, c[24, 0] → 0,
c[24, 1] → 0, c[24, 2] → 0, c[24, 3] → 0, c[24, 4] → 1, c[24, 5] → 1, c[24, 6] → 0,
c[24, 7] → 0, c[24, 8] → 0, c[25, 0] → 0, c[25, 1] → 0, c[25, 2] → 0, c[25, 3] → 0,
c[25, 4] → 1, c[25, 5] → 0, c[25, 6] → 0, c[25, 7] → 0, c[25, 8] → 0, c[29, 0] → 0,
c[29, 1] → 1, c[29, 2] → 0, c[29, 3] → 0, c[29, 4] → 0, c[29, 5] → 0, c[29, 6] → 0,
c[29, 7] → 0, c[29, 8] → 0, c[32, 0] → 0, c[32, 1] → 0, c[32, 2] → 1, c[32, 3] → 0,
c[32, 4] → 0, c[32, 5] → 0, c[32, 6] → 1, c[32, 7] → 0, c[32, 8] → 0, c[33, 0] → 1,
c[33, 1] → 0, c[33, 2] → 0, c[33, 3] → 0, c[33, 4] → 0, c[33, 5] → 1, c[33, 6] → 0,
c[33, 7] → 0, c[33, 8] → 0, c[34, 0] → 0, c[34, 1] → 0, c[34, 2] → 0, c[34, 3] → 0,
c[34, 4] → 1, c[34, 5] → 0, c[34, 6] → 0, c[34, 7] → 0, c[34, 8] → 0, c[35, 0] → 0,
c[35, 1] → 1, c[35, 2] → 0, c[35, 3] → 0, c[35, 4] → 0, c[35, 5] → 0, c[35, 6] → 0,
c[35, 7] → 0, c[35, 8] → 0, c[36, 0] → 0, c[36, 1] → 1, c[36, 2] → 0, c[36, 3] → 0,
c[36, 4] → 0, c[36, 5] → 0, c[36, 6] → 0, c[36, 7] → 0, c[36, 8] → 0, c[37, 0] → 0,
c[37, 1] → 0, c[37, 2] → 0, c[37, 3] → 0, c[37, 4] → 1, c[37, 5] → 1, c[37, 6] → 0,
c[37, 7] → 0, c[37, 8] → 0, c[38, 0] → 0, c[38, 1] → 0, c[38, 2] → 1, c[38, 3] → 1,
c[38, 4] → 0, c[38, 5] → 0, c[38, 6] → 2, c[38, 7] → 0, c[38, 8] → 0, c[39, 0] → 0,
c[39, 1] → 0, c[39, 2] → 0, c[39, 3] → 0, c[39, 4] → 1, c[39, 5] → 1, c[39, 6] → 0,
c[39, 7] → 0, c[39, 8] → 0, c[40, 0] → 0, c[40, 1] → 0, c[40, 2] → 0, c[40, 3] → 1,
c[40, 4] → 0, c[40, 5] → 0, c[40, 6] → 1, c[40, 7] → 0, c[40, 8] → 0, c[41, 0] → 0,
c[41, 1] → 0, c[41, 2] → 0, c[41, 3] → 0, c[41, 4] → 1, c[41, 5] → 1, c[41, 6] → 0,
c[41, 7] → 0, c[41, 8] → 0, c[42, 0] → 0, c[42, 1] → 1, c[42, 2] → 0, c[42, 3] → 0,
c[42, 4] → 0, c[42, 5] → 0, c[42, 6] → 0, c[42, 7] → 2, c[42, 8] → 1, c[43, 0] → 0,
c[43, 1] → 1, c[43, 2] → 0, c[43, 3] → 0, c[43, 4] → 0, c[43, 5] → 0, c[43, 6] → 0,
c[43, 7] → 1, c[43, 8] → 0, c[44, 0] → 1, c[44, 1] → 0, c[44, 2] → 0, c[44, 3] → 0,
c[44, 4] → 1, c[44, 5] → 2, c[44, 6] → 0, c[44, 7] → 0, c[44, 8] → 0, c[45, 0] → 0,
c[45, 1] → 1, c[45, 2] → 0, c[45, 3] → 0, c[45, 4] → 0, c[45, 5] → 0, c[45, 6] → 0,
c[45, 7] → 1, c[45, 8] → 0, c[47, 0] → 0, c[47, 1] → 1, c[47, 2] → 0, c[47, 3] → 0,
c[47, 4] → 0, c[47, 5] → 0, c[47, 6] → 0, c[47, 7] → 1, c[47, 8] → 0};
Print["number of known variables before:", Length@knownsols];
equations = coeffList;
equations = equations /. knownsols;
equations = DeleteDuplicates@ Select[equations, Max[exponents[#]] == 1 &];
Print["number of equations used:", Length[equations]];
equations = Map[# == 0 &, equations];
equations = Join[equations, knownsols /. Rule → Equal];
solution = Solve[equations, unknowns];
If[Length[solution] == 1,
  newSols = Select[solution[[1]], NumericQ[#[[2]]] &];
  knownsols = Union[knownsols, newSols];
  Print["number of known variables after: ", {Length[#], #} &@knownsols];
]
]
number of known variables before:261
number of equations used:45

```

number of known variables after:

[illegible]

[illegible]

final output

```

In[ ]:= {x[extractxindex[#]], ToString[ # /. variableToVerlindeImage /. solution /.
        ringElement → Identity]} & /@ clustervariables

% //
TableForm
Out[ ]:= {{x[1], V1}, {x[2], V3}, {x[3], V2 + V6}, {x[4], V8}, {x[5], V0 + V4 + 2 V5},
          {x[6], V0}, {x[7], V0 + V5}, {x[8], V0}, {x[9], V1 + V7}, {x[10], V3}, {x[11], V2},
          {x[12], V8}, {x[13], V2}, {x[14], V4}, {x[15], V7 + V8}, {x[16], V7 + V8},
          {x[17], V4}, {x[18], V2 + V6}, {x[19], V2}, {x[20], V2 + V6}, {x[21], V0 + V4 + 2 V5},
          {x[22], V3 + V6}, {x[23], V2}, {x[24], V4 + V5}, {x[25], V4}, {x[26], V1},
          {x[27], V2}, {x[28], V2 + V6}, {x[29], V1}, {x[30], V0 + V5}, {x[31], V0 + V4 + 2 V5},
          {x[32], V2 + V6}, {x[33], V0 + V5}, {x[34], V4}, {x[35], V1}, {x[36], V1},
          {x[37], V4 + V5}, {x[38], V2 + V3 + 2 V6}, {x[39], V4 + V5}, {x[40], V3 + V6},
          {x[41], V4 + V5}, {x[42], V1 + 2 V7 + V8}, {x[43], V1 + V7}, {x[44], V0 + V4 + 2 V5},
          {x[45], V1 + V7}, {x[46], V1 + V7}, {x[47], V1 + V7}, {x[48], V0 + V4 + 2 V5}}

```

Out[]//TableForm=

x[1]	V1
x[2]	V3
x[3]	V2 + V6
x[4]	V8
x[5]	V0 + V4 + 2 V5
x[6]	V0
x[7]	V0 + V5
x[8]	V0
x[9]	V1 + V7
x[10]	V3
x[11]	V2
x[12]	V8
x[13]	V2
x[14]	V4
x[15]	V7 + V8
x[16]	V7 + V8
x[17]	V4
x[18]	V2 + V6
x[19]	V2
x[20]	V2 + V6
x[21]	V0 + V4 + 2 V5
x[22]	V3 + V6
x[23]	V2
x[24]	V4 + V5
x[25]	V4
x[26]	V1
x[27]	V2
x[28]	V2 + V6
x[29]	V1
x[30]	V0 + V5
x[31]	V0 + V4 + 2 V5
x[32]	V2 + V6
x[33]	V0 + V5
x[34]	V4
x[35]	V1
x[36]	V1
x[37]	V4 + V5
x[38]	V2 + V3 + 2 V6
x[39]	V4 + V5
x[40]	V3 + V6
x[41]	V4 + V5
x[42]	V1 + 2 V7 + V8
x[43]	V1 + V7
x[44]	V0 + V4 + 2 V5
x[45]	V1 + V7
x[46]	V1 + V7
x[47]	V1 + V7
x[48]	V0 + V4 + 2 V5