

Pensieve Header: Adding 6 to the 18 conjectures, to make it 24.

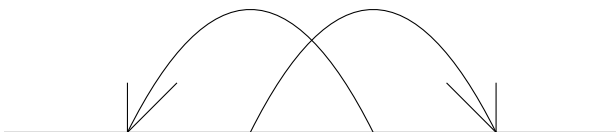
Program and Testing

Drawing Arrow Diagrams

```
In[48]:= SetAttributes[Diag, Orderless];
Draw[expr_] := expr /. diag_Diag => Draw[diag];
Draw[diag_Diag] := Module[
  {n = Length[diag]},
  Graphics[{
    Line[{{0, 0}, {2 n + 1, 0}}],
    (List @@ diag) /. {
      Ar[i_, j_] => {
        BezierCurve[{
          {i, 0}, {(i + j) / 2, Abs[i - j]}, {j, 0}
        }],
        Line[{{j - 0.2 + 0.2 Sign[i - j], 0.4}, {j, 0}, {j + 0.2 + 0.2 Sign[i - j], 0.4}]}
    ]
  ]
];
```

```
In[51]:= Draw[Diag[Ar[3, 1], Ar[2, 4]]]
```

Out[51]=



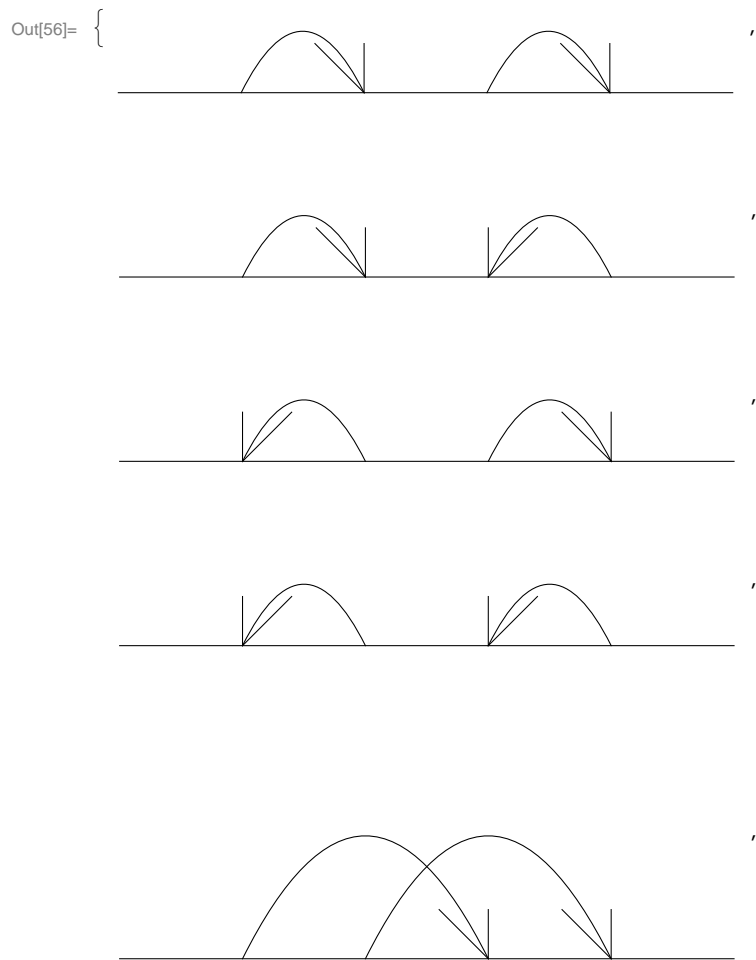
Generating Arrow Diagrams

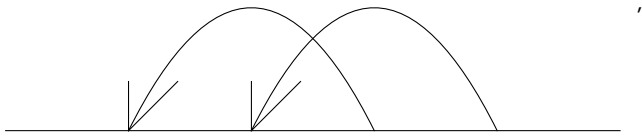
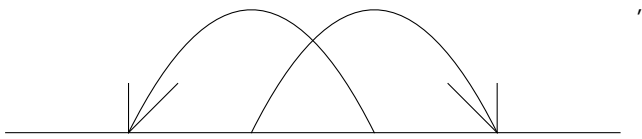
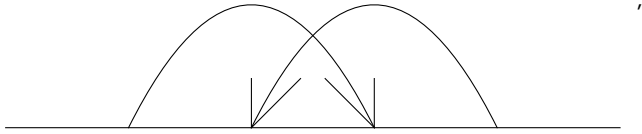
```
In[52]:= Place[{Ar}, {i_, j_}] := {Diag[Ar[i, j]], Diag[Ar[j, i]]};
Place[{Ar, objs_}, {i_, rest_}] := Flatten[Table[
  Outer[Join,
    Place[{Ar}, {i, {rest}[[k]]}],
    Place[{objs}, Delete[{rest}, k]]
  ],
  {k, Length[{rest}}]
]];
Diagrams[k_.*Ar] := Place[Table[Ar, {k}], Range[2 k]];
```

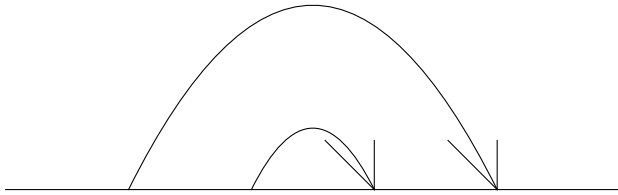
```
In[55]:= Diagrams[2 Ar]
```

```
Out[55]= {Diag[Ar[1, 2], Ar[3, 4]], Diag[Ar[1, 2], Ar[4, 3]], Diag[Ar[2, 1], Ar[3, 4]],
  Diag[Ar[2, 1], Ar[4, 3]], Diag[Ar[1, 3], Ar[2, 4]], Diag[Ar[1, 3], Ar[4, 2]],
  Diag[Ar[2, 4], Ar[3, 1]], Diag[Ar[3, 1], Ar[4, 2]], Diag[Ar[1, 4], Ar[2, 3]],
  Diag[Ar[1, 4], Ar[3, 2]], Diag[Ar[2, 3], Ar[4, 1]], Diag[Ar[3, 2], Ar[4, 1]]}
```

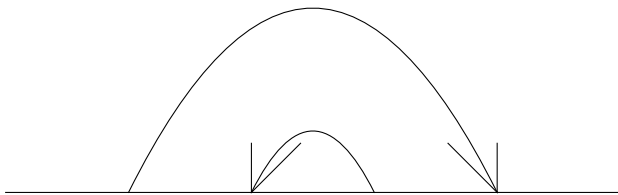
In[56]:= **Draw[Diagrams [2 Ar]]**



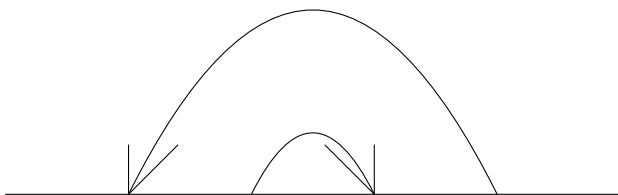




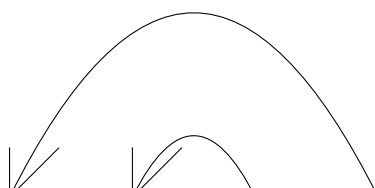
,



,



,



}

Generating Relators

```

In[57]:= Place[{r : (TC | R3b), objs__}, {i_, rest__}] := Flatten[Table[
  Outer[Join,
    Place[{r}, {i, {rest}[[j]], {rest}[[k]]}],
    Place[{objs}, Delete[{rest}, {{j}, {k}}]]
  ],
  {k, 2, Length[{rest]}}, {j, 1, k - 1}
]];
Place[{R3b}, {i_, j_, k_}] :=
  Permutations[{i, j, k}] /. {i1_, j1_, k1_} => Diag[R3b[i1, j1, k1]];
Diagrams[R3b] := Place[{R3b}, {1, 2, 3}];
Diagrams[R3b + k_. * Ar] /; k > 0 := Flatten[
  Place[#, Range[2 k + 3]] & /@ Permutations[Table[Ar, {k}] ~ Append ~ R3b]
];
Place[{TC}, {i_, j_, k_}] := Diag /@ {TC[i, j, k], TC[j, k, i], TC[k, i, j]};
Diagrams[TC] := Place[{TC}, {1, 2, 3}];
Diagrams[TC + k_. * Ar] /; k > 0 := Flatten[
  Place[#, Range[2 k + 3]] & /@ Permutations[Table[Ar, {k}] ~ Append ~ TC]
];
Diagrams[TC + k_. * Ar] /; k < 0 := {};
Place[{R2c}, {i_, j_}] := {Diag[R2c[i, j]], Diag[R2c[j, i]]};
Place[{r : (R2c), objs__}, {i_, rest__}] := Flatten[Table[
  Outer[Join,
    Place[{r}, {i, {rest}[[j]]}],
    Place[{objs}, Delete[{rest}, j]]
  ],
  {j, 1, Length[{rest]}]
}];
Diagrams[R2c] := Place[{R2c}, {1, 2}];
Diagrams[R2c + k_. * Ar] /; k > 0 := Flatten[
  Place[#, Range[2 k + 2]] & /@ Permutations[Table[Ar, {k}] ~ Append ~ R2c]
];
Diagrams[R2c + k_. * Ar] /; k < 0 := {};
Place[{R1d}, {i_}] := {Diag[R1d[i]]};
Place[{R1a}, {i_}] := {Diag[R1a[i]]};
Place[{r : (R1d | R1a), objs__}, {i_, rest__}] := Flatten[
  Outer[Join,
    Place[{r}, {i}],
    Place[{objs}, {rest}]
  ]
];
Diagrams[R1d] := Place[{R1d}, {1}];
Diagrams[R1a] := Place[{R1a}, {1}];
Diagrams[R1d + k_. * Ar] /; k > 0 := Flatten[
  Place[#, Range[2 k + 1]] & /@ Permutations[Table[Ar, {k}] ~ Append ~ R1d]
];
Diagrams[R1a + k_. * Ar] /; k > 0 := Flatten[
  Place[#, Range[2 k + 1]] & /@ Permutations[Table[Ar, {k}] ~ Append ~ R1a]
];
Diagrams[R1d + k_. * Ar] /; k < 0 := {};
Diagrams[R1a + k_. * Ar] /; k < 0 := {};

```

```
In[79]= Take[Diagrams[R3b + 1 Ar], 5]
```

```
Out[79]= {Diag[Ar[1, 2], R3b[3, 4, 5]], Diag[Ar[1, 2], R3b[3, 5, 4]],  
         Diag[Ar[1, 2], R3b[4, 3, 5]], Diag[Ar[1, 2], R3b[4, 5, 3]], Diag[Ar[1, 2], R3b[5, 3, 4]]}
```

```
In[80]= {Diagrams[R1d + Ar], Diagrams[R1a + Ar]}
```

```
Out[80]= {{Diag[Ar[1, 2], R1d[3]], Diag[Ar[2, 1], R1d[3]], Diag[Ar[1, 3], R1d[2]],  
          Diag[Ar[3, 1], R1d[2]], Diag[Ar[2, 3], R1d[1]], Diag[Ar[3, 2], R1d[1]]},  
         {Diag[Ar[1, 2], R1a[3]], Diag[Ar[2, 1], R1a[3]], Diag[Ar[1, 3], R1a[2]],  
          Diag[Ar[3, 1], R1a[2]], Diag[Ar[2, 3], R1a[1]], Diag[Ar[3, 2], R1a[1]]}}
```

Generating Relations

```

In[81]:= NormalizeDiag[diag_Diag] := Module[
  {indices = Union@@ (List @@ diag /. Ar -> List)},
  diag /. Thread[indices -> Range[Length[indices]]]
];

R[Diag[lft___, R3b[i_, j_, k_], rgt___]] := (
  (* RHS of (9) *)
  NormalizeDiag[Diag[lft, Ar[i, j], Ar[i+0.5, k], Ar[j+0.5, k+0.5], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, j], Ar[i+0.5, k], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, j], Ar[j+0.5, k], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, k], Ar[j, k+0.5], rgt]]
  (* LHS of (9) *)
  - NormalizeDiag[Diag[lft, Ar[j, k], Ar[i, k+0.5], Ar[i+0.5, j+0.5], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k], Ar[i+0.5, j], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, j+0.5], Ar[j, k], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k+0.5], Ar[j, k], rgt]]
);

RWrong1[Diag[lft___, R3b[i_, j_, k_], rgt___]] := (
  (* RHS of (9) *)
  NormalizeDiag[Diag[lft, Ar[i, j], Ar[i+0.5, k], Ar[j+0.5, k+0.5], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, j], Ar[i+0.5, k], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, j], Ar[j+0.5, k], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, k], Ar[j, k+0.5], rgt]]
  (* LHS of (9) *)
  + NormalizeDiag[Diag[lft, Ar[j, k], Ar[i, k+0.5], Ar[i+0.5, j+0.5], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k], Ar[i+0.5, j], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, j+0.5], Ar[j, k], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k+0.5], Ar[j, k], rgt]]
);

RWrong2[Diag[lft___, R3b[i_, j_, k_], rgt___]] := (
  (* RHS of (9) *)
  NormalizeDiag[Diag[lft, Ar[i, j], Ar[i+0.5, k], Ar[j+0.5, k+0.5], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, j], Ar[i+0.5, k], rgt]]
  + NormalizeDiag[Diag[lft, Ar[i, j], Ar[j+0.5, k], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k], Ar[j, k+0.5], rgt]]
  (* LHS of (9) *)
  - NormalizeDiag[Diag[lft, Ar[j, k], Ar[i, k+0.5], Ar[i+0.5, j+0.5], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k], Ar[i+0.5, j], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, j+0.5], Ar[j, k], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k+0.5], Ar[j, k], rgt]]
);

```

```

RMinus[n_, Diag[lft___, R3b[i_, j_, k_], rgt___]] := Module[
  {Expand2, Expand3, n1},
  n1 = n - Length[{lft, rgt}];
  Expand2[Diag[Ar[i1_, j1_], Ar[i2_, j2_]]] := Sum[
    (-1)^(k1 + k2) * NormalizeDiag[Flatten[Diag[
      lft,
      Diag@@Table[Ar[i1 + 0.01 q, j1 + 0.01 q], {q, k1}],
      Diag@@Table[Ar[i2 + 0.01 q, j2 + 0.01 q], {q, k2}],
      rgt
    ]]],
    {k1, n1 - 1}, {k2, n1 - k1}
  ];
  Expand3[Diag[Ar[i1_, j1_], Ar[i2_, j2_], Ar[i3_, j3_]]] := Sum[
    (-1)^(k1 + k2 + k3) * NormalizeDiag[Flatten[Diag[
      lft,
      Diag@@Table[Ar[i1 + 0.01 q, j1 + 0.01 q], {q, k1}],
      Diag@@Table[Ar[i2 + 0.01 q, j2 + 0.01 q], {q, k2}],
      Diag@@Table[Ar[i3 + 0.01 q, j3 + 0.01 q], {q, k3}],
      rgt
    ]]],
    {k1, n1 - 2}, {k2, n1 - k1 - 1}, {k3, n1 - k1 - k2}
  ];
  Expand[Plus[
    (
      Diag[Ar[i, j], Ar[i + 0.5, k]] +
      Diag[Ar[i, j], Ar[j + 0.5, k]] + Diag[Ar[i, k], Ar[j, k + 0.5]]
      - Diag[Ar[i, k], Ar[i + 0.5, j]] - Diag[Ar[i, j + 0.5], Ar[j, k]] -
      Diag[Ar[i, k + 0.5], Ar[j, k]]
    ) /. d_Diag -> Expand2[d],
    (Diag[Ar[i, j], Ar[i + 0.5, k], Ar[j + 0.5, k + 0.5]] -
      Diag[Ar[j, k], Ar[i, k + 0.5], Ar[i + 0.5, j + 0.5]]) /. d_Diag -> Expand3[d]
  ]
];
R[n_, Diag[lft___, R2c[i_, j_], rgt___]] := Module[
  {n1},
  n1 = n - Length[{lft, rgt}];
  Sum[
    Expand[(-1)^k * s * NormalizeDiag[Flatten[Diag[
      lft,
      Diag@@Table[Ar[i + 0.01 q, j + s * 0.01 q], {q, k}],
      rgt
    ]]]],
  ]
];

```

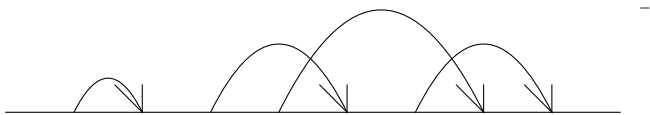
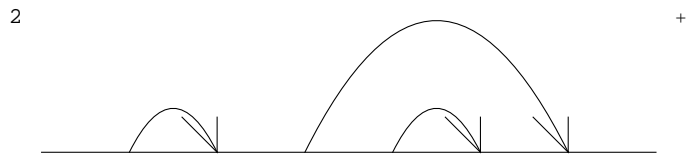
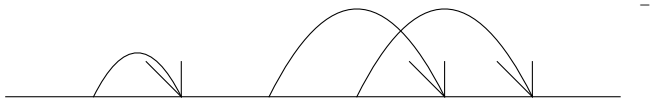
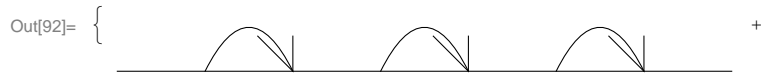


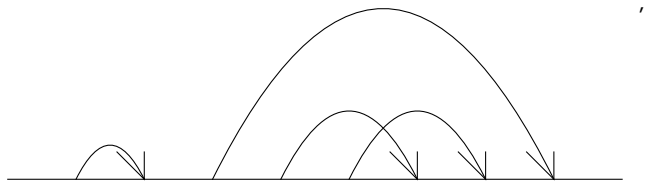
```

      {k, 2, n1}, {s, -1, 1, 2}
    ]
  ];
R[n_, Diag[lft___, Rld[i_], rgt___]] := Module[
  {n1},
  n1 = n - Length[{lft, rgt}];
  Sum[
    Expand[(-1)^k * NormalizeDiag[Flatten[Diag[
      lft,
      Diag@@Table[Ar[i + 0.01 q, i + 0.5 + 0.01 q], {q, k}],
      rgt
    ]]],
    {k, 2, n1}
  ]
];
R[n_, Diag[lft___, Rla[i_], rgt___]] := Module[
  {n1},
  n1 = n - Length[{lft, rgt}];
  Sum[
    Expand[(-1)^k * NormalizeDiag[Flatten[Diag[
      lft,
      Diag@@Table[Ar[i + 0.5 + 0.01 q, i + 0.01 q], {q, k}],
      rgt
    ]]],
    {k, 2, n1}
  ]
];
R[Diag[lft___, TC[i_, j_, k_], rgt___]] := (
  +NormalizeDiag[Diag[lft, Ar[k, i], Ar[k + 0.5, j], rgt]]
  - NormalizeDiag[Diag[lft, Ar[k + 0.5, i], Ar[k, j], rgt]]
);
ContainsShortArrow[diag_Diag] := (1 == Min[List@@diag /. Ar[i_, j_] => Abs[i - j]]);
DescendingQ[diag_Diag] := And @@ (OrderedQ /@ diag);

In[92]:= Draw[R /@ Take[Diagrams[R3b + 1 Ar], 2]]

```

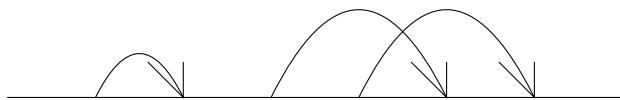




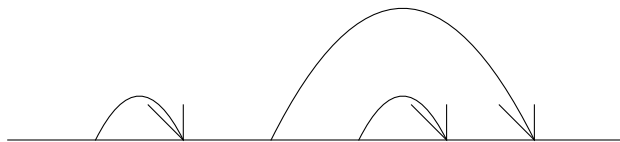
'



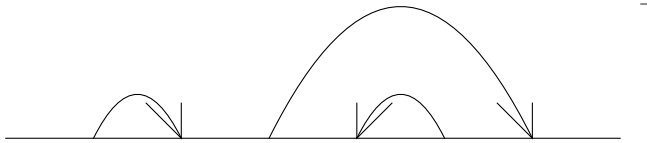
-



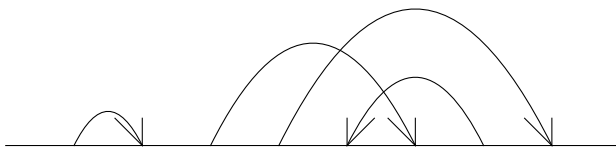
+



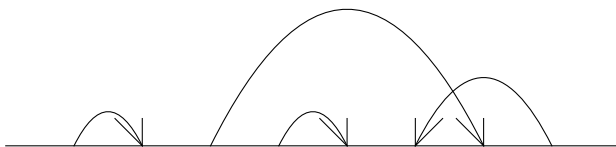
-



-



+



}

Producing "Quotient Spaces"

```
In[93]:= Options[PSpace] = {Debug → False, Gr → False, ReduceEquivalences → True};
PSpace[m_] /; m < 2 := QuotientSpace[Diagrams[m Ar], {}];
PSpace[m_, reltypes_List, opts___] /; m ≥ 2 := Module[
{
  debug = Debug /. {opts} /. Options[PSpace],
  gr = Gr /. {opts} /. Options[PSpace],
  reduceequivalences = ReduceEquivalences /. {opts} /. Options[PSpace],
  diags, rels, i, filter, equalities, equivclasses, pos, redrule
```

```

},
If[debug, Print[Date[], ": Starting work..."]];
diags = Join@@Table[Diagrams[k Ar], {k, If[gr, m, 1], m}];
filter = Identity;
reltypes /. {
  "R1" => (
    diags = Select[diags, !ContainsShortArrow[#] &]
  ),
  "Descending" => (
    filter = Select[#, DescendingQ] &;
    diags = Select[diags, DescendingQ]
  )
};
If[debug, Print[Date[], ": Computed ", Length[diags], " diagrams..."]];
rels = Join@@(reltypes /. {
  "R1" -> {},
  "Descending" -> {},
  "Round" => If[MemberQ[reltypes, "Descending"],
    diags /.
    d_Diag => Plus[
      NormalizeDiag[d /. i_Integer => i-1 /. Ar[0, i_] => Ar[i, 2 Length[d]]],
      d
    ],
    diags /.
    d_Diag => (NormalizeDiag[d /. {1 -> 2 Length[d], i_Integer => i-1}] - d)
  ],
  "R3b" => (
    (R /@ Join@@Table[Diagrams[R3b+k Ar] // filter, {k, If[gr, m-2, 0], m-2}]) /.
    diag_Diag /; Length[diag] > m => 0
  ),
  "RWrong1" => (
    (RWrong1 /@ Join@@Table[Diagrams[R3b+k Ar], {k, If[gr, m-2, 0], m-2}]) /.
    diag_Diag /; Length[diag] > m => 0
  ),
  "RWrong2" => (
    (RWrong2 /@ Join@@Table[Diagrams[R3b+k Ar], {k, If[gr, m-2, 0], m-2}]) /.
    diag_Diag /; Length[diag] > m => 0
  ),
  "R3bMinus" => (
    RMinus[m, #] & /@
    Join@@Table[Diagrams[R3b+k Ar] // filter, {k, If[gr, m-2, 0], m-2}]]

```

```

    ),
    "R2c" => (
      R[m, #] & /@
      Join @@ Table[Diagrams[R2c + k Ar] // filter, {k, If[gr, m - 2, 0], m - 2}]
    ),
    "R1d" => (
      R[m, #] & /@
      Join @@ Table[Diagrams[R1d + k Ar] // filter, {k, If[gr, m - 2, 0], m - 2}]
    ),
    "R1a" => (
      R[m, #] & /@
      Join @@ Table[Diagrams[R1a + k Ar] // filter, {k, If[gr, m - 2, 0], m - 2}]
    ),
    "TC" => (
      R /@ Join @@ Table[Diagrams[TC + k Ar], {k, If[gr, m - 2, 0], m - 2}]
    )
  });
reltypes /. {
  "R1" => (
    rels = DeleteCases[rels /. {d_Diag /; ContainsShortArrow[d] => 0}, 0]
  )
};
If[debug, Print[Date[], ": Computed ", Length[rels], " relations..."]];
If[reduceequivalences,
  equalities = Cases[rels, d1_Diag - d2_Diag => {d1, d2}];
  equivclasses = Fold[
    (
      pos = First /@ Position[#1, #2];
      Append[Delete[#1, List /@ pos], Union @@ (#1[[pos]])]
    ) &,
    equalities,
    Union @@ equalities
  ];
  redrule = Dispatch[Flatten[Table[
    class = equivclasses[[i]];
    (# -> First[class]) & /@ Rest[class],
    {i, Length[equivclasses]}
  ]]];
  diags = Union[diags /. redrule];
  rels = Union[rels /. redrule];
];
QuotientSpace[diags, rels]
]

```

Computing Dimensions

```

In[96]:= Options[Dim] = {Debug → False, UseLinBox → False};
Dim[QuotientSpace[diags_List, rels_List], opts___] := Module[
{
  debug = Debug /. {opts} /. Options[Dim],
  uselinbox = UseLinBox /. {opts} /. Options[Dim],
  diagtoindex, mat, rel, matfile, rank
},
diagtoindex = Dispatch[Thread[Rule[diags, Range[Length[diags]]]];
If[uselinbox,
  matfile = OpenWrite["matfile.matrix"];
  WriteString[matfile, Length[rels], " ", Length[diags], " S\n"];
  Do[
    rel = rels[[i]];
    WriteString[matfile, StringJoin @@ Flatten[{
      ToString[Count[{rel}, _Diag, Infinity]], " ",
      {
        ToString[(# /. diagtoindex) - 1], " ",
        ToString[Coefficient[rel, #]], " "
      } & /@ Cases[{rel}, diag_Diag, Infinity],
      "\n"
    }]],
    {i, Length[rels]}
  ];
  matfile = Close[matfile];
  rank = RunThrough["./ComputeRank " <> matfile, Random[]][[2]];
  Length[diags] - rank,
  (* else *) mat = SparseArray[
    Join @@ Table[
      rel = rels[[i]];
      {i, # /. diagtoindex} → Coefficient[rel, #] & /@
      Cases[{rel}, diag_Diag, Infinity],
      {i, Length[rels]}
    ],
    {Length[rels], Length[diags]}
  ];
  If[debug, Print[Date[], ": Computed mat..."]];
  Length[diags] - MatrixRank[mat]
]
];
DimP[m_, reltypes_List, opts___] := Dim[PSpace[m, reltypes, opts], opts]

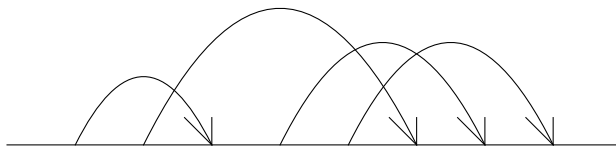
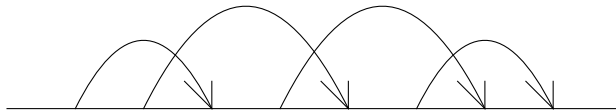
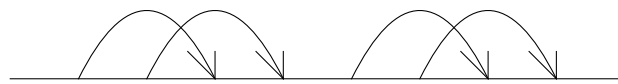
```

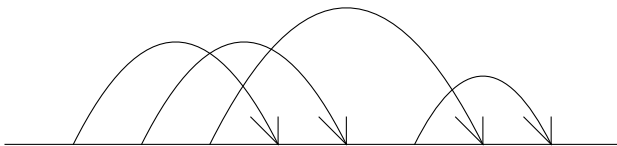
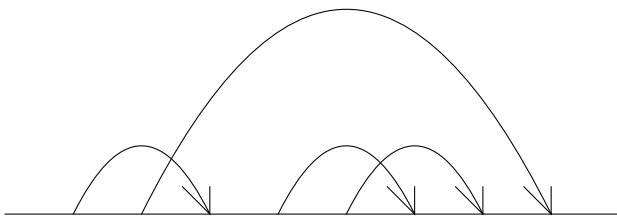
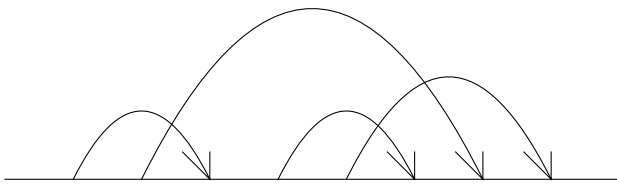
Analysis

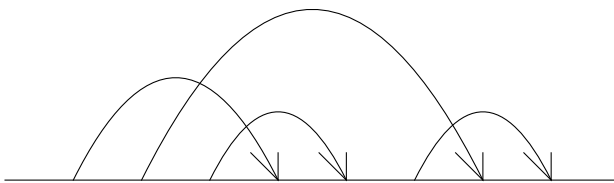
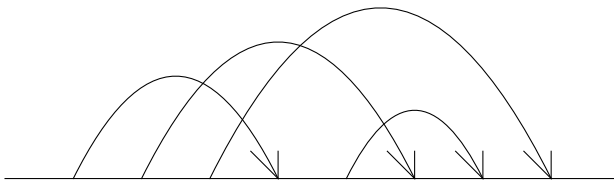
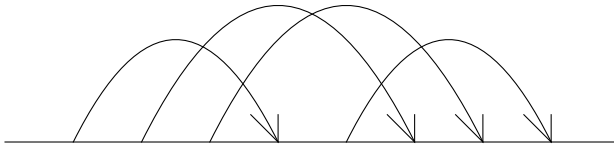
```
In[99]:= Analyze[m_Integer, reltypes_List, opts___Rule] := Module[
  {qs},
  Table[
    {
      qs = PSpace[n, reltypes, Gr → True, opts];
      (Length /@ qs) → Dim[qs, opts],
      qs = PSpace[n, reltypes, Gr → False, opts];
      (Length /@ qs) → Dim[qs, opts]
    },
    {n, 2, m}
  ]
];
Analyze[reltypes_List, opts___Rule] := Analyze[5, reltypes, opts];
```

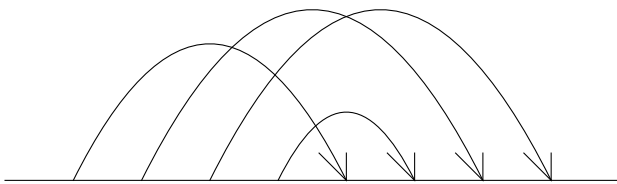
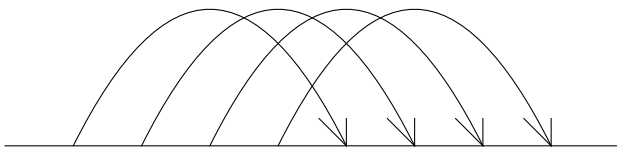
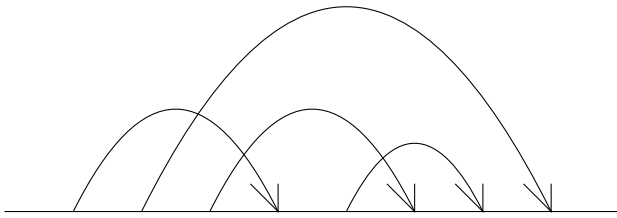
```
In[128]:= Draw[PSpace[4, {"R3b", "R2c", "R1", "Round", "Descending"}, Gr → True]]
```

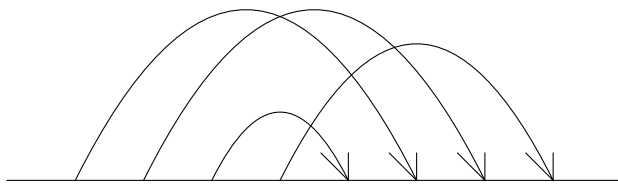
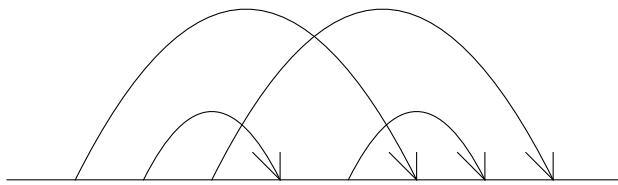
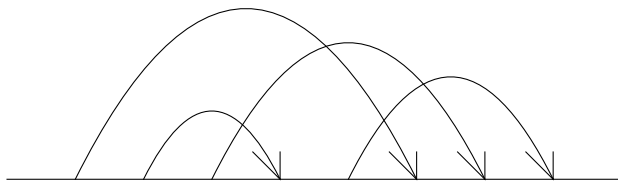
Out[128]= QuotientSpace [{

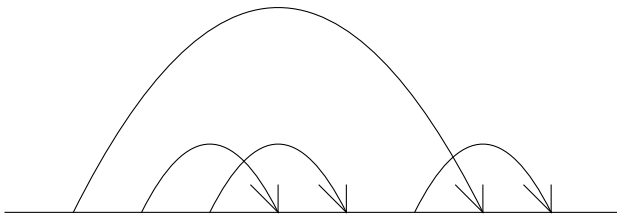
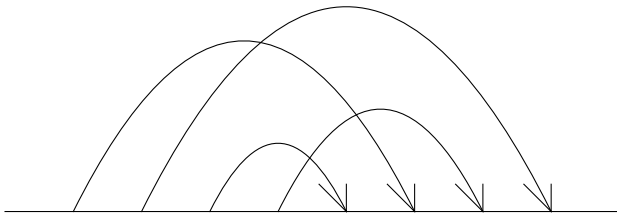


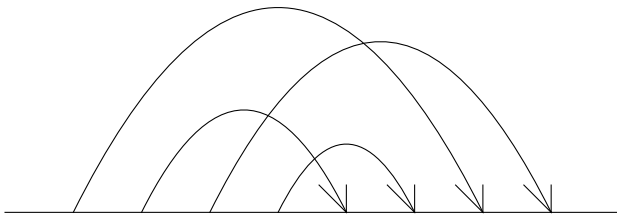
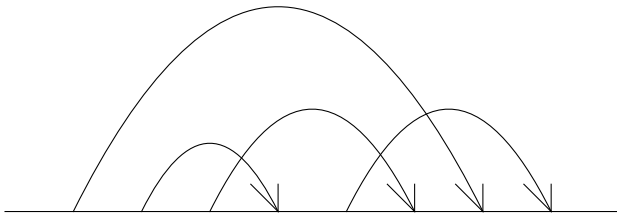


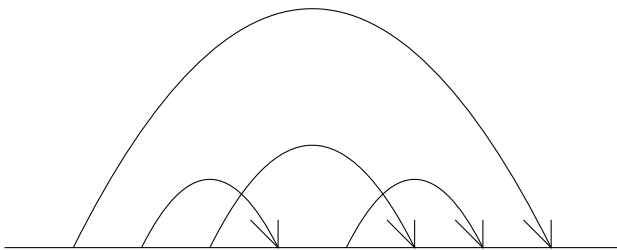
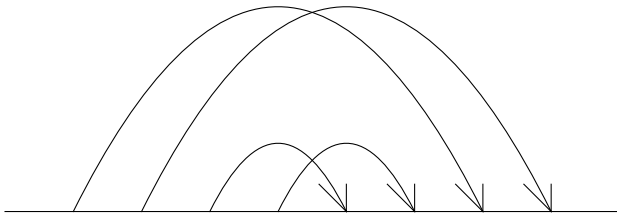


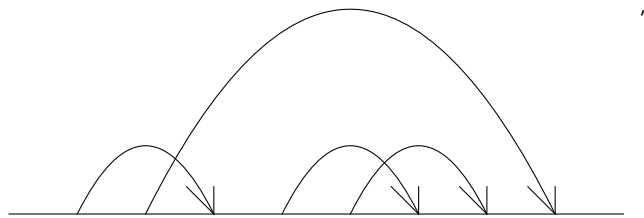
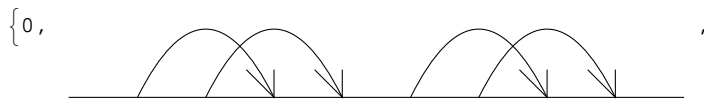
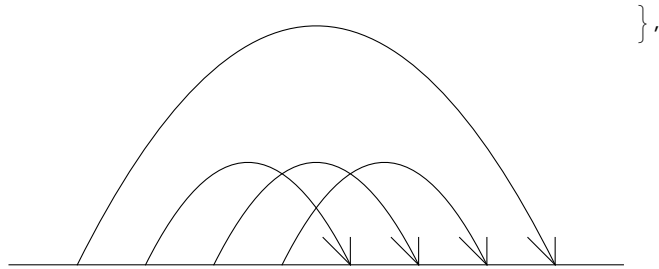


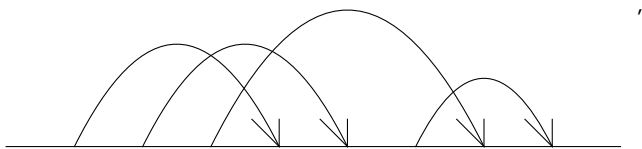
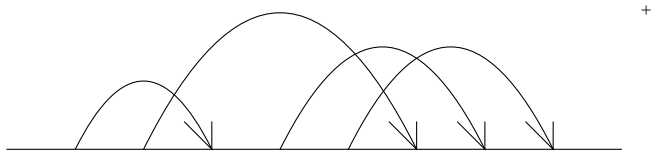
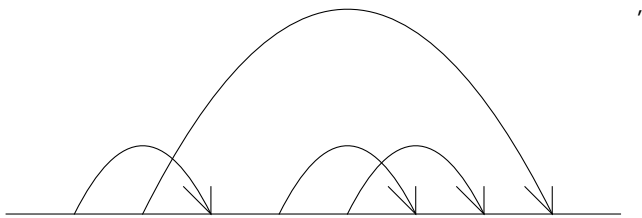
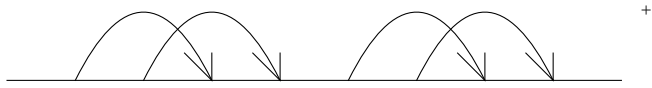


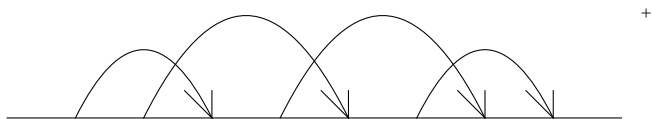
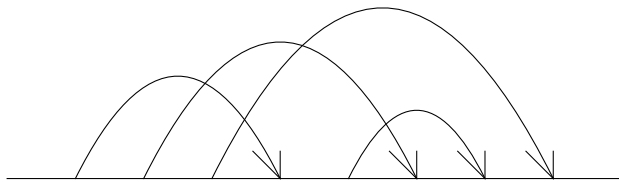
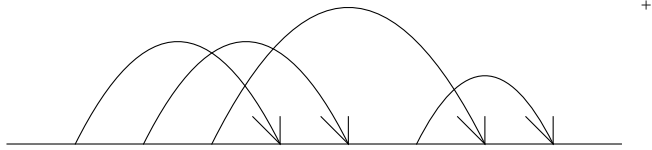


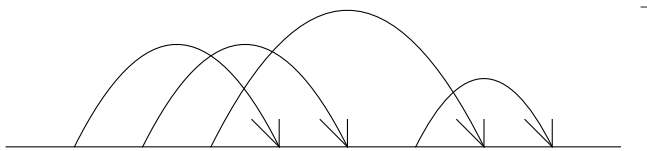
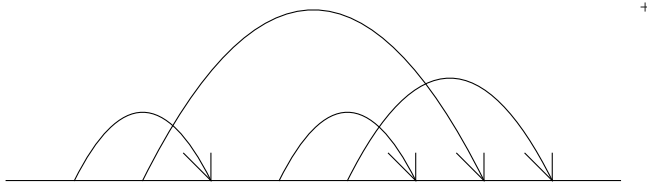
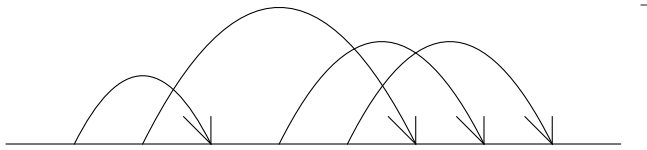


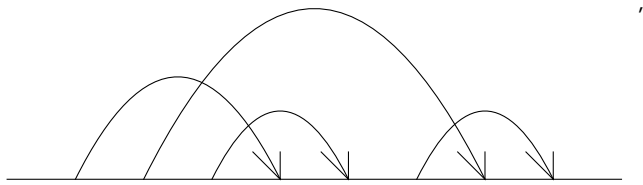




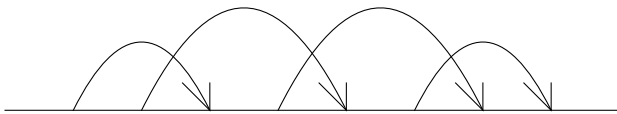




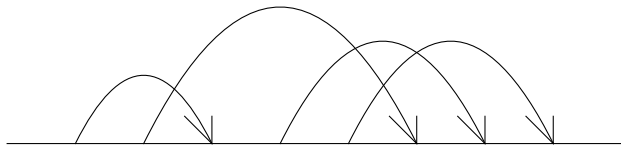




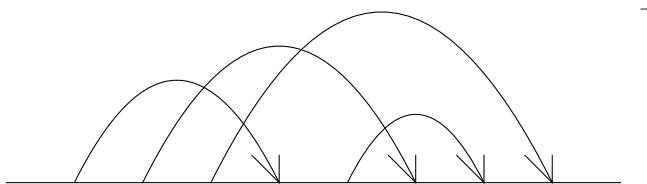
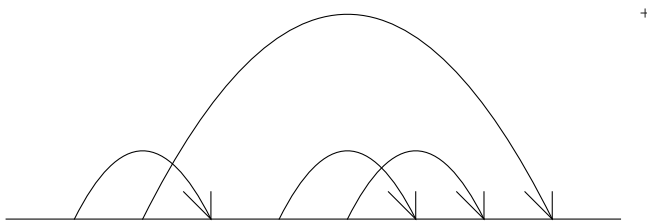
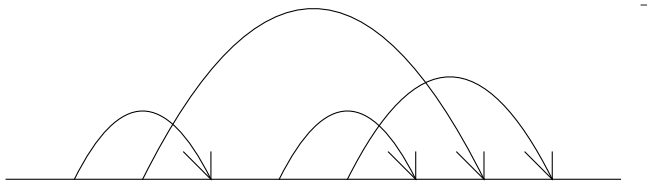
,

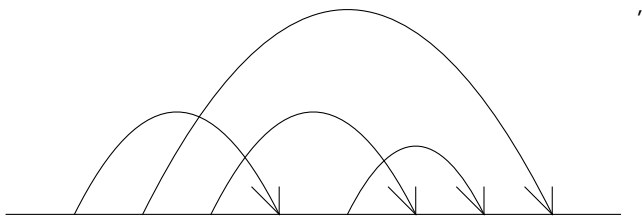
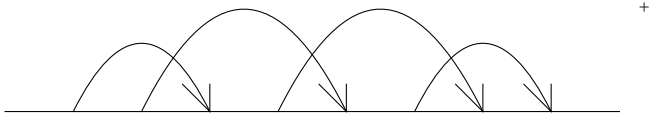
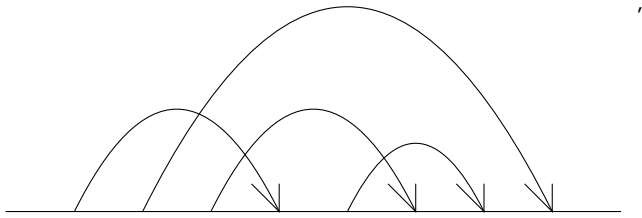


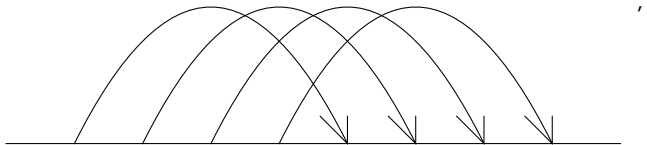
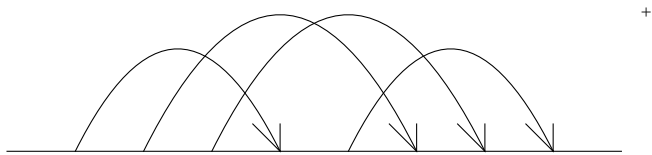
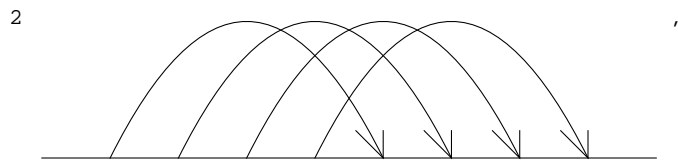
-

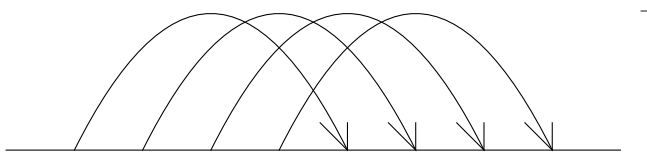
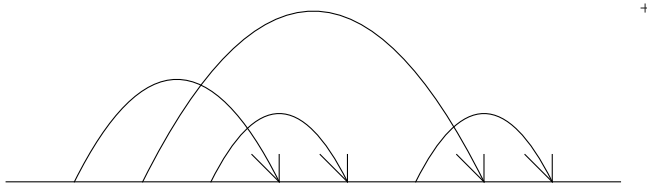
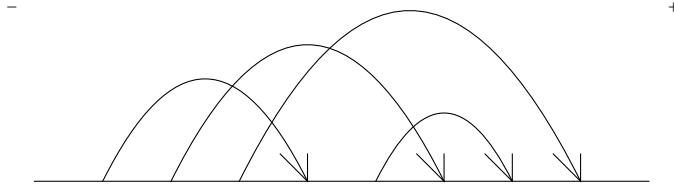


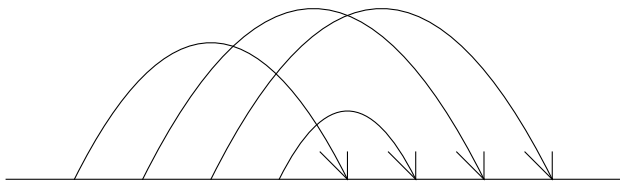
+



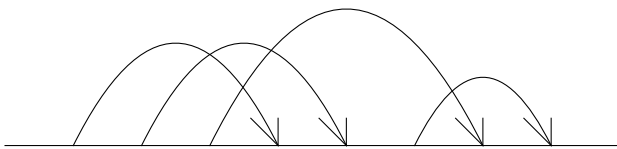




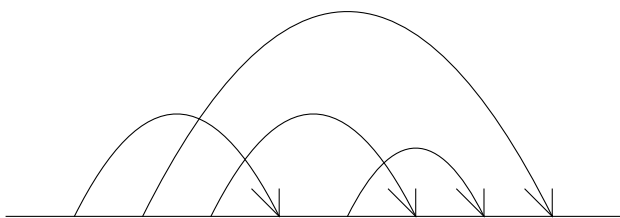




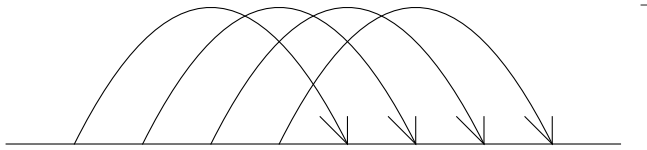
,



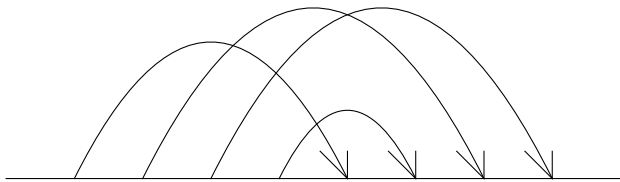
-



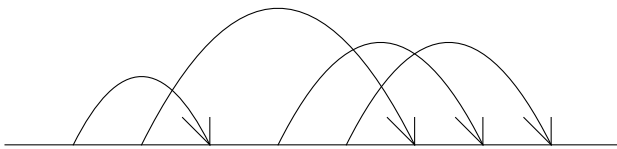
+



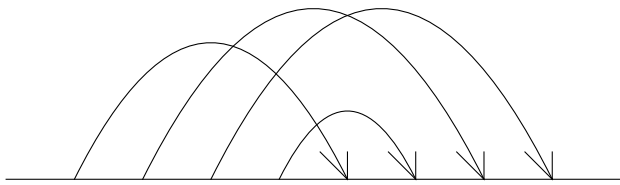
-



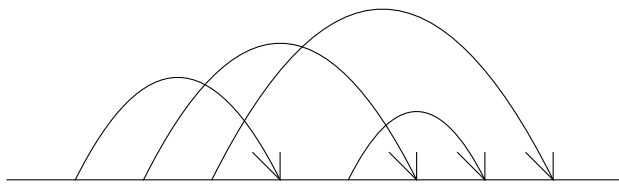
,



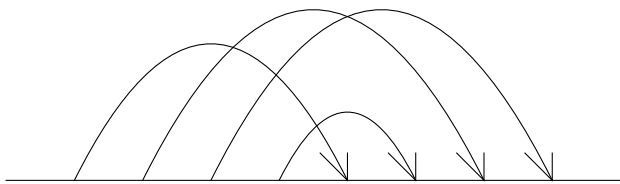
+



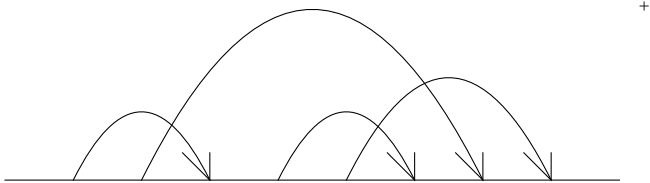
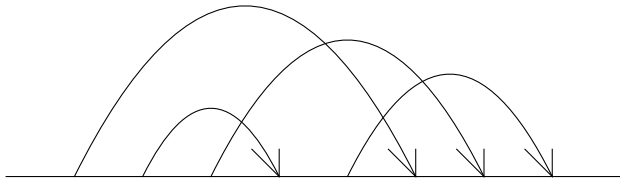
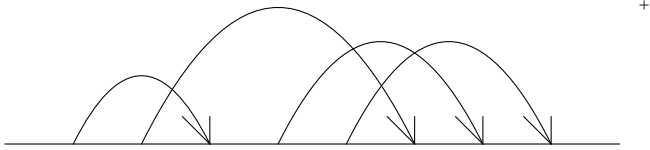
,

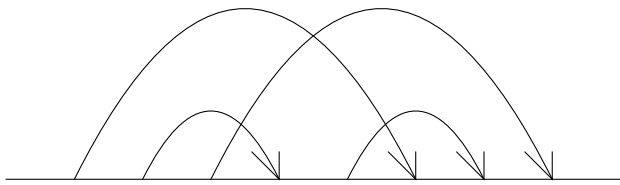


+

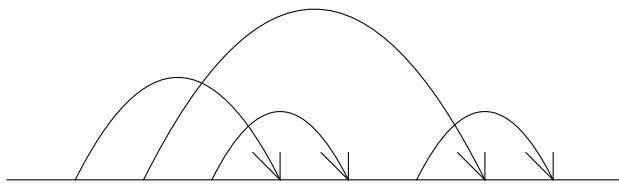


,

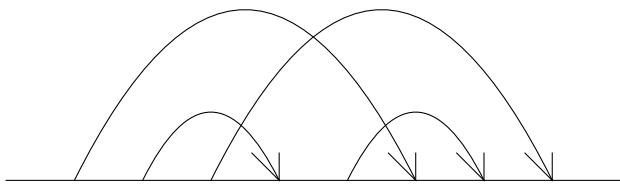




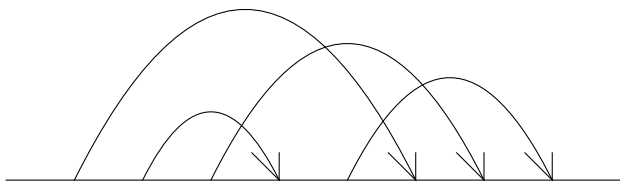
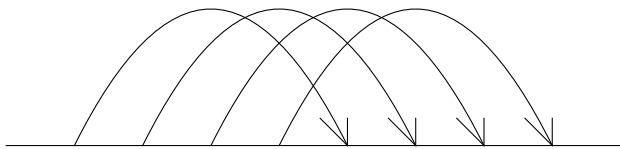
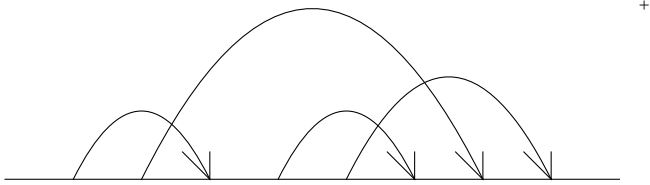
,

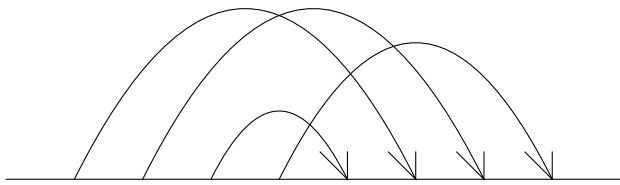


+

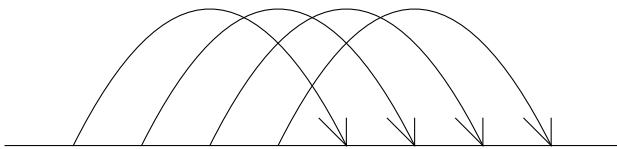


,

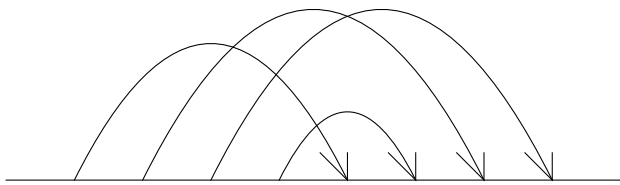




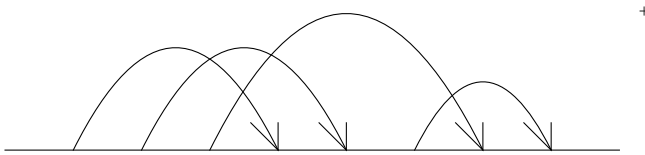
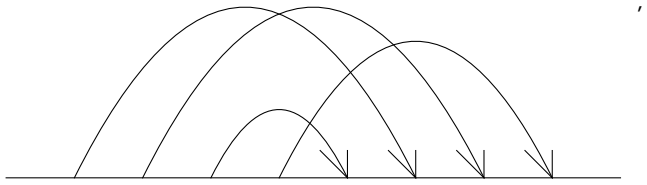
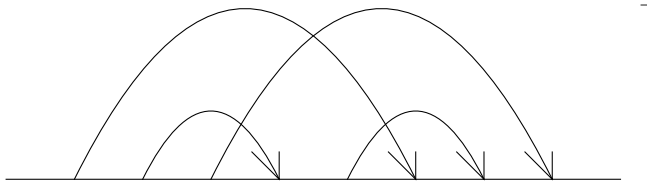
,

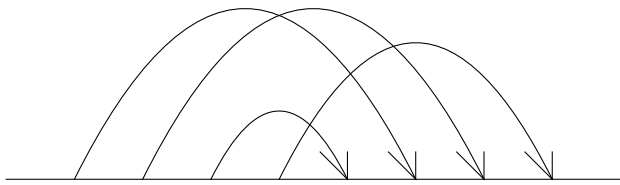


-

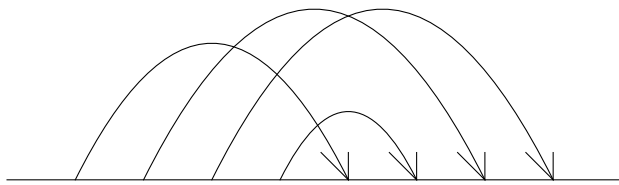


+

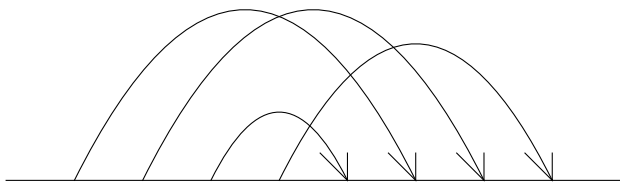




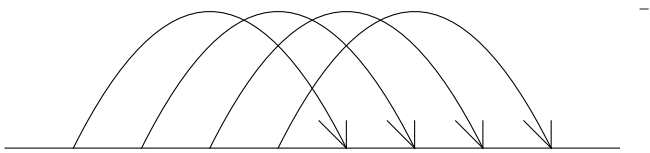
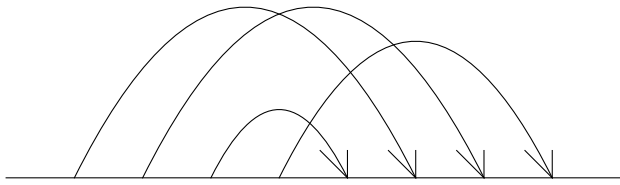
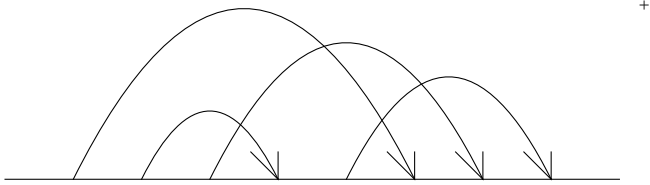
,

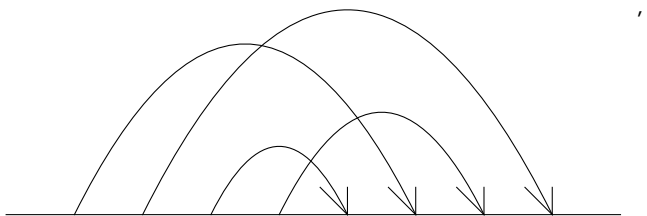
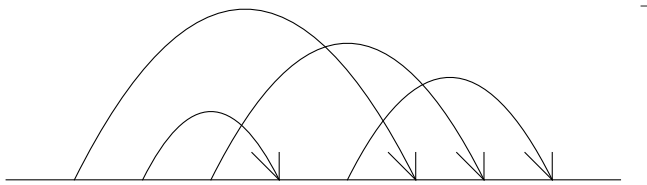
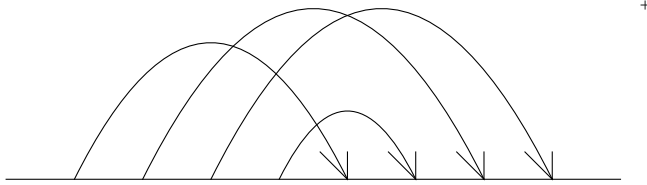


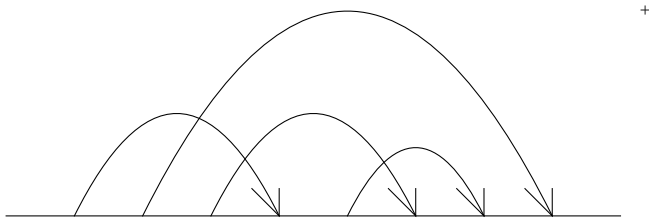
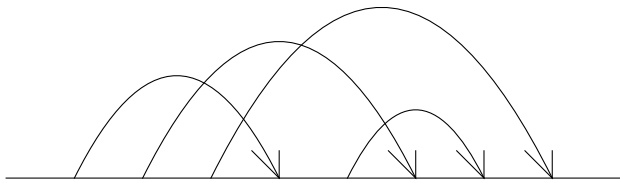
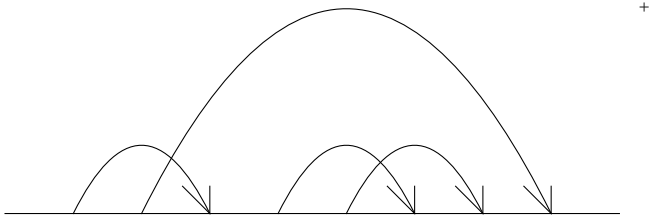
+

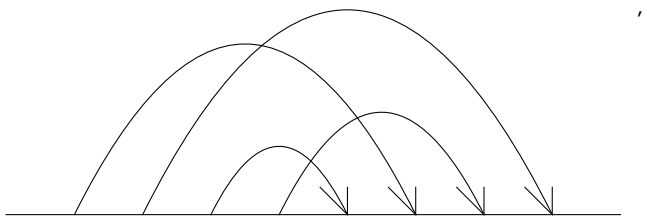
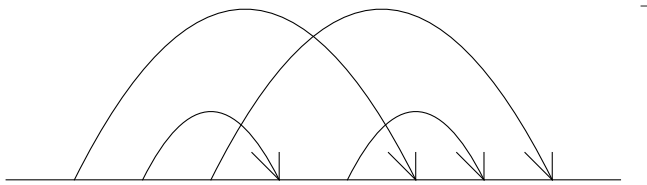
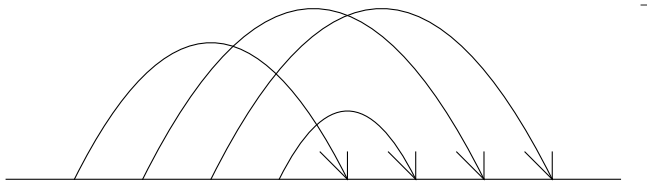


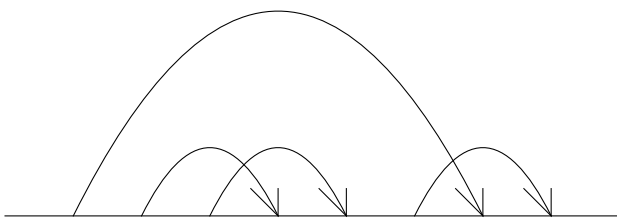
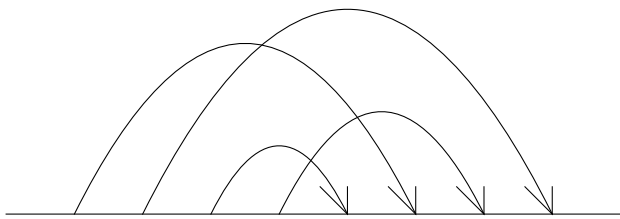
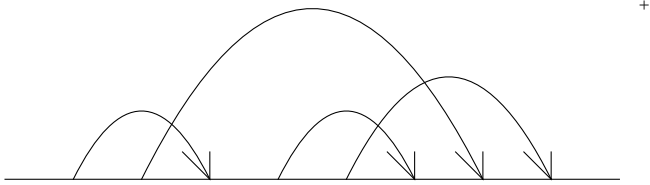
,

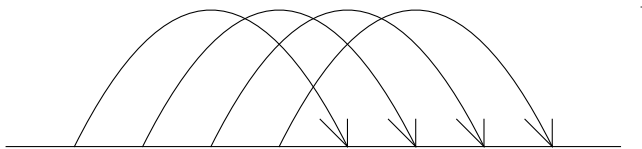
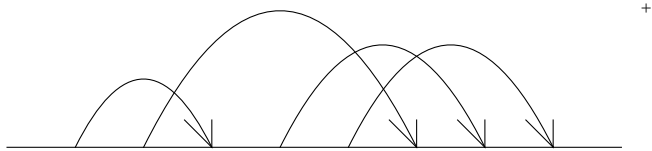
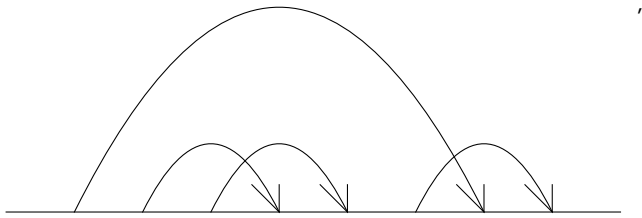
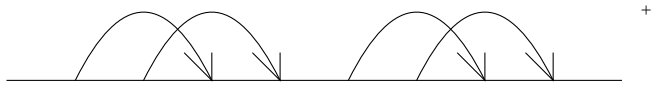


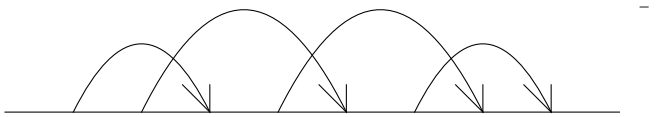
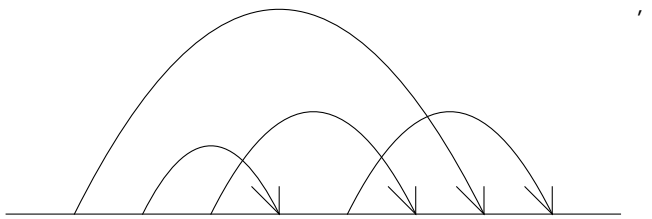
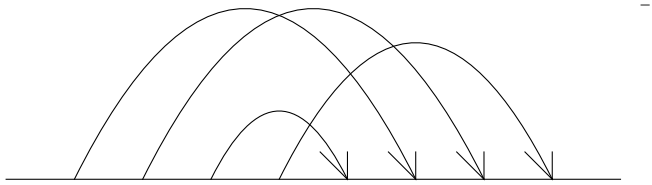


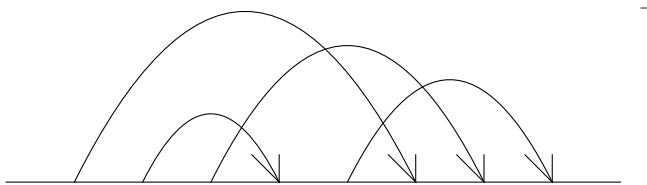
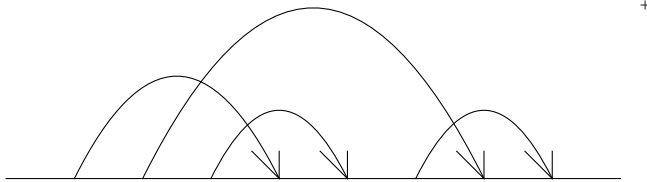
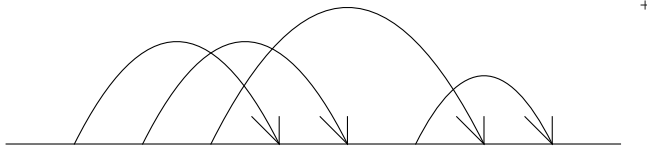


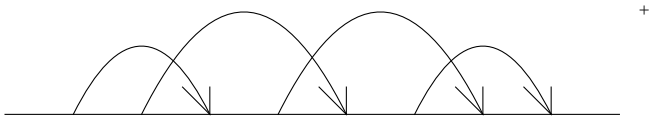
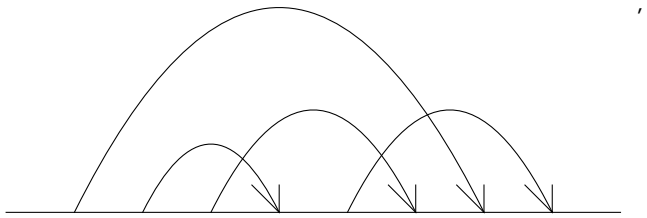
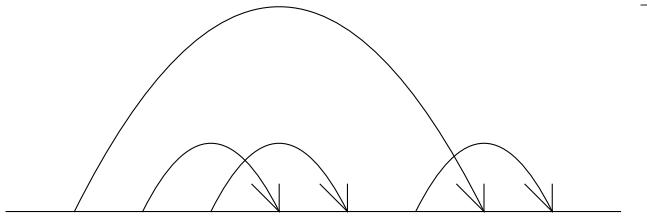


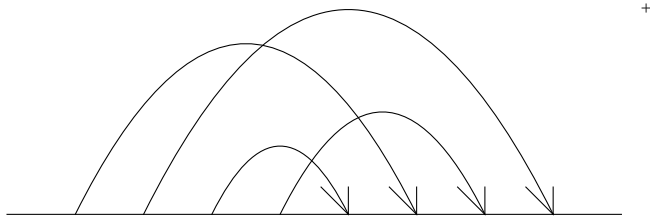
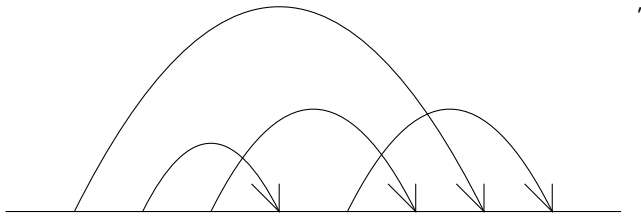


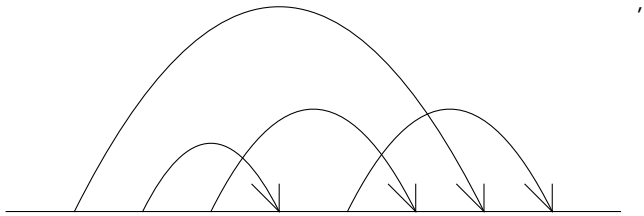




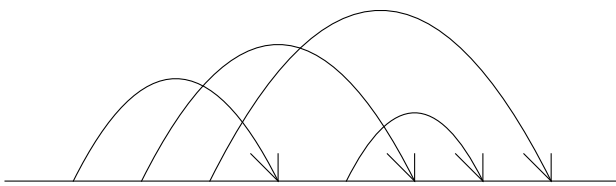




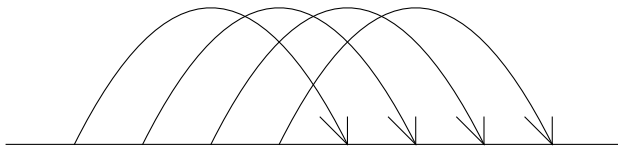




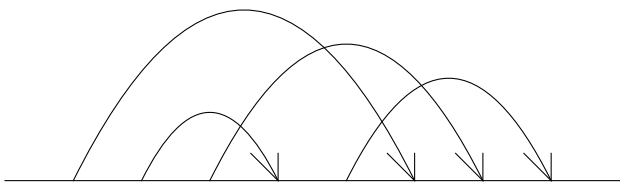
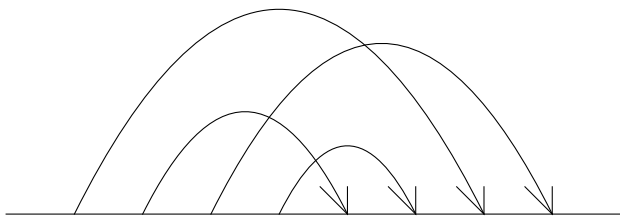
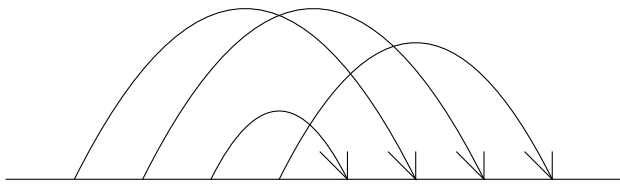
,

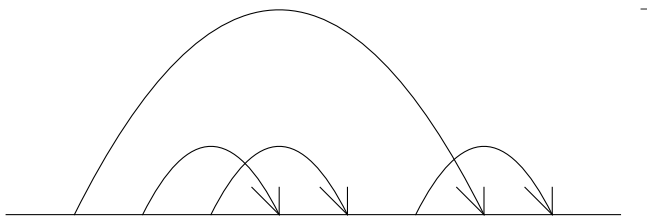
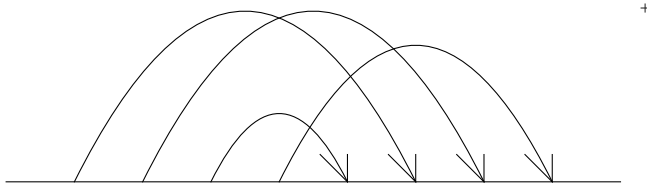
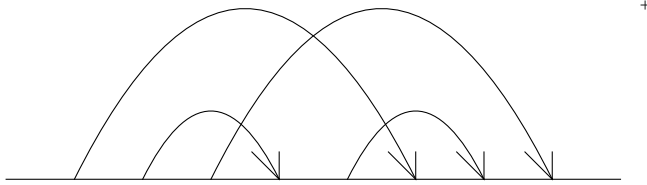


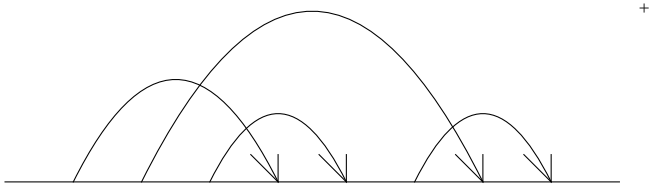
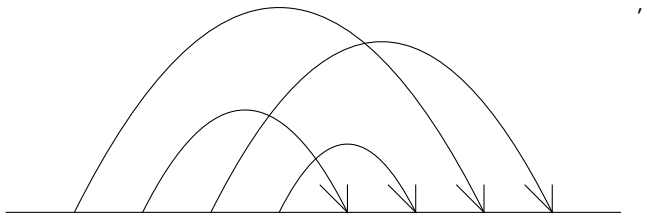
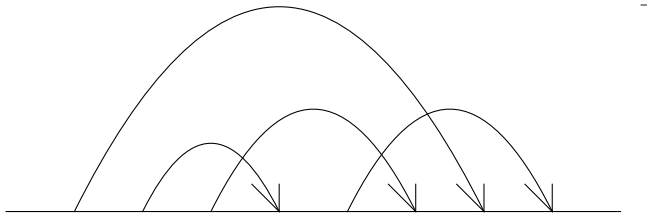
+

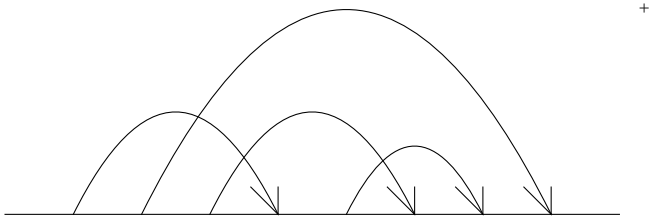
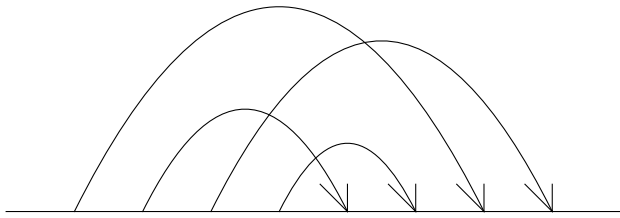


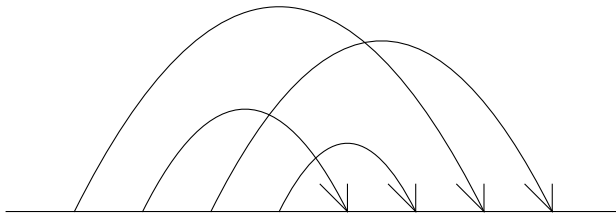
-



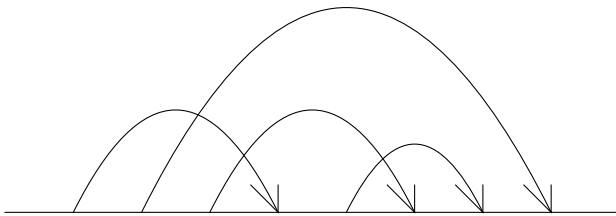




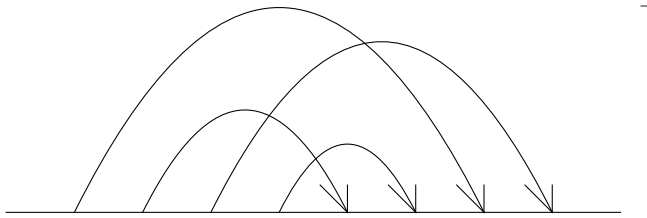
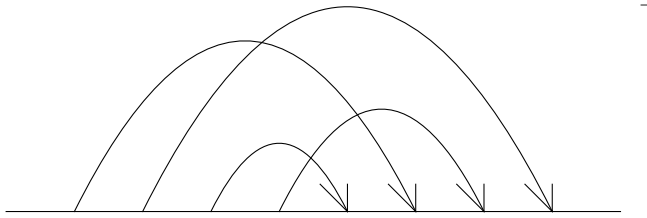


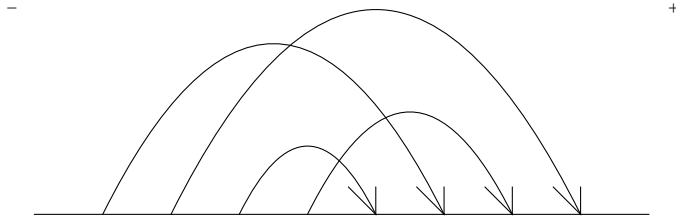
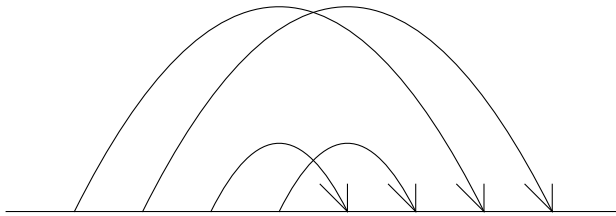


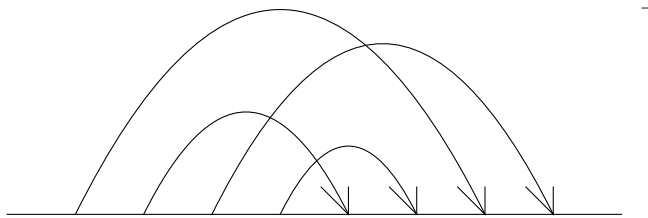
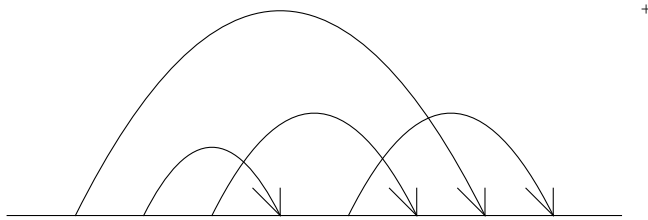
,

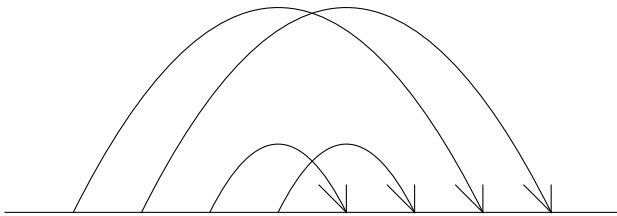
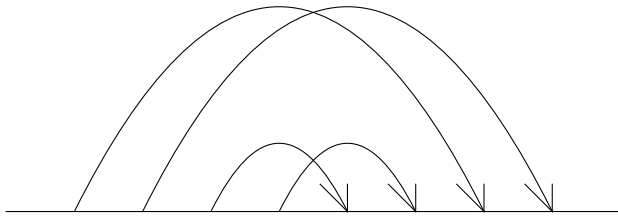


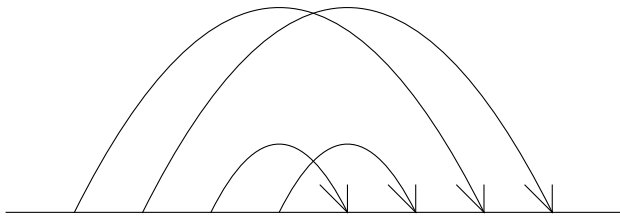
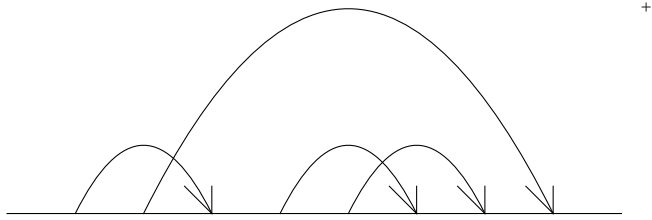
+

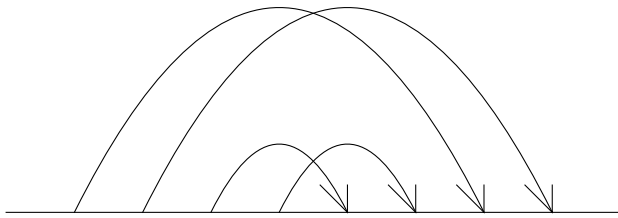
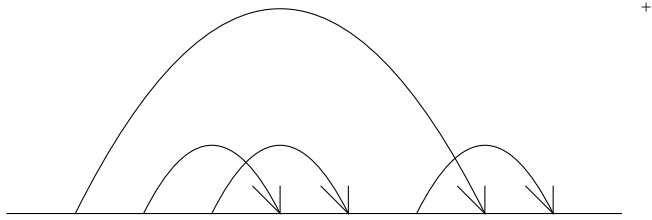


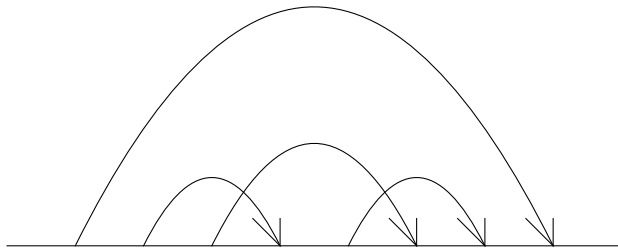




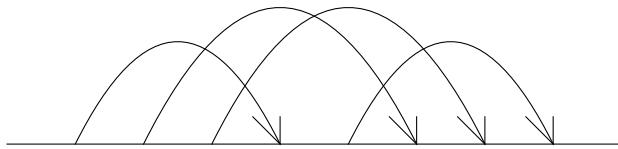




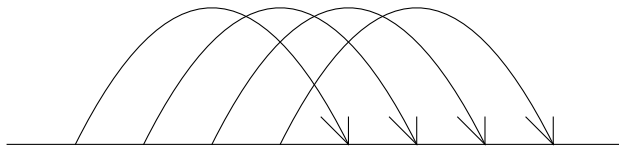




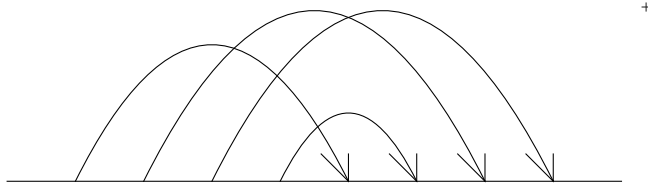
,



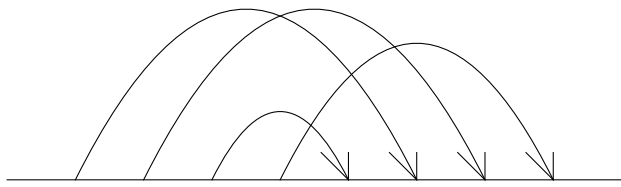
-



+

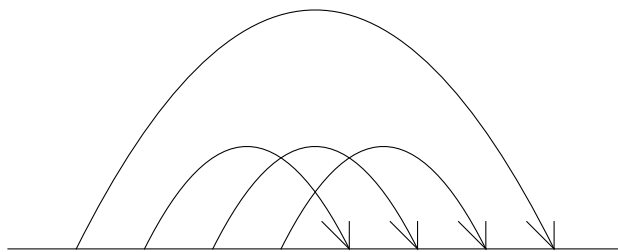


+

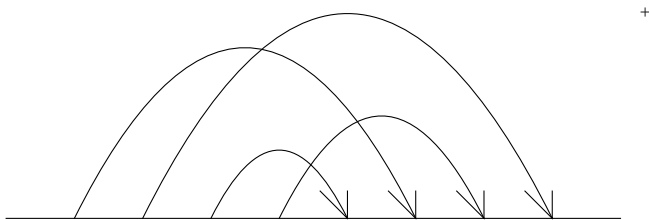
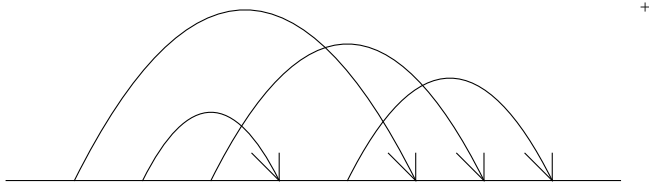
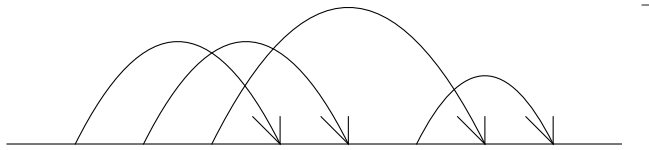


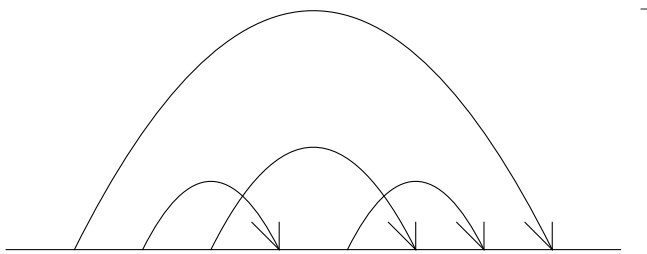
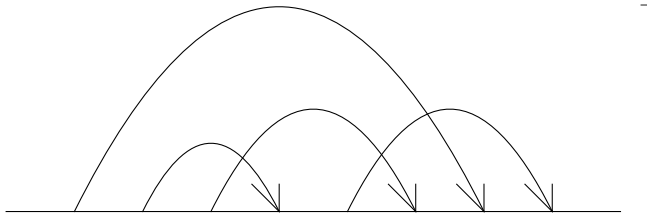
-

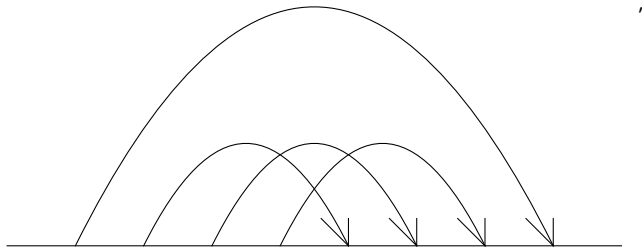
2



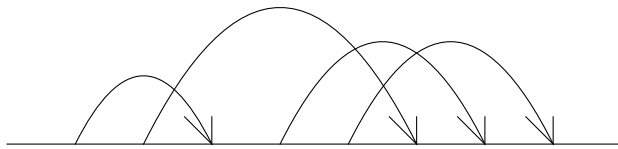
,



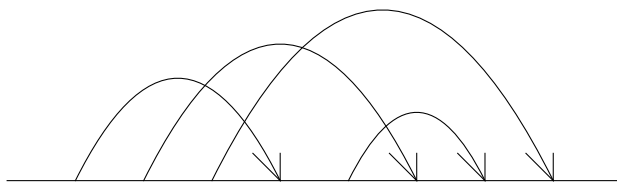




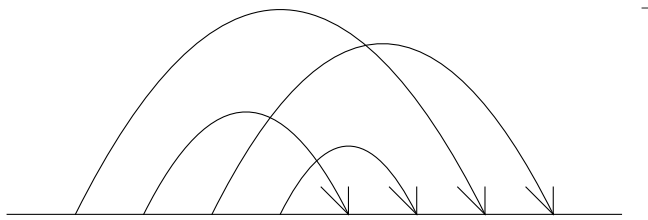
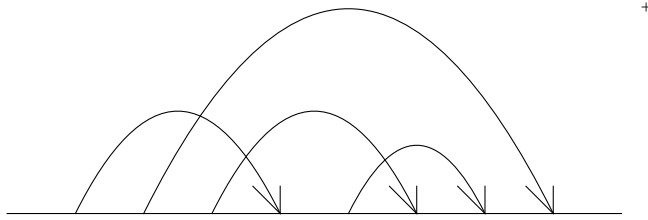
,

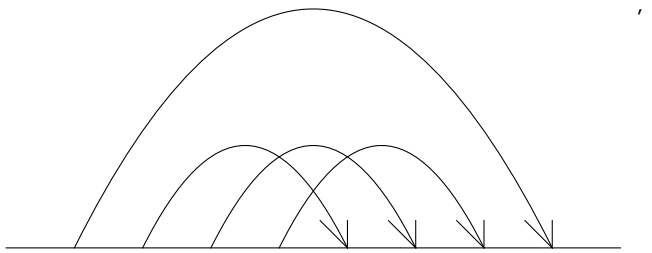
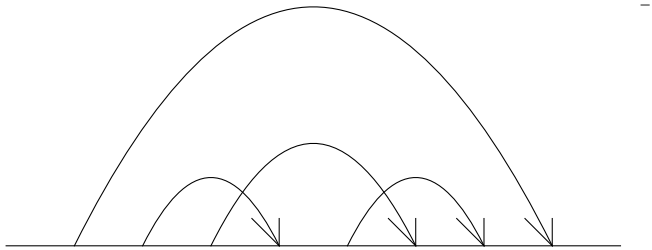


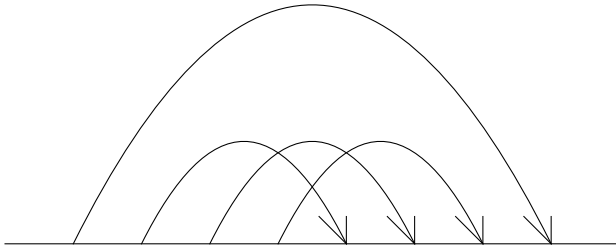
-

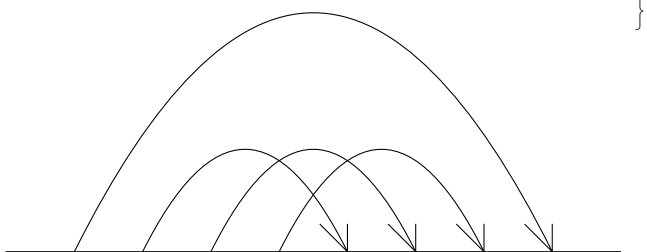
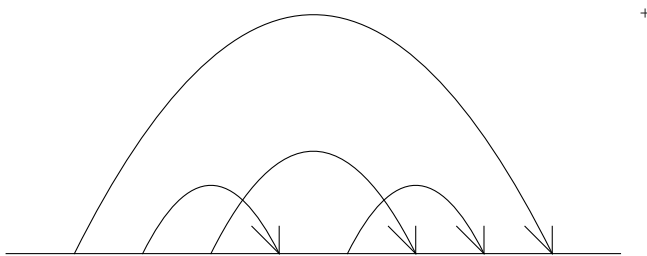
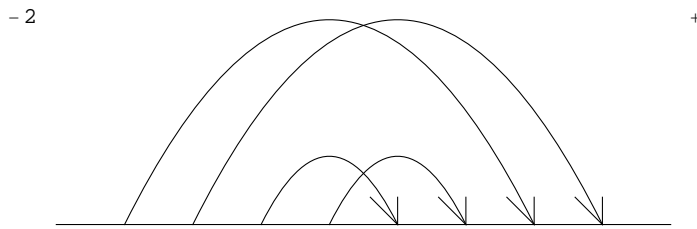


+









Runs

```
In[101]:= Analyze[{"R3b", "R2c", "R1", "Round", "Descending"}] // MatrixForm
```


Out[101]//MatrixForm=

$$\left(\begin{array}{cc} \text{QuotientSpace}[1, 2] \rightarrow 0 & \text{QuotientSpace}[1, 2] \rightarrow 0 \\ \text{QuotientSpace}[3, 5] \rightarrow 0 & \text{QuotientSpace}[3, 6] \rightarrow 0 \\ \text{QuotientSpace}[22, 47] \rightarrow 1 & \text{QuotientSpace}[27, 62] \rightarrow 0 \\ \text{QuotientSpace}[200, 490] \rightarrow 1 & \text{QuotientSpace}[236, 605] \rightarrow 0 \end{array} \right)$$

In[102]:= **Analyze**["R3b", "R2c", "Round", "Descending"] // **MatrixForm**

Out[102]//MatrixForm=

$$\left(\begin{array}{cc} \text{QuotientSpace}[2, 4] \rightarrow 0 & \text{QuotientSpace}[3, 5] \rightarrow 0 \\ \text{QuotientSpace}[9, 21] \rightarrow 0 & \text{QuotientSpace}[13, 26] \rightarrow 0 \\ \text{QuotientSpace}[63, 169] \rightarrow 1 & \text{QuotientSpace}[82, 204] \rightarrow 0 \\ \text{QuotientSpace}[570, 1693] \rightarrow 2 & \text{QuotientSpace}[694, 1981] \rightarrow 0 \end{array} \right)$$

In[103]:= **Analyze**["R3b", "R1d", "R1", "Round", "Descending"] // **MatrixForm**

Out[103]//MatrixForm=

$$\left(\begin{array}{cc} \text{QuotientSpace}[1, 2] \rightarrow 0 & \text{QuotientSpace}[1, 2] \rightarrow 0 \\ \text{QuotientSpace}[3, 5] \rightarrow 0 & \text{QuotientSpace}[3, 6] \rightarrow 0 \\ \text{QuotientSpace}[22, 47] \rightarrow 1 & \text{QuotientSpace}[27, 60] \rightarrow 0 \\ \text{QuotientSpace}[200, 490] \rightarrow 1 & \text{QuotientSpace}[236, 588] \rightarrow 0 \end{array} \right)$$

In[104]:= **Analyze**["R3b", "Round", "Descending"] // **MatrixForm**

Out[104]//MatrixForm=

$$\left(\begin{array}{cc} \text{QuotientSpace}[3, 3] \rightarrow 0 & \text{QuotientSpace}[4, 4] \rightarrow 0 \\ \text{QuotientSpace}[15, 24] \rightarrow 0 & \text{QuotientSpace}[19, 28] \rightarrow 0 \\ \text{QuotientSpace}[105, 208] \rightarrow 4 & \text{QuotientSpace}[124, 236] \rightarrow 1 \\ \text{QuotientSpace}[945, 2203] \rightarrow 8 & \text{QuotientSpace}[1069, 2439] \rightarrow 3 \end{array} \right)$$

In[105]:= **Analyze**["R2c", "R1", "Round", "Descending"] // **MatrixForm**

Out[105]//MatrixForm=

$$\left(\begin{array}{cc} \text{QuotientSpace}[1, 2] \rightarrow 0 & \text{QuotientSpace}[1, 2] \rightarrow 0 \\ \text{QuotientSpace}[3, 4] \rightarrow 0 & \text{QuotientSpace}[3, 4] \rightarrow 0 \\ \text{QuotientSpace}[22, 29] \rightarrow 2 & \text{QuotientSpace}[27, 37] \rightarrow 1 \\ \text{QuotientSpace}[200, 247] \rightarrow 12 & \text{QuotientSpace}[236, 301] \rightarrow 12 \end{array} \right)$$

In[106]:= **Analyze**["R2c", "Round", "Descending"] // **MatrixForm**

Out[106]//MatrixForm=

$$\left(\begin{array}{cc} \text{QuotientSpace}[2, 3] \rightarrow 0 & \text{QuotientSpace}[3, 4] \rightarrow 0 \\ \text{QuotientSpace}[9, 11] \rightarrow 1 & \text{QuotientSpace}[13, 15] \rightarrow 1 \\ \text{QuotientSpace}[63, 74] \rightarrow 8 & \text{QuotientSpace}[82, 98] \rightarrow 7 \\ \text{QuotientSpace}[570, 643] \rightarrow 47 & \text{QuotientSpace}[694, 815] \rightarrow 49 \end{array} \right)$$