

## Program and Testing

### Drawing Arrow Diagrams

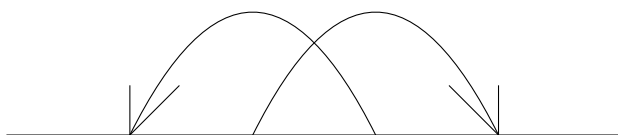
```

In[1]:= 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}]]
      },
    Cut[i_] => Line[{{i, 0.3}, {i, -0.3}}]
  ]
];

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

```

Out[4]=



### Generating Arrow Diagrams

```

In[5]:= 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]}]
]];

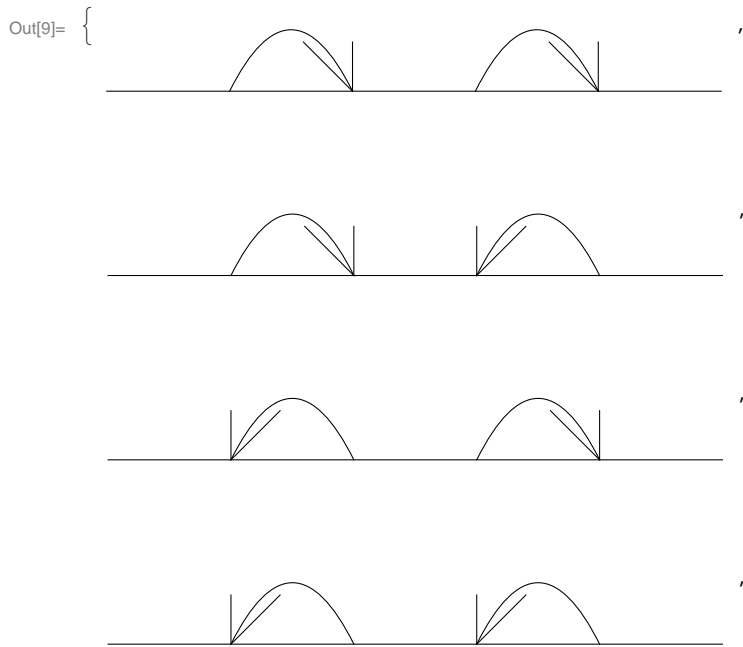
```

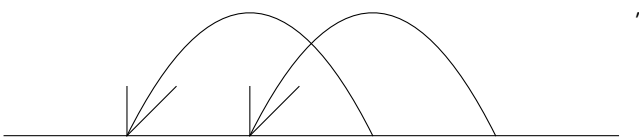
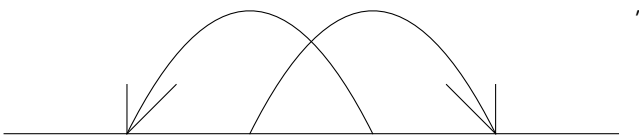
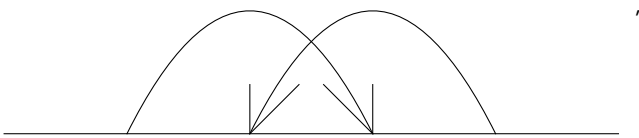
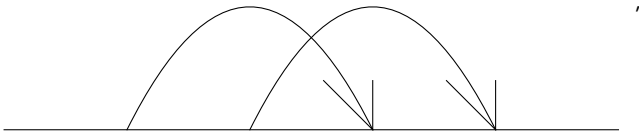
```
In[7]:= Diagrams[spec_] := Flatten[
  Place[
    #,
    Range[
      spec /. {Ar -> 2, Cut -> 1, HTC -> 3, R3b -> 3, TC -> 3, R2c -> 2, R1a -> 1, R1d -> 1}
    ] & /@ Permutations[Flatten[
      {spec} /. Plus -> List /. k_Integer * s_Symbol -> Table[s, {k}]
    ]
  ]
]
```

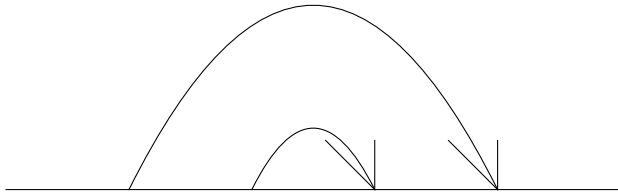
```
In[8]:= Diagrams[2 Ar]
```

```
Out[8]= {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]]}
```

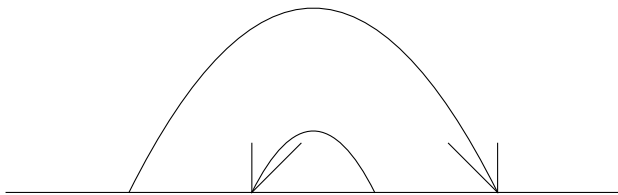
```
In[9]:= Draw[Diagrams[2 Ar]]
```



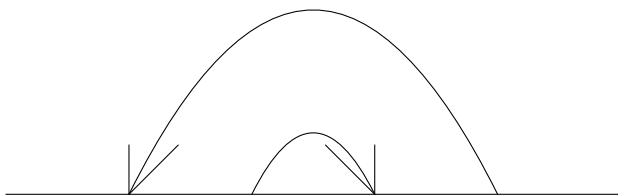




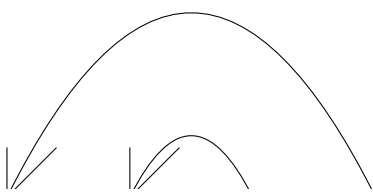
,



,



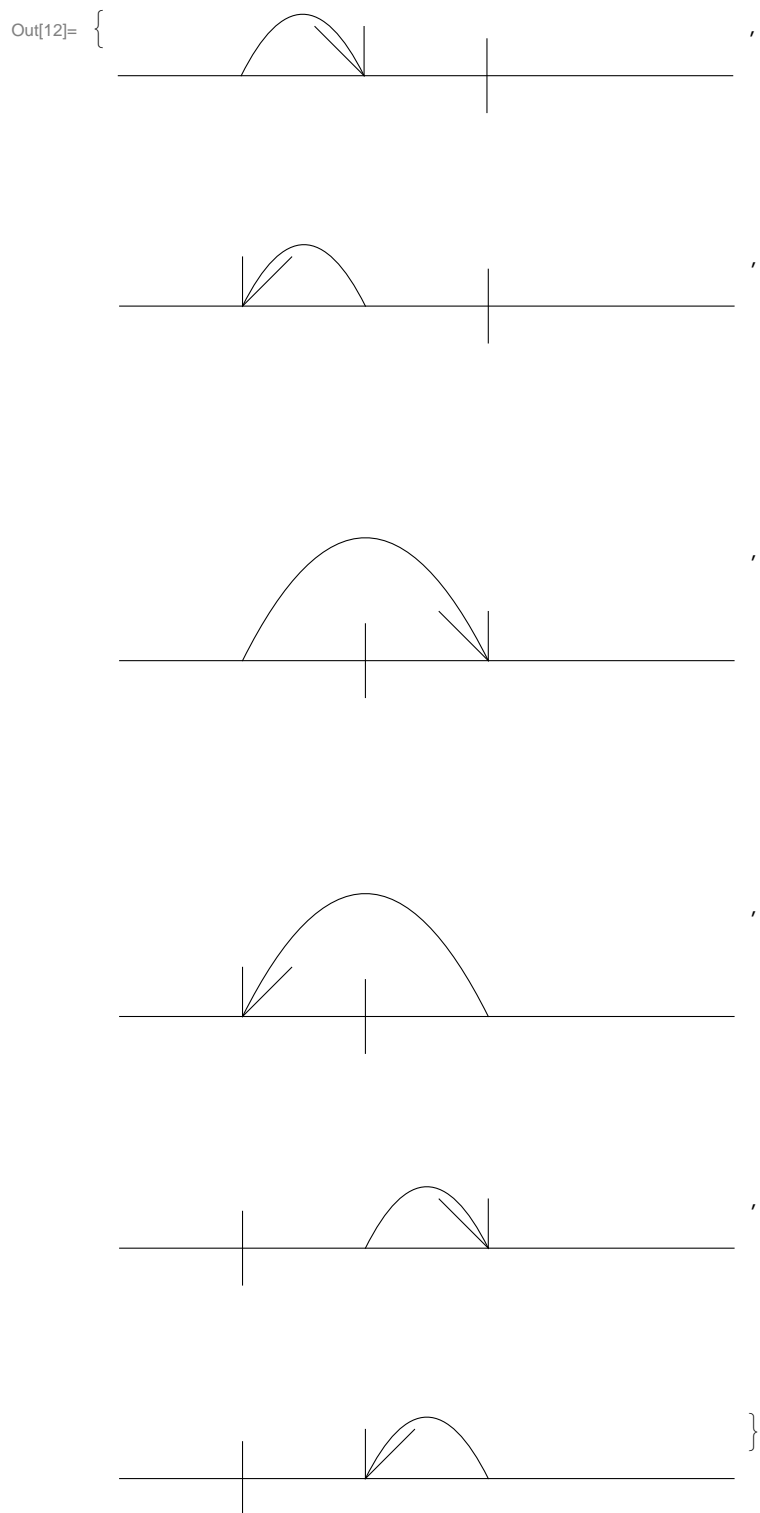
,



}

```
In[10]:= Place[{Cut}, {i_}] := {Diag[Cut[i]]};  
Place[{Cut, objs__}, {i_, rest__}] := Flatten[  
  Outer[Join,  
    Place[{Cut}, {i}],  
    Place[{objs}, {rest}]  
  ]  
];
```

```
In[12]:= Diagrams [Cut + 1 Ar] // Draw
```



## The "Round" Case

```

In[13]:= DiagramRotateLeft [d_Diag] := Module [
  {labels = Union @@ (List @@ Apply[List, d, {1}])},
  d /. Thread[Rule[labels, RotateLeft[labels]]]
];

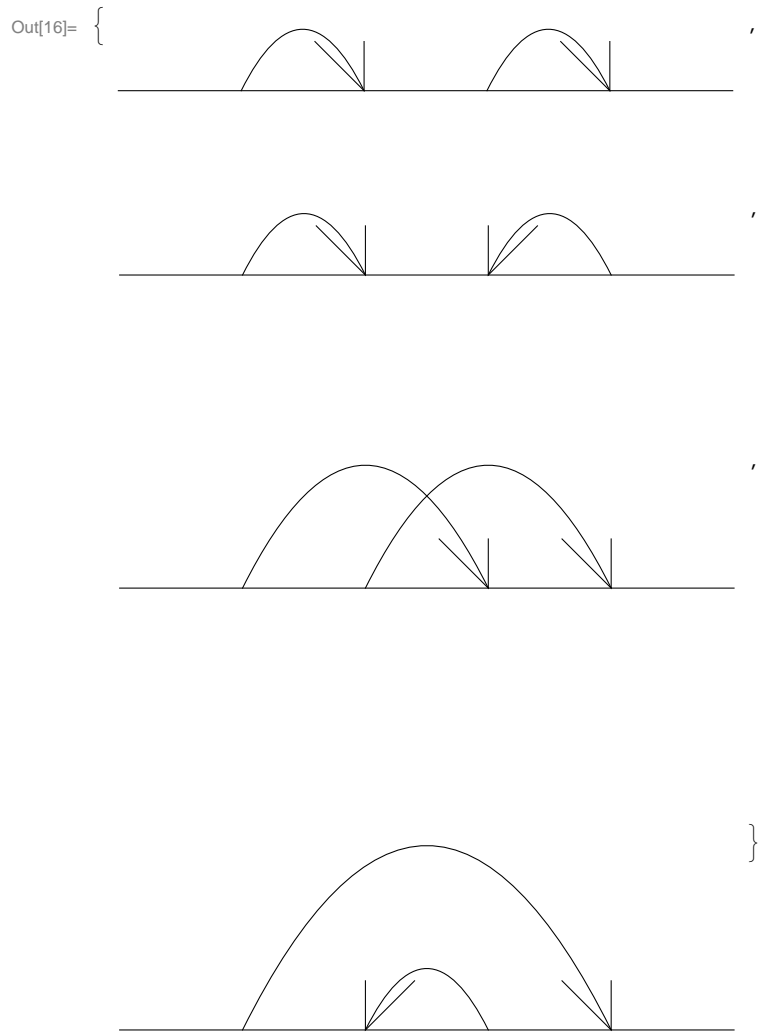
DiagramRotateToMinimal [expr_] := expr /. d_Diag => Module [
  {bd = d, rd = DiagramRotateLeft [d]},
  While[rd != d,
    bd = First[Sort[{bd, rd}]];
    rd = DiagramRotateLeft [rd]];
  bd
];

In[15]:= Union[DiagramRotateToMinimal [Diagrams [2 Ar]]]

Out[15]= {Diag[Ar [1, 2], Ar [3, 4]], Diag[Ar [1, 2], Ar [4, 3]],
  Diag[Ar [1, 3], Ar [2, 4]], Diag[Ar [1, 4], Ar [3, 2]]}

```

```
In[16]:= Draw[Union[DiagramRotateToMinimal [Diagrams [2 Ar]]]]
```





## Generating Relators

```

In[20]:= Place[{r : (HTC | 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]];

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}];

Place[{HTC}, {i_, j_, k_}] :=
  Permutations[{i, j, k}] /. {i1_, j1_, k1_} => Diag[HTC[i1, j1, k1]];

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]}]
}];

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}]
  ]
];

In[30]:= Take[Diagrams[R3b + 1 Ar], 5]

Out[30]= {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[31]:= Take[Diagrams[HTC + 1 Ar], 5]

Out[31]= {Diag[Ar[1, 2], HTC[3, 4, 5]], Diag[Ar[1, 2], HTC[3, 5, 4]],
  Diag[Ar[1, 2], HTC[4, 3, 5]], Diag[Ar[1, 2], HTC[4, 5, 3]], Diag[Ar[1, 2], HTC[5, 3, 4]]}

In[32]:= {Diagrams[R1d + Ar], Diagrams[R1a + Ar]}

Out[32]= {{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]]}}

```

```
In[36]:= Take[Diagrams[Cut + Ar + HTC], 5]
```

```
Out[36]= {Diag[Ar[1, 2], Cut[3], HTC[4, 5, 6]],  
  Diag[Ar[1, 2], Cut[3], HTC[4, 6, 5]], Diag[Ar[1, 2], Cut[3], HTC[5, 4, 6]],  
  Diag[Ar[1, 2], Cut[3], HTC[5, 6, 4]], Diag[Ar[1, 2], Cut[3], HTC[6, 4, 5]]}
```

## Generating Relations

```

In[37]:= NormalizeDiag[diag_Diag] := Module[
  {indices = Union@@(List@@diag /. Ar | Cut → 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
    ]]]],
  ]
];

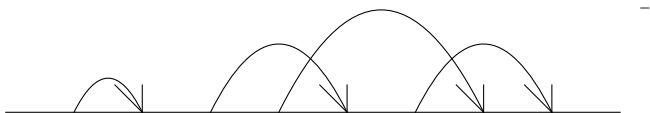
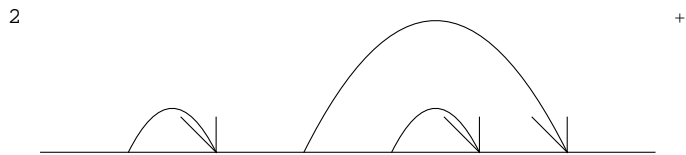
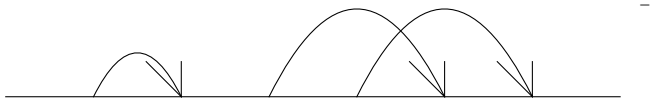
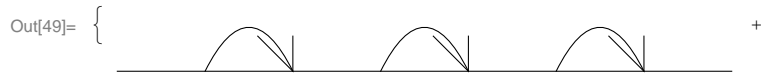
```

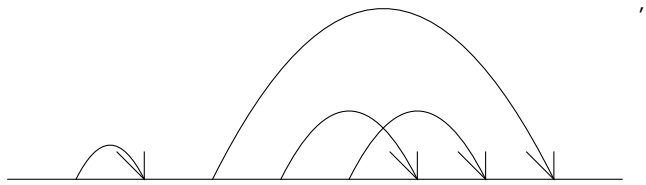
```

      {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]]
);
R[Diag[lft___, HTC[i_, j_, k_], rgt___]] := (
  +NormalizeDiag[Diag[lft, Ar[i, k], Ar[k + 0.5, j], rgt]]
  - NormalizeDiag[Diag[lft, Ar[i, k + 0.5], Ar[k, j], rgt]]
);
ContainsShortArrow[diag_Diag] := (1 == Min[List@@diag /. Ar[i_, j_] => Abs[i - j]]);
DescendingQ[diag_Diag] := And @@ (OrderedQ /@ diag);

```

```
In[49]:= Draw[R /@ Take[Diagrams[R3b + 1 Ar], 2]]
```

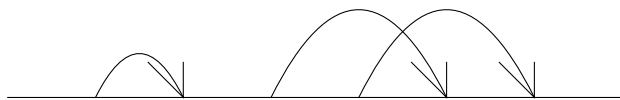




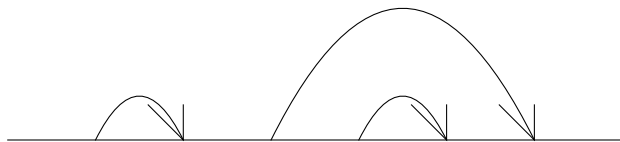
'



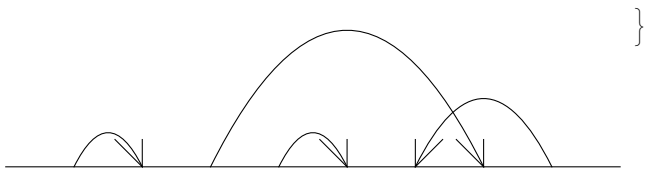
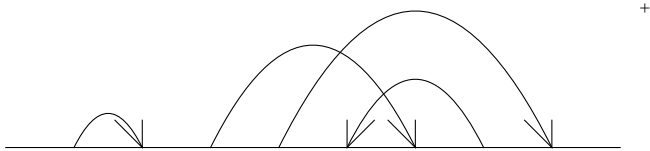
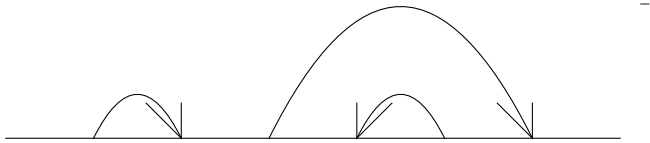
-



+

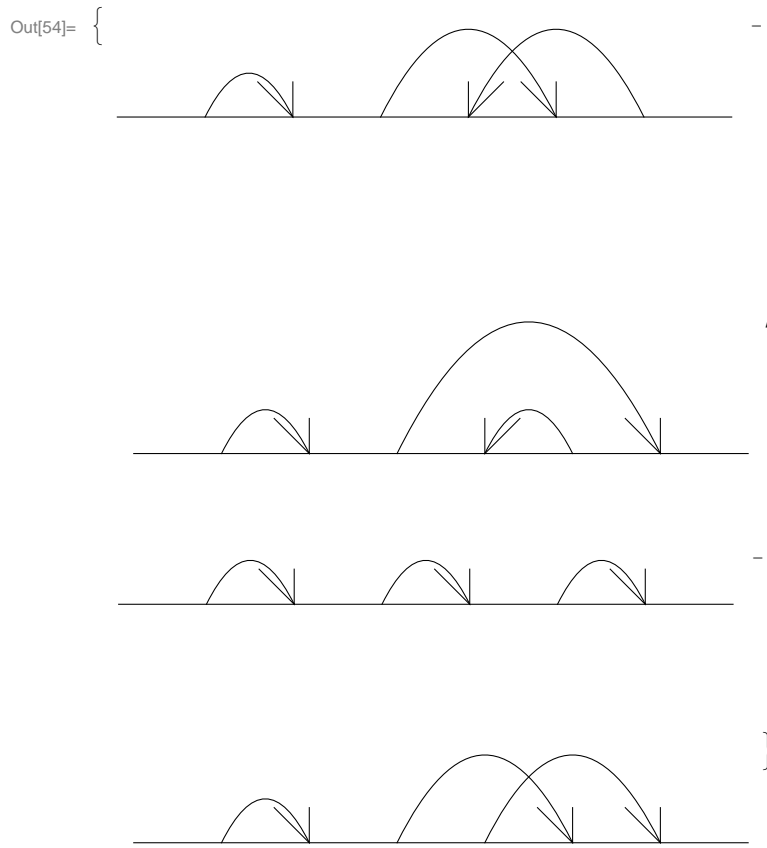


-

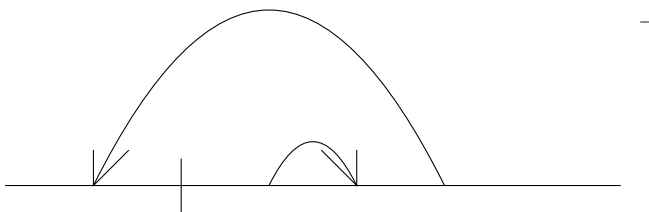
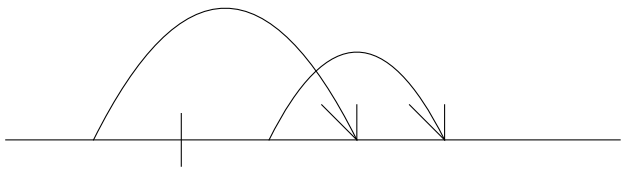
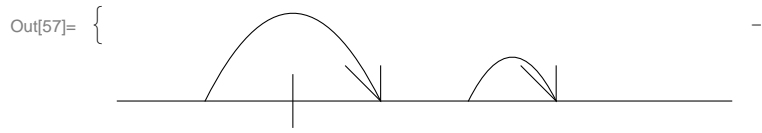


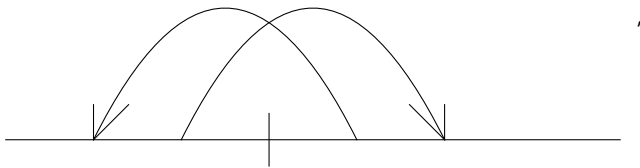
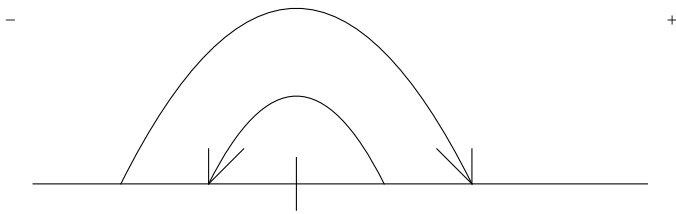
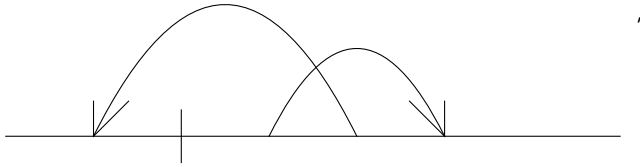
```
In[54]:= Draw[R /@ Take[Diagrams[HTC + 1 Ar], 2]]
```

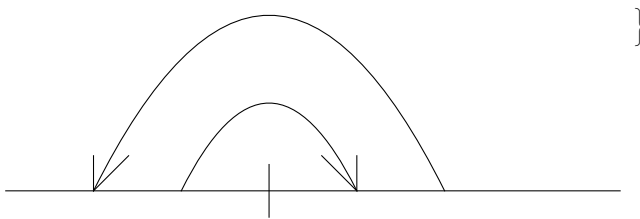
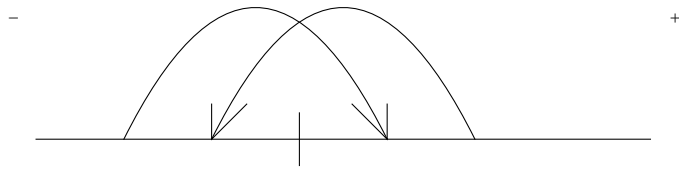
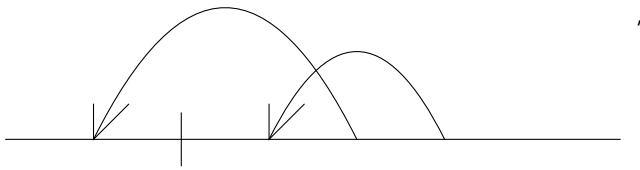
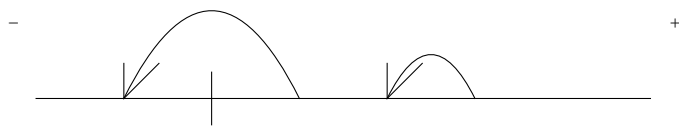




```
In[57]:= Draw[R /@ Take[Diagrams[HTC + Cut], -5]]
```







Producing "Quotient Spaces"

```

In[64]:= 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" => (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}]
),
"HTC" => (
  R /@ Join @@ Table[Diagrams [HTC + k Ar], {k, If[gr, m - 2, 0], m - 2}]
)
});
relytypes /. {
  "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]
]

In[72]:= PSpace[m_, 2] /; m < 2 := QuotientSpace[Diagrams[m Ar], {}];
PSpace[m_, 2, reotypes_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[Cut + k Ar], {k, If[gr, m, 1], m}];
filter = Identity;
reotypes /. {
  "R1" => (
    diags = Select[diags, !ContainsShortArrow[#] &]
  ),
  "Descending" => (
    filter = Select[#, DescendingQ] &;
    diags = Select[diags, DescendingQ]
  )
};
If[debug, Print[Date[], ": Computed ", Length[diags], " diagrams..."]];
rels = Join@@(reotypes /. {
  "R1" → {},
  "Descending" → {},
  "Round" => (diags /.
    d_Diag => (NormalizeDiag[d /. {1 → 2 Length[d], i_Integer => i - 1}] - d)
  ),
  "R3b" => (
    (R /@ Join@@Table[Diagrams[Cut + R3b + k Ar] // filter,
      {k, If[gr, m - 2, 0], m - 2}]) /. diag_Diag /; Length[diag] > m + 1 => 0
  ),
  "RWrong1" => (
    (RWrong1 /@ Join@@Table[Diagrams[Cut + R3b + k Ar], {k, If[gr, m - 2, 0], m - 2}]) /.
      diag_Diag /; Length[diag] > m + 1 => 0
  )
),

```

```

"RWrong2" => (
  (RWrong2 /@ Join@@ Table[Diagrams[Cut + R3b + k Ar], {k, If[gr, m - 2, 0], m - 2}]) /.
  diag_Diag /; Length[diag] > m + 1 => 0
),
"R3bMinus" => (
  RMinus[m, #] & /@
  Join@@ Table[Diagrams[Cut + R3b + k Ar] // filter, {k, If[gr, m - 2, 0], m - 2}]
),
"R2c" => (
  R[m, #] & /@
  Join@@ Table[Diagrams[Cut + R2c + k Ar] // filter, {k, If[gr, m - 2, 0], m - 2}]
),
"R1d" => (
  R[m, #] & /@
  Join@@ Table[Diagrams[Cut + R1d + k Ar] // filter, {k, If[gr, m - 2, 0], m - 2}]
),
"R1a" => (
  R[m, #] & /@
  Join@@ Table[Diagrams[Cut + R1a + k Ar] // filter, {k, If[gr, m - 2, 0], m - 2}]
),
"TC" => (
  R /@ Join@@ Table[Diagrams[Cut + TC + k Ar], {k, If[gr, m - 2, 0], m - 2}]
),
"HTC" => (
  R /@ Join@@ Table[Diagrams[Cut + HTC + 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

```

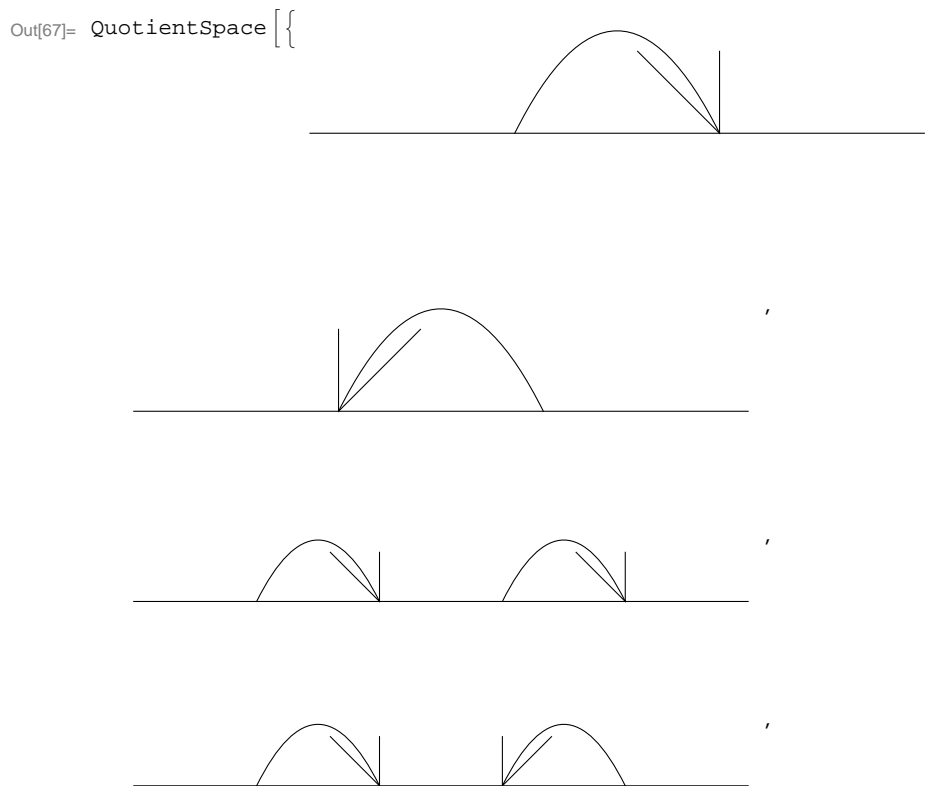


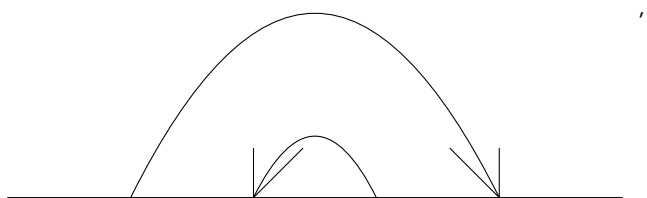
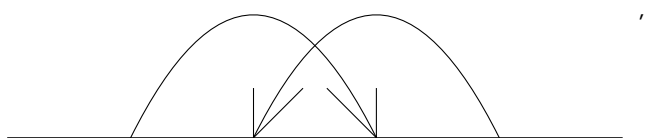
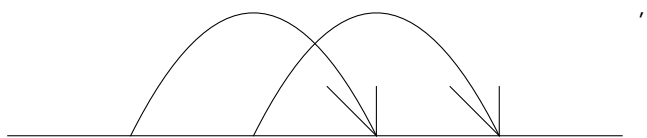
```

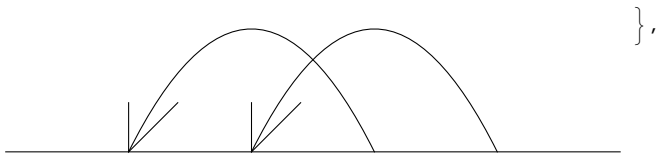
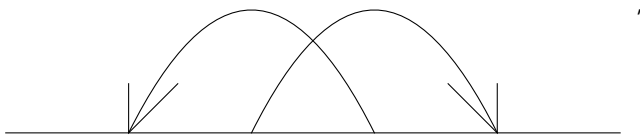
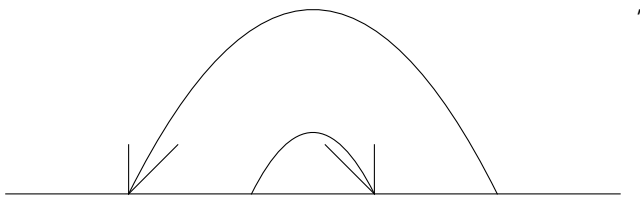
];
redrule = Dispatch[Flatten[Table[
  class = equivclasses[[i]];
  (# -> First[class]) & /@ Rest[class],
  {i, Length[equivclasses]}
]]];
diags = Union[diags /. redrule];
rels = Union[rels /. redrule];
];
QuotientSpace[diags, rels]
]

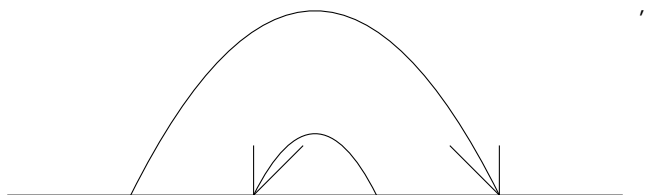
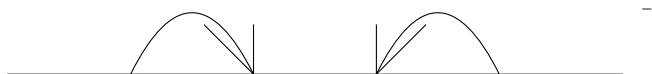
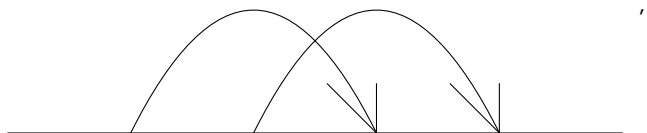
```

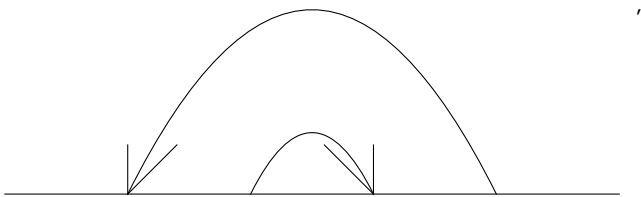
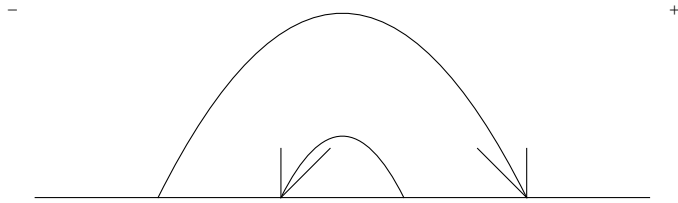
In[67]:= Draw[PSpace[2, {"R3b", "R2c"}]]

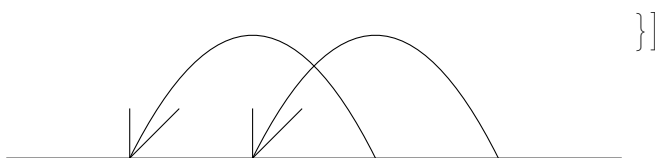
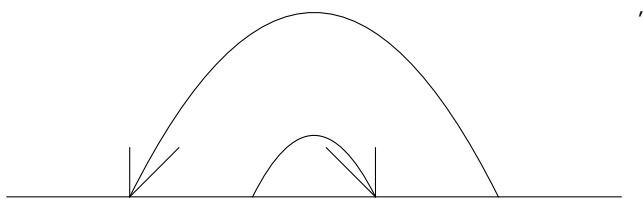






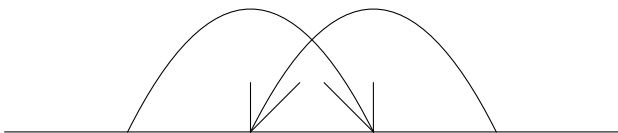
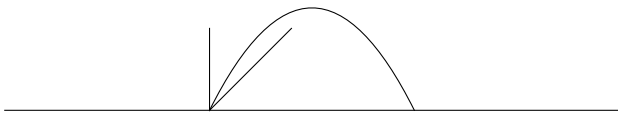
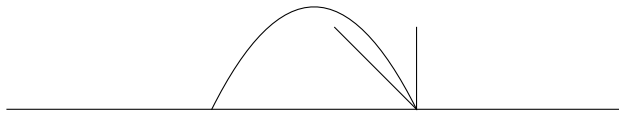


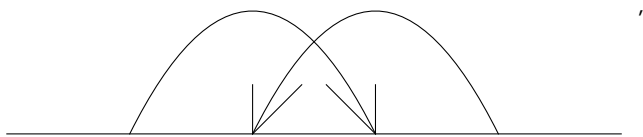
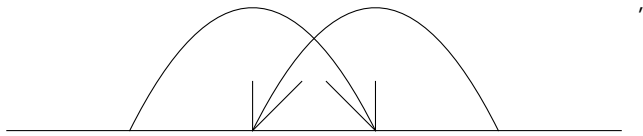
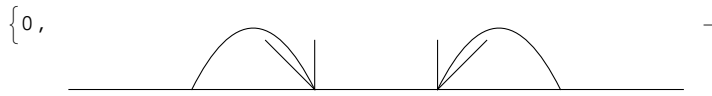




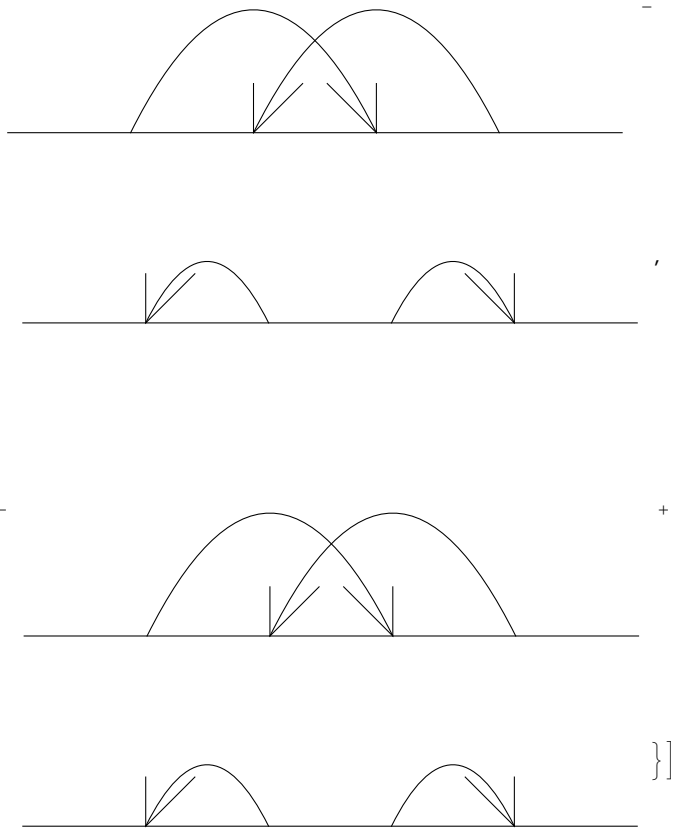
In[68]:= Draw[PSpace[2, {"R3b", "R2c", "HTC"}]]

Out[68]= QuotientSpace { {




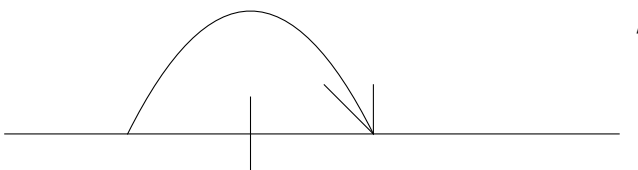




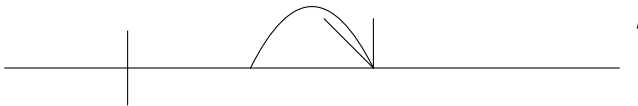


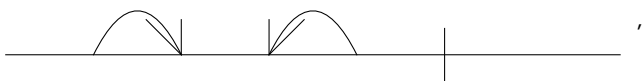
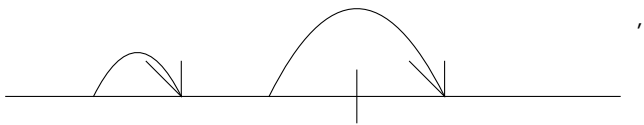
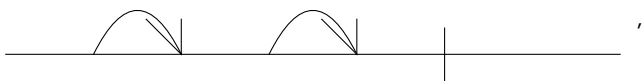
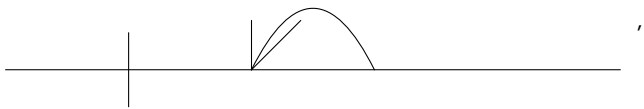
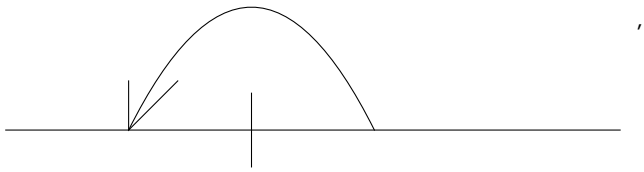
```
In[74]:= Draw[PSpace[2, 2, {"R3b"}]]
```

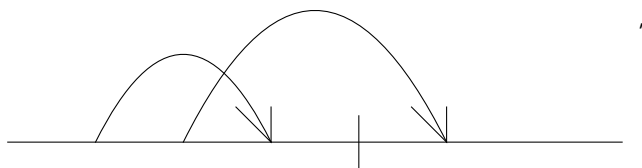
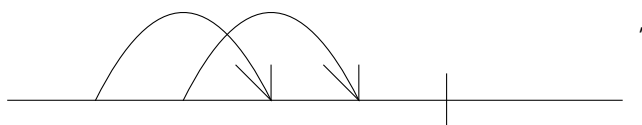
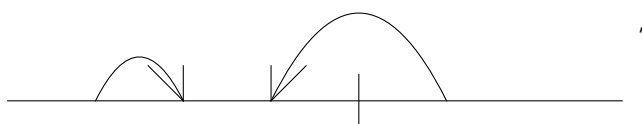
Out[74]= QuotientSpace [ {  ,

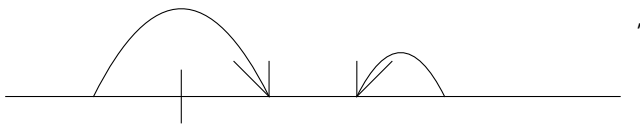
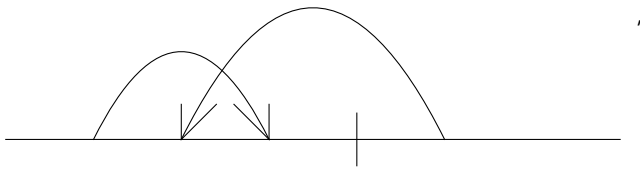
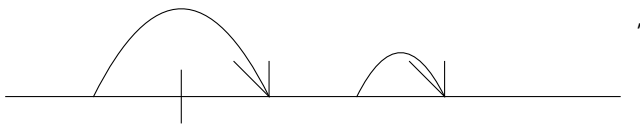
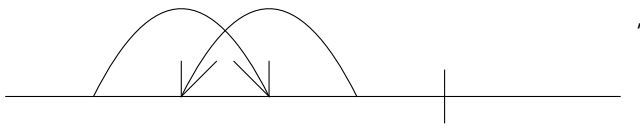
 ,

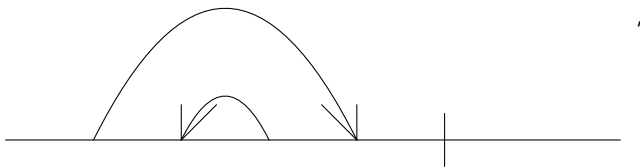
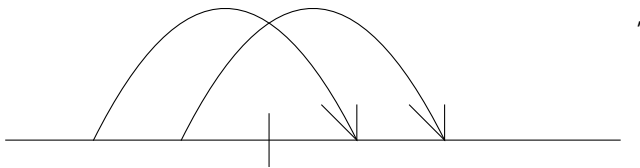
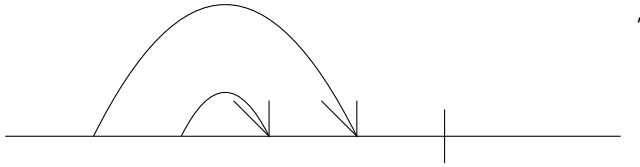
 ,

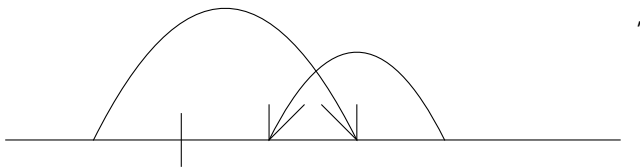
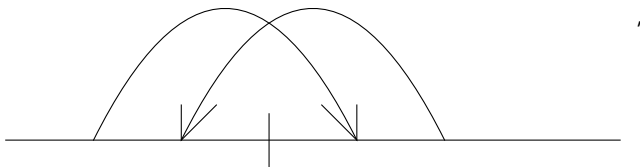
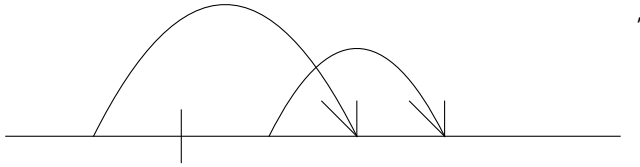
 ,

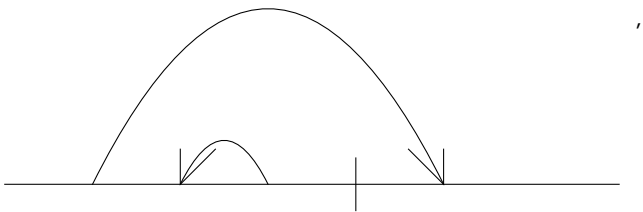
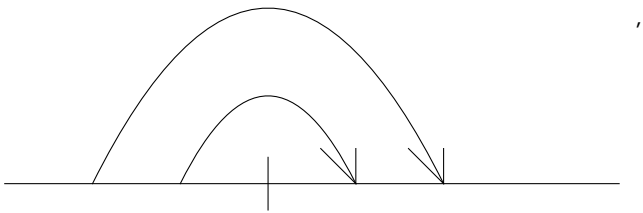
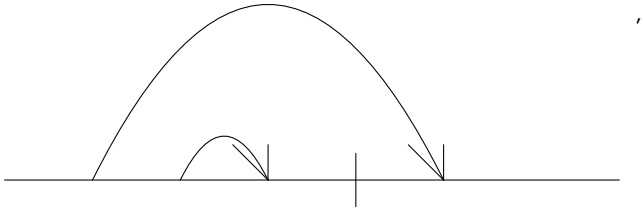




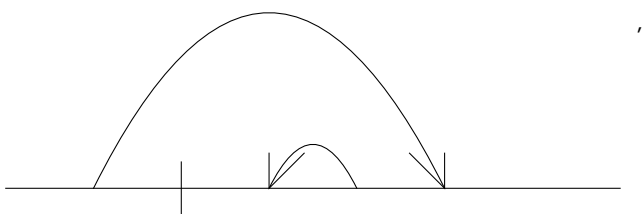
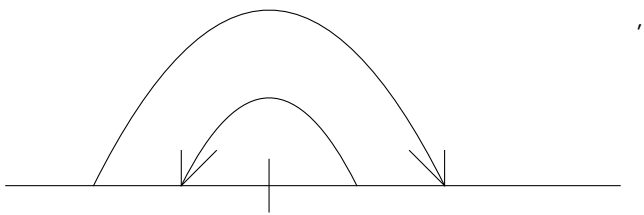
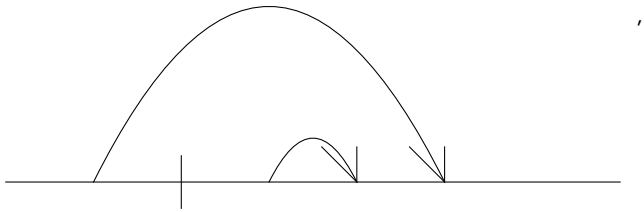


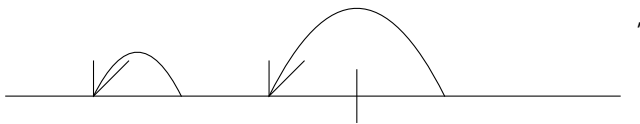
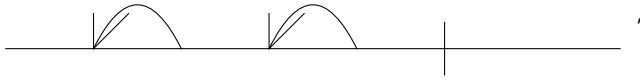
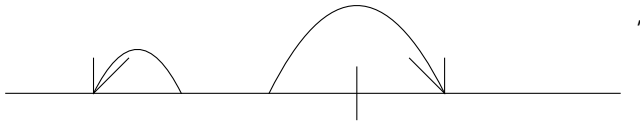
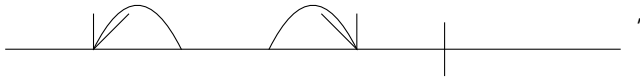


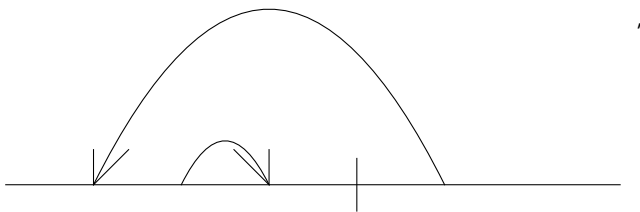
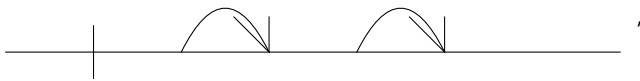
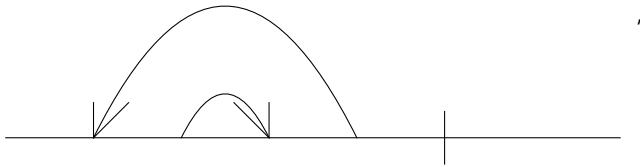


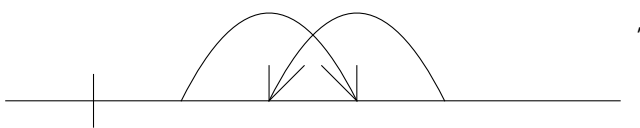
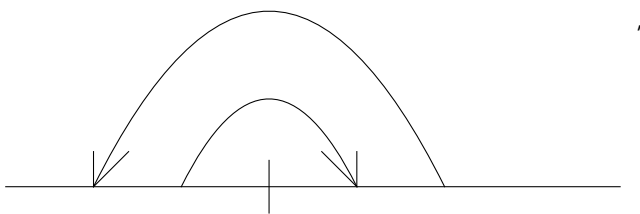
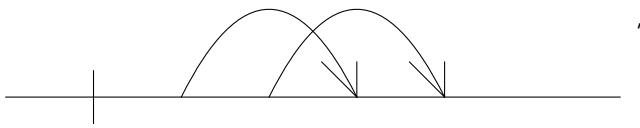
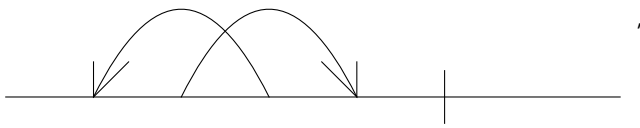


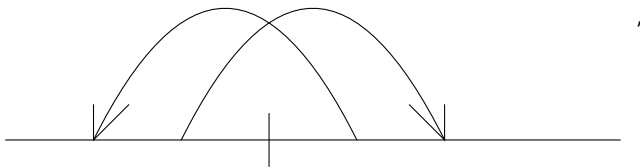
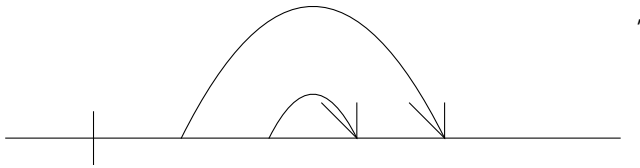
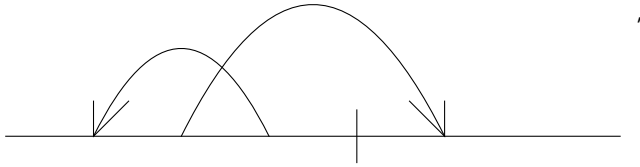


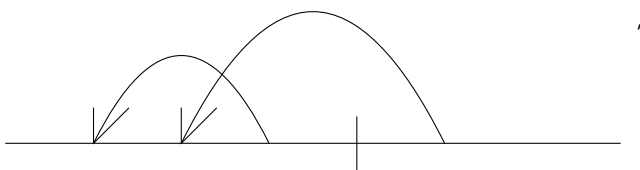
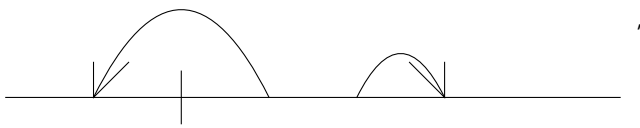
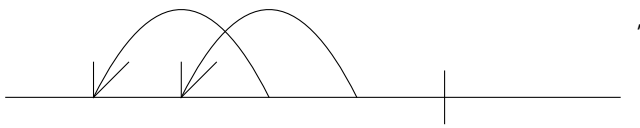
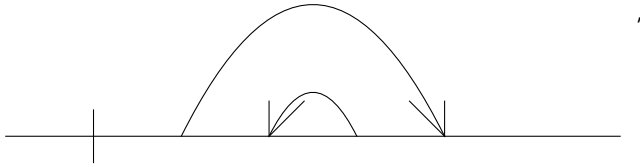


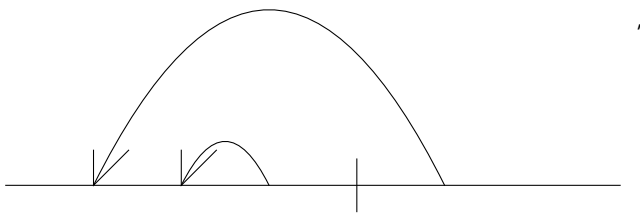
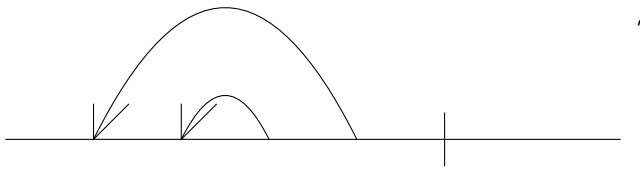
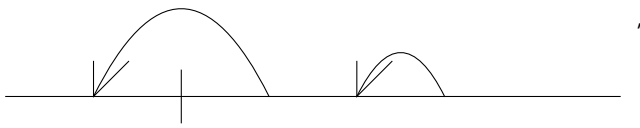


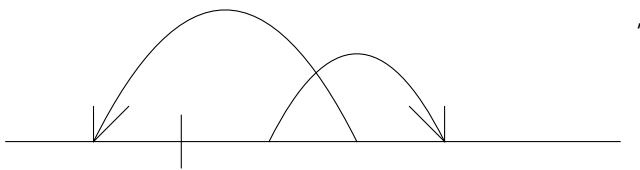
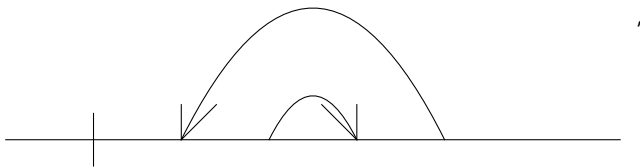
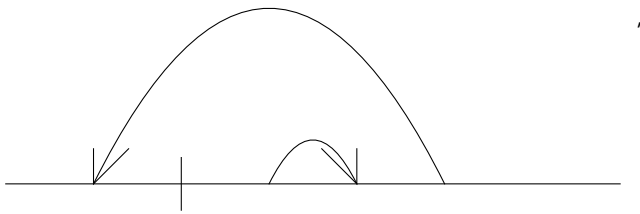




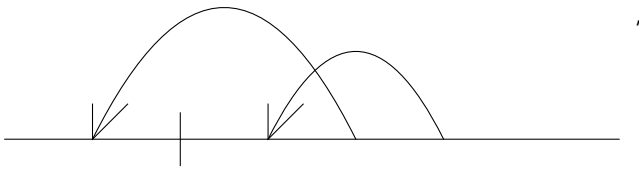
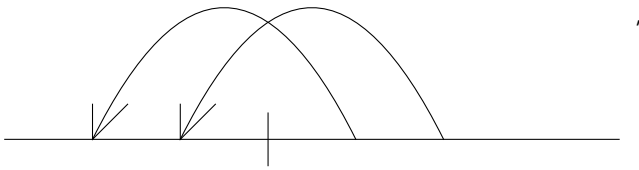
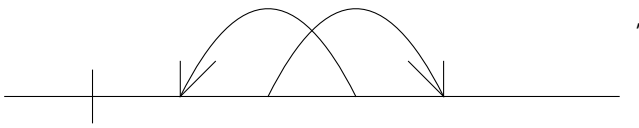


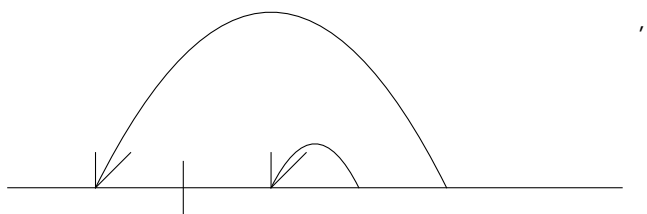
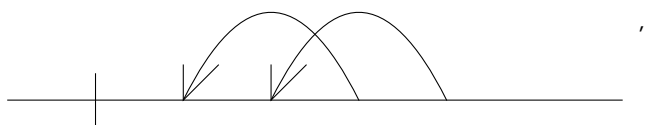
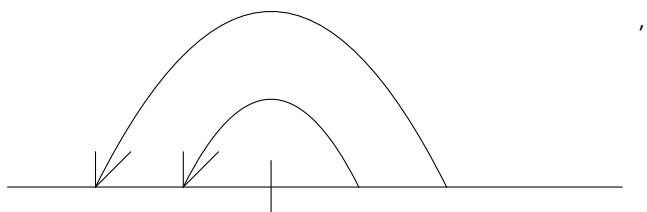


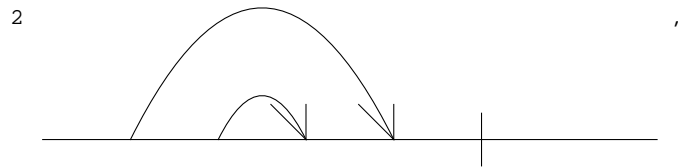
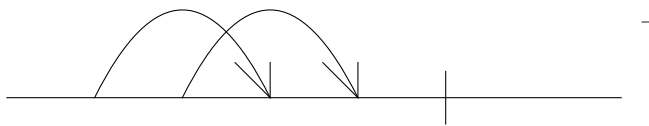
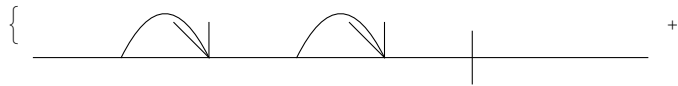
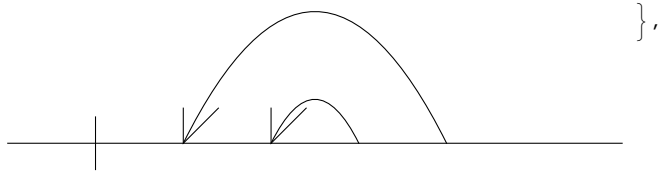


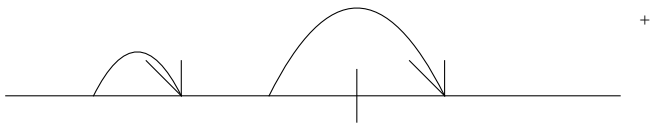
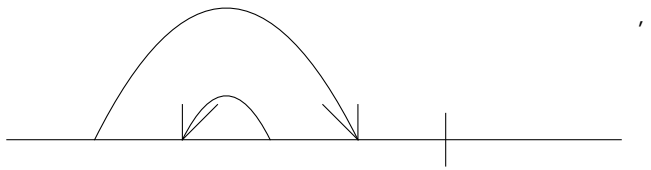
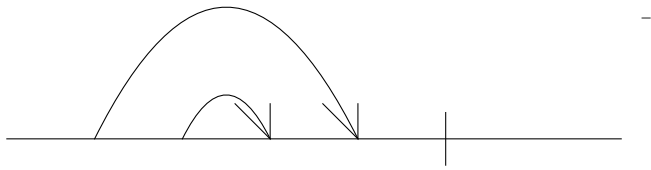
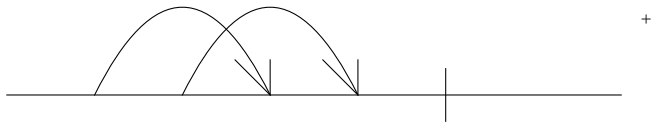


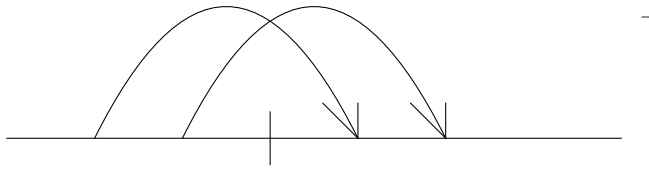




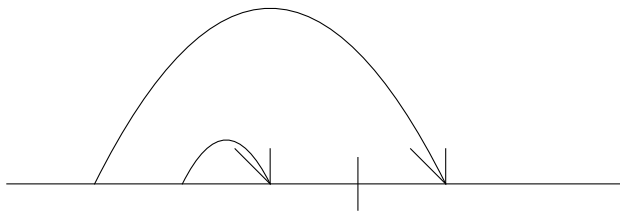




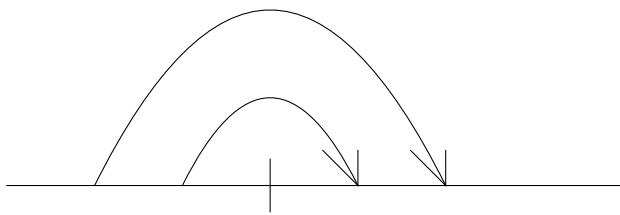




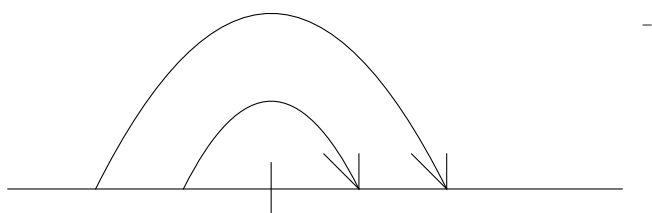
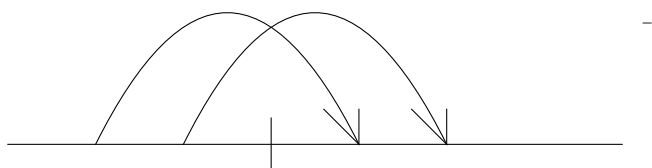
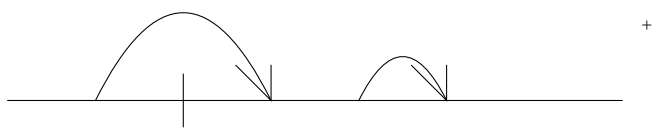
-

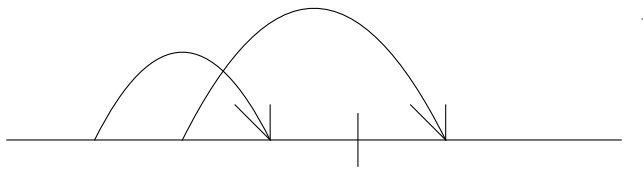
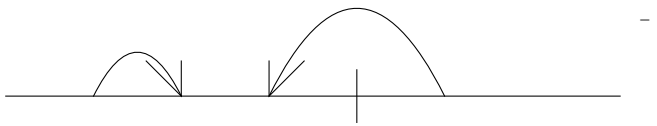
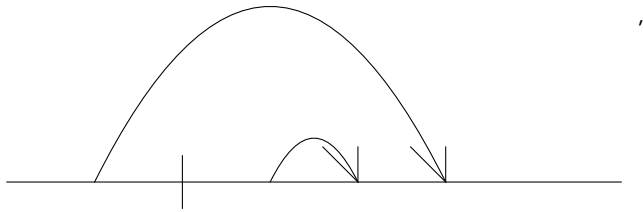


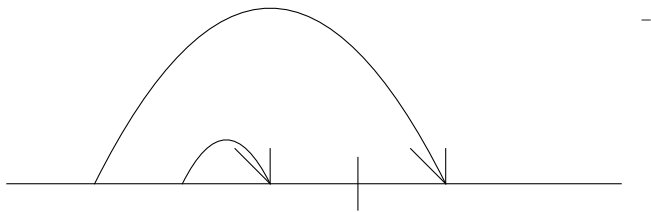
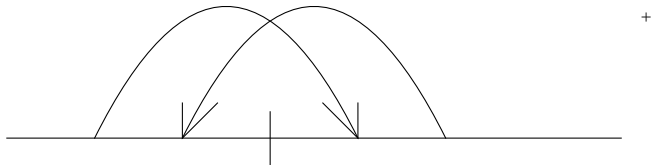
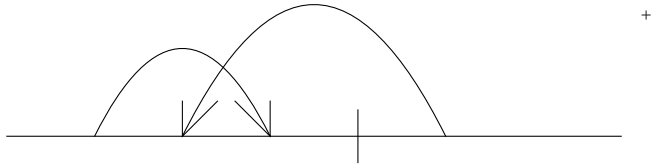
-



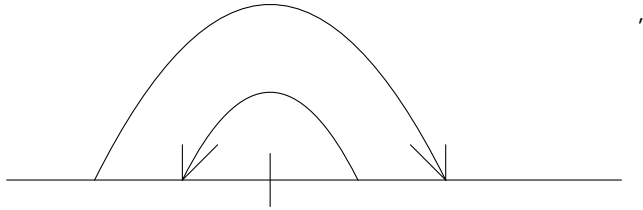
,



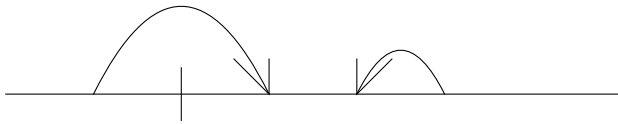




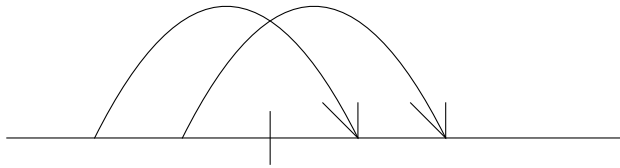




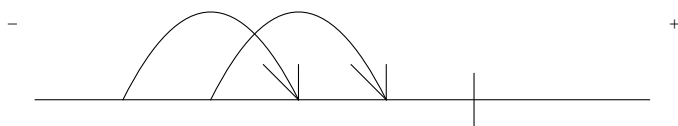
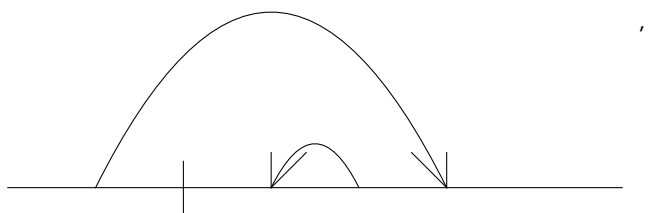
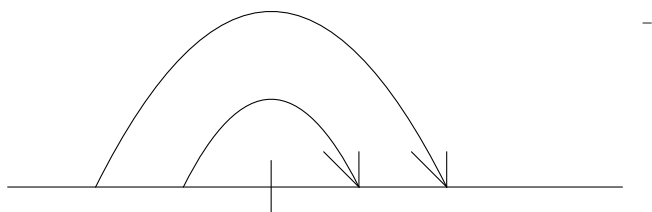
,

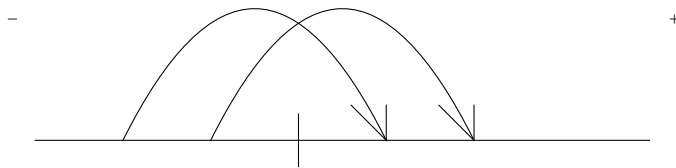
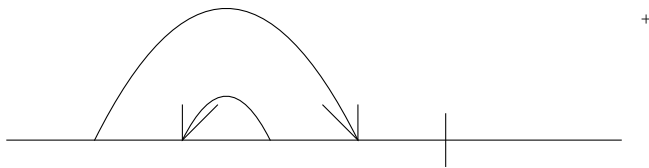
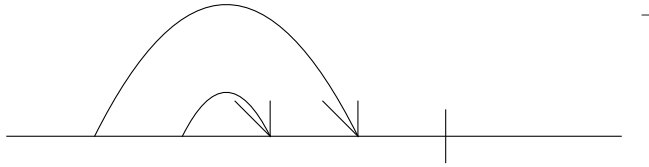


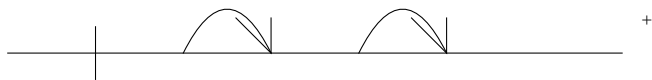
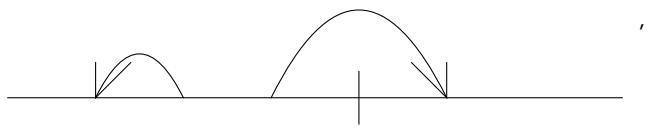
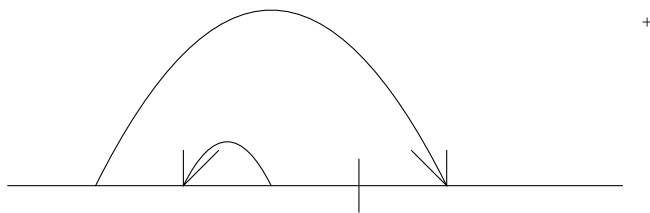
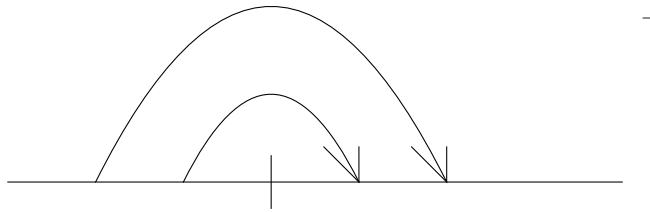
-

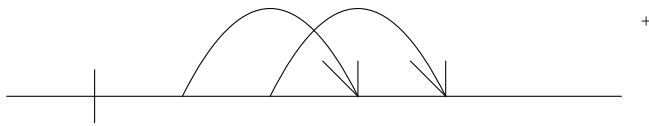
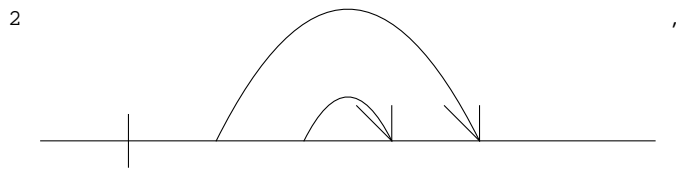
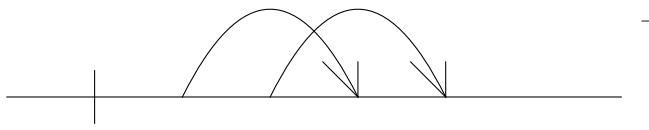


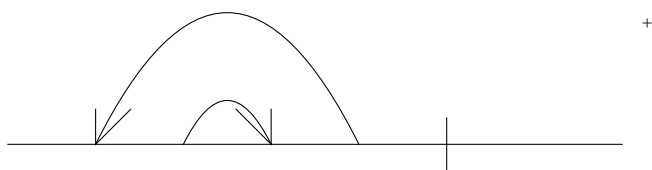
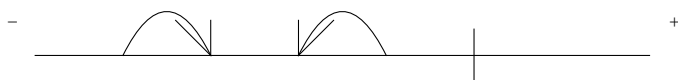
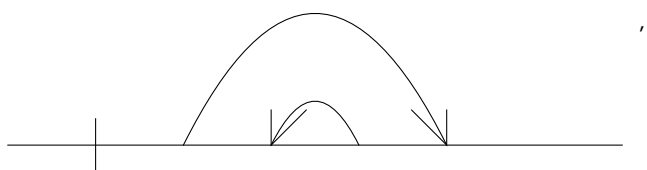
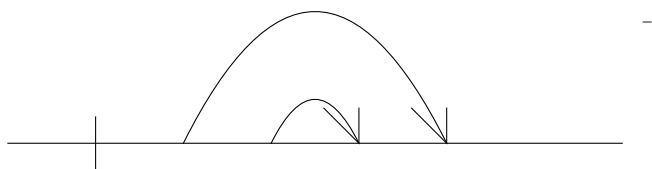
+

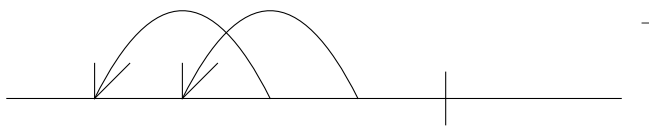




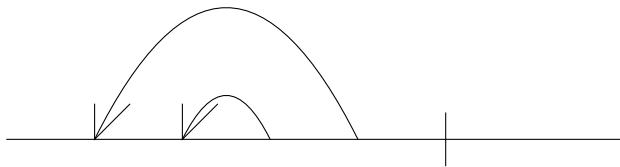








-

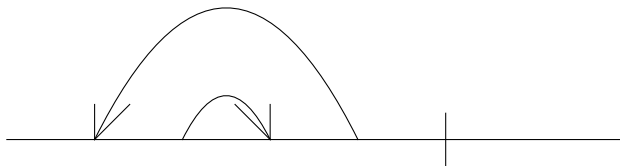


,

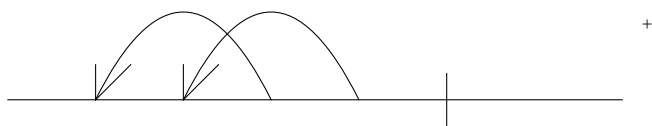
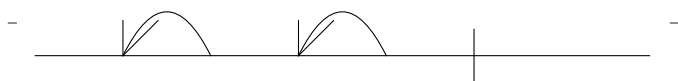
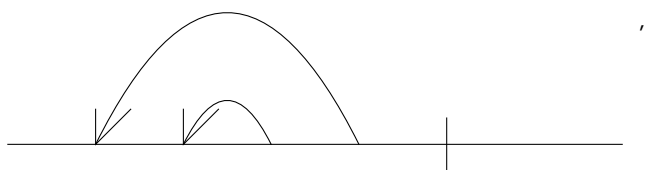
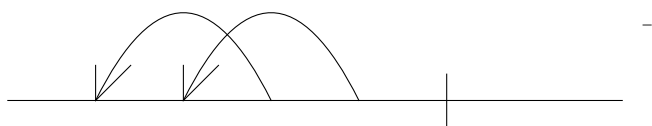


-

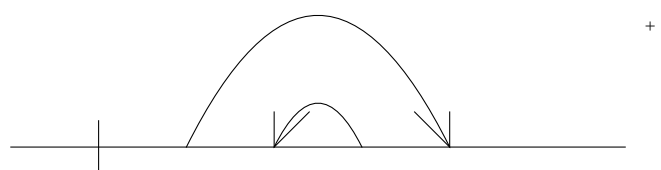
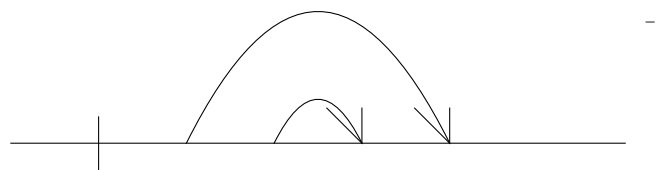
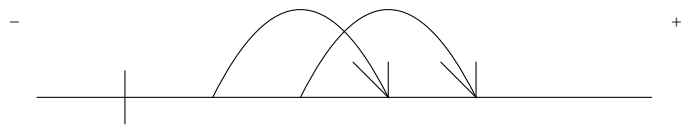
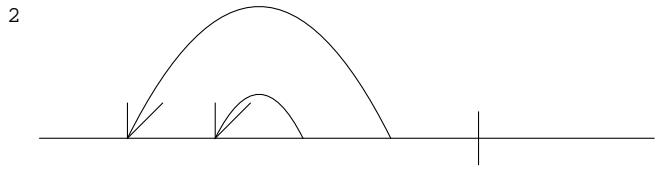
+

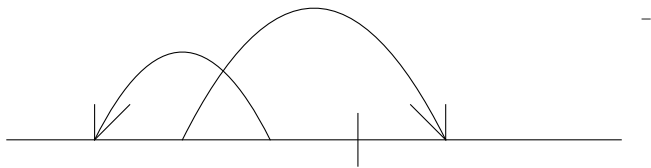
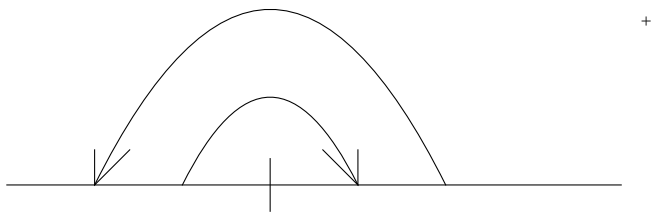
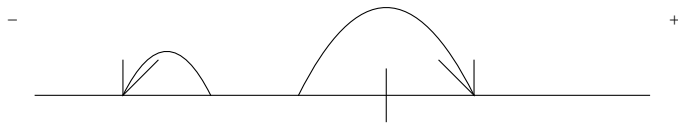


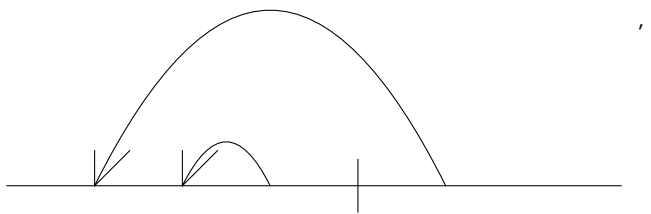
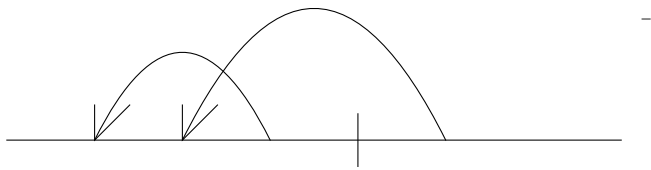
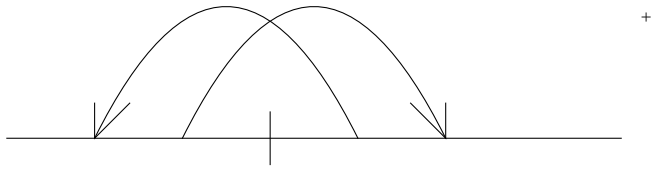
+

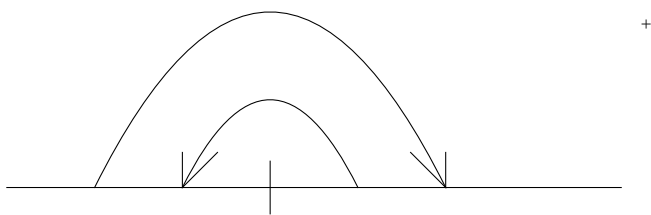
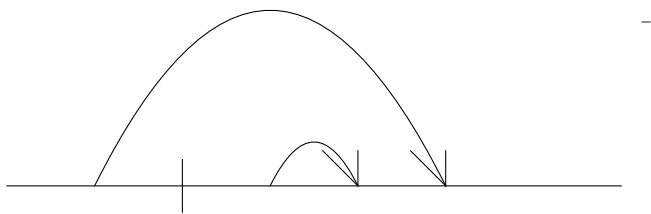
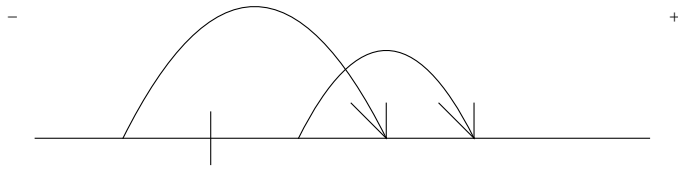


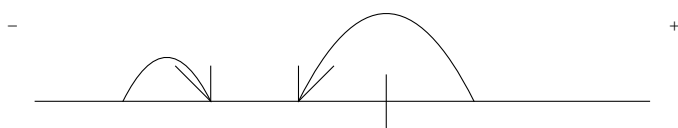
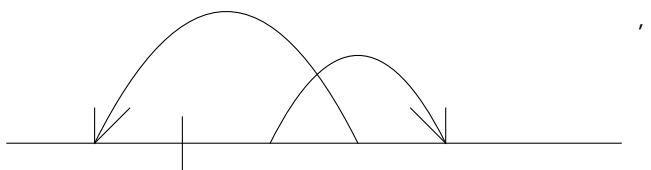
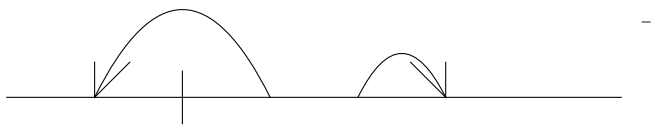
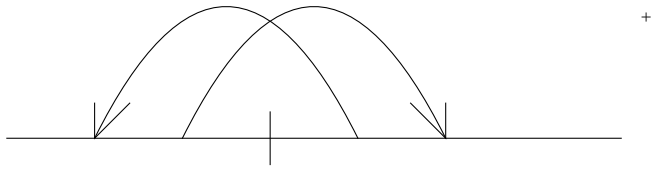


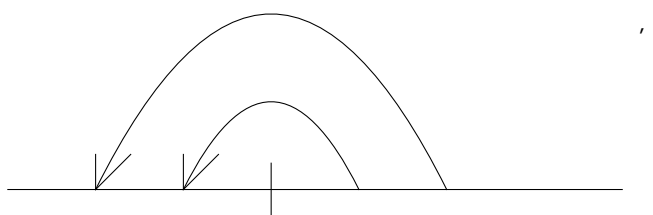
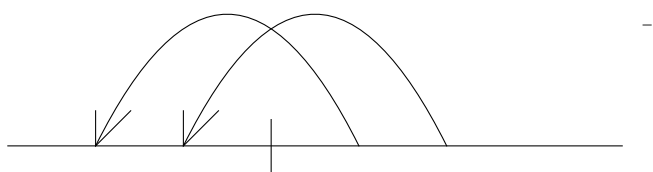
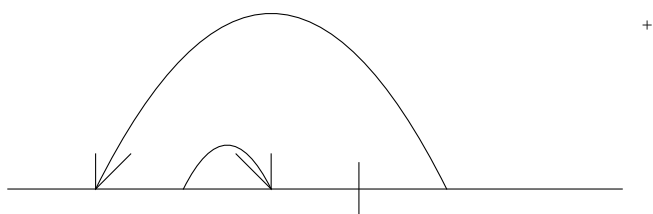


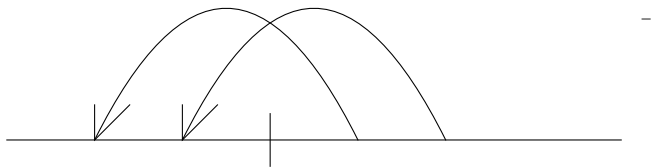
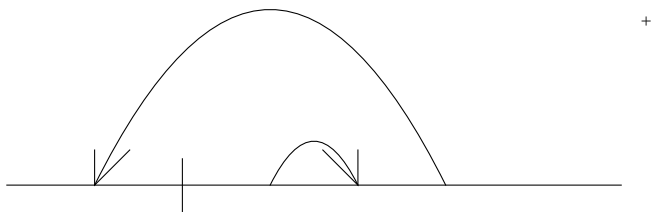
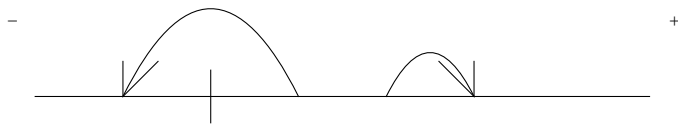


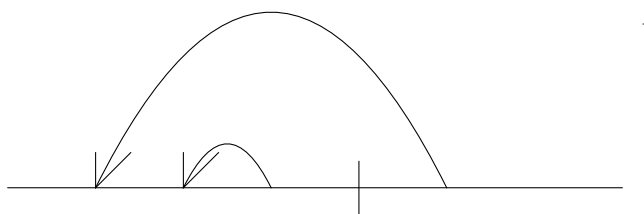
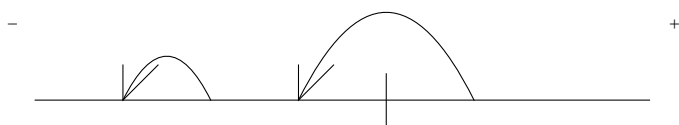
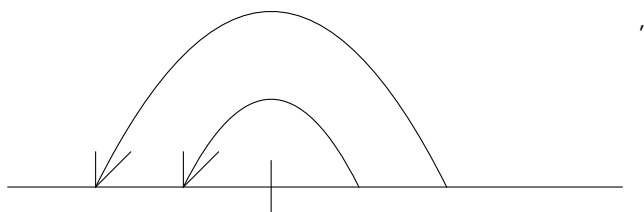




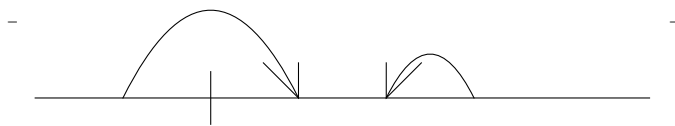
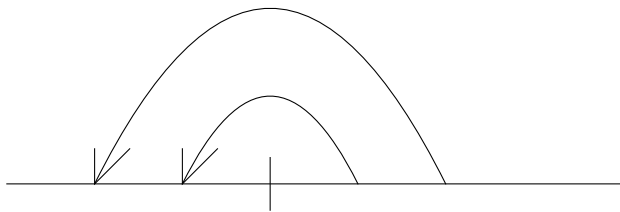
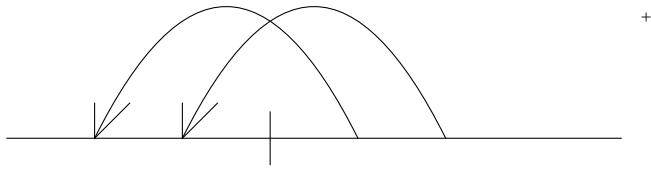


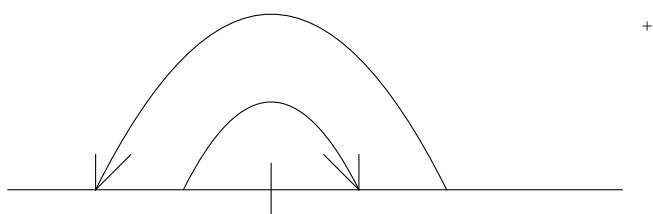
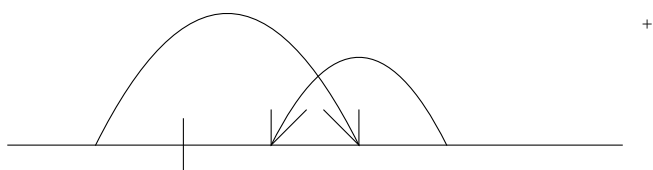
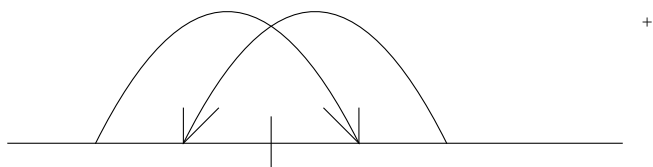


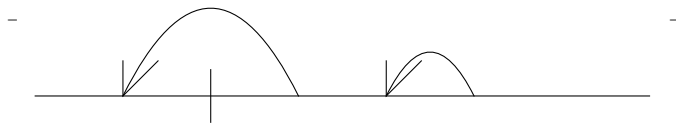
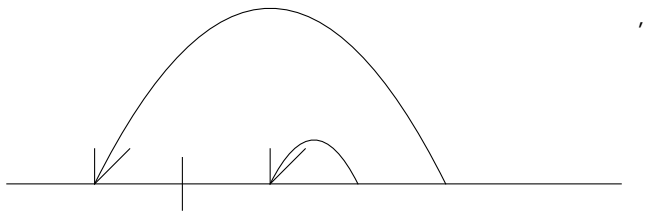
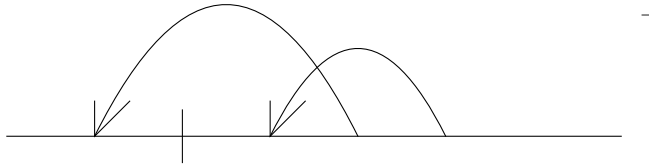


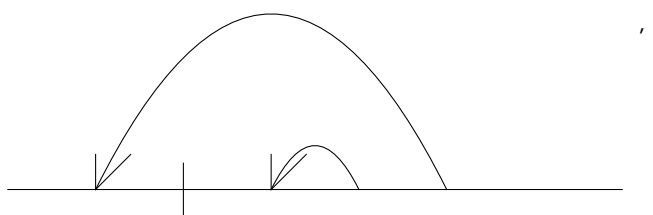
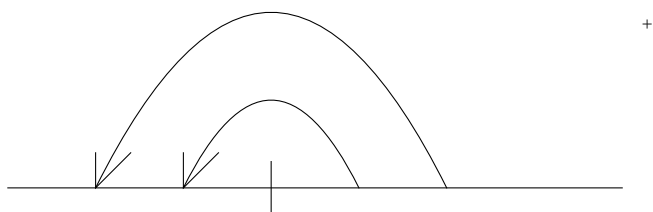
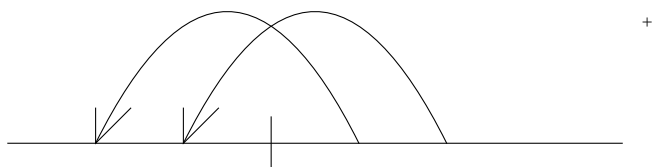


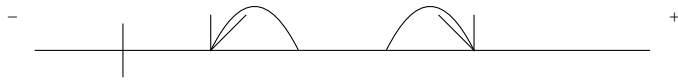
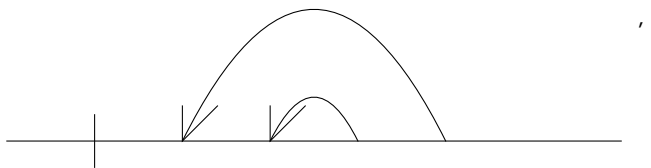
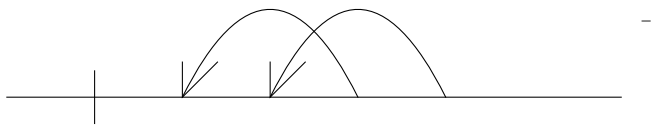
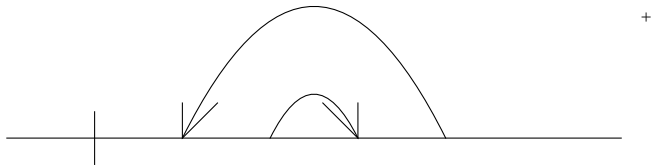
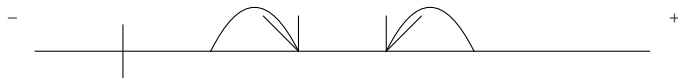


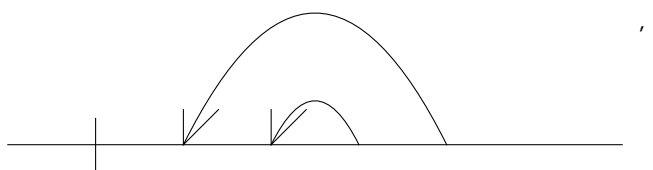
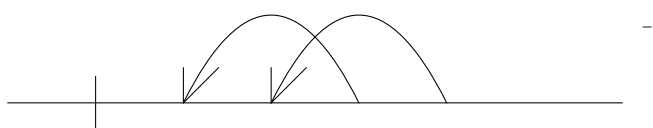
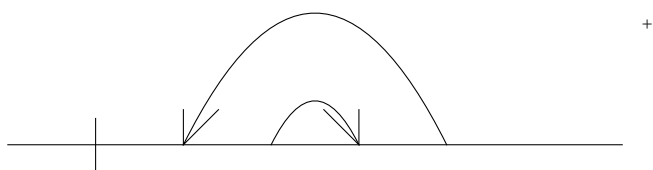


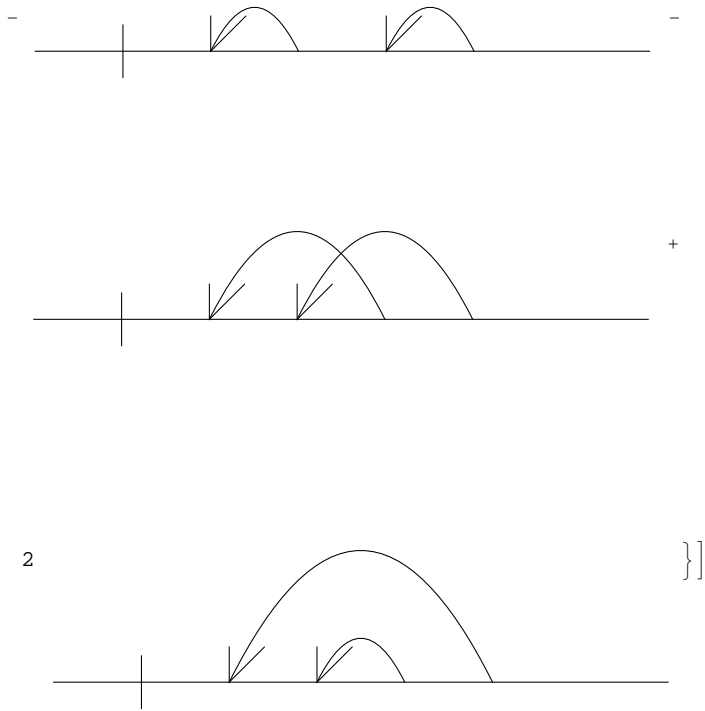












Computing Dimensions

```

In[75]:= 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[83]:= 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[4, reltypes, opts];
Analyze[m_Integer, 2, reltypes_List, opts___Rule] := Module[
  {qs},
  Table[
    {
      qs = PSpace[n, 2, reltypes, Gr → True, opts];
      (Length /@ qs) → Dim[qs, opts],
      qs = PSpace[n, 2, reltypes, Gr → False, opts];
      (Length /@ qs) → Dim[qs, opts]
    },
    {n, 2, m}
  ]
];
Analyze[2, reltypes_List, opts___Rule] := Analyze[4, 2, reltypes, opts];
```

## Runs

### Runs as in the paper

```
Analyze[{"R3b", "R2c", "R1", "Round"}] // MatrixForm
```

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 3] \rightarrow 0 & \text{QuotientSpace}[1, 3] \rightarrow 0 \\ \text{QuotientSpace}[12, 29] \rightarrow 1 & \text{QuotientSpace}[12, 31] \rightarrow 1 \\ \text{QuotientSpace}[107, 297] \rightarrow 4 & \text{QuotientSpace}[118, 336] \rightarrow 5 \end{array} \right)$$

```
Analyze[{"R3b", "R2c", "R1"}] // MatrixForm
```

$$\left( \begin{array}{cc} \text{QuotientSpace}[4, 4] \rightarrow 2 & \text{QuotientSpace}[4, 4] \rightarrow 2 \\ \text{QuotientSpace}[32, 37] \rightarrow 7 & \text{QuotientSpace}[34, 39] \rightarrow 9 \\ \text{QuotientSpace}[460, 749] \rightarrow 42 & \text{QuotientSpace}[500, 829] \rightarrow 51 \end{array} \right)$$

```
Analyze[{"R3b", "R2c", "R1", "Descending"}] // MatrixForm
```

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 1] \rightarrow 0 & \text{QuotientSpace}[1, 1] \rightarrow 0 \\ \text{QuotientSpace}[3, 3] \rightarrow 1 & \text{QuotientSpace}[3, 4] \rightarrow 1 \\ \text{QuotientSpace}[22, 23] \rightarrow 6 & \text{QuotientSpace}[27, 33] \rightarrow 7 \end{array} \right)$$


---

**Analyze[{"R3b", "R2c", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[3, 3] \rightarrow 1 & \text{QuotientSpace}[4, 3] \rightarrow 2 \\ \text{QuotientSpace}[16, 21] \rightarrow 2 & \text{QuotientSpace}[21, 24] \rightarrow 4 \\ \text{QuotientSpace}[160, 329] \rightarrow 7 & \text{QuotientSpace}[187, 362] \rightarrow 11 \end{array} \right)$$

**Analyze[{"R3b", "R2c"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[10, 7] \rightarrow 5 & \text{QuotientSpace}[12, 7] \rightarrow 7 \\ \text{QuotientSpace}[96, 121] \rightarrow 15 & \text{QuotientSpace}[110, 129] \rightarrow 22 \\ \text{QuotientSpace}[1332, 2401] \rightarrow 67 & \text{QuotientSpace}[1466, 2553] \rightarrow 89 \end{array} \right)$$

**Analyze[{"R3b", "R2c", "Descending"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[2, 2] \rightarrow 1 & \text{QuotientSpace}[3, 2] \rightarrow 2 \\ \text{QuotientSpace}[9, 11] \rightarrow 2 & \text{QuotientSpace}[13, 13] \rightarrow 4 \\ \text{QuotientSpace}[63, 96] \rightarrow 8 & \text{QuotientSpace}[82, 114] \rightarrow 12 \end{array} \right)$$


---

**Analyze[{"R3b", "R1", "R1d", "R1a", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 3] \rightarrow 0 & \text{QuotientSpace}[1, 3] \rightarrow 0 \\ \text{QuotientSpace}[12, 29] \rightarrow 1 & \text{QuotientSpace}[12, 31] \rightarrow 1 \\ \text{QuotientSpace}[107, 297] \rightarrow 4 & \text{QuotientSpace}[118, 334] \rightarrow 5 \end{array} \right)$$

**Analyze[{"R3b", "R1d", "R1a", "R1"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[4, 4] \rightarrow 2 & \text{QuotientSpace}[4, 4] \rightarrow 2 \\ \text{QuotientSpace}[32, 37] \rightarrow 7 & \text{QuotientSpace}[34, 39] \rightarrow 9 \\ \text{QuotientSpace}[460, 749] \rightarrow 42 & \text{QuotientSpace}[500, 821] \rightarrow 51 \end{array} \right)$$

**Analyze[{"R3b", "R1d", "R1", "Descending"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 1] \rightarrow 0 & \text{QuotientSpace}[1, 1] \rightarrow 0 \\ \text{QuotientSpace}[3, 3] \rightarrow 1 & \text{QuotientSpace}[3, 4] \rightarrow 1 \\ \text{QuotientSpace}[22, 23] \rightarrow 6 & \text{QuotientSpace}[27, 31] \rightarrow 7 \end{array} \right)$$


---

**Analyze[{"R3b", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[4, 3] \rightarrow 2 & \text{QuotientSpace}[5, 3] \rightarrow 3 \\ \text{QuotientSpace}[22, 21] \rightarrow 5 & \text{QuotientSpace}[27, 23] \rightarrow 8 \\ \text{QuotientSpace}[218, 353] \rightarrow 19 & \text{QuotientSpace}[245, 379] \rightarrow 27 \end{array} \right)$$

**Analyze[{"R3b"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[12, 6] \rightarrow 7 & \text{QuotientSpace}[14, 6] \rightarrow 9 \\ \text{QuotientSpace}[120, 120] \rightarrow 27 & \text{QuotientSpace}[134, 126] \rightarrow 36 \\ \text{QuotientSpace}[1680, 2520] \rightarrow 139 & \text{QuotientSpace}[1814, 2646] \rightarrow 175 \end{array} \right)$$

**Analyze[{"R3b", "Descending"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[3, 1] \rightarrow 2 & \text{QuotientSpace}[4, 1] \rightarrow 3 \\ \text{QuotientSpace}[15, 10] \rightarrow 6 & \text{QuotientSpace}[19, 11] \rightarrow 9 \\ \text{QuotientSpace}[105, 105] \rightarrow 24 & \text{QuotientSpace}[124, 116] \rightarrow 33 \end{array} \right)$$


---

**Analyze[{"R2c", "R1", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 2] \rightarrow 0 & \text{QuotientSpace}[1, 2] \rightarrow 0 \\ \text{QuotientSpace}[12, 13] \rightarrow 4 & \text{QuotientSpace}[12, 13] \rightarrow 4 \\ \text{QuotientSpace}[107, 75] \rightarrow 44 & \text{QuotientSpace}[118, 82] \rightarrow 48 \end{array} \right)$$

**Analyze[{"R2c", "R1"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[4, 2] \rightarrow 2 & \text{QuotientSpace}[4, 2] \rightarrow 2 \\ \text{QuotientSpace}[32, 5] \rightarrow 28 & \text{QuotientSpace}[34, 5] \rightarrow 30 \\ \text{QuotientSpace}[460, 41] \rightarrow 420 & \text{QuotientSpace}[500, 51] \rightarrow 450 \end{array} \right)$$

**Analyze[{"R2c", "R1", "Descending"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 1] \rightarrow 0 & \text{QuotientSpace}[1, 1] \rightarrow 0 \\ \text{QuotientSpace}[3, 2] \rightarrow 2 & \text{QuotientSpace}[3, 2] \rightarrow 2 \\ \text{QuotientSpace}[22, 5] \rightarrow 18 & \text{QuotientSpace}[27, 8] \rightarrow 20 \end{array} \right)$$

**Analyze[{"R2c", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[3, 1] \rightarrow 3 & \text{QuotientSpace}[4, 1] \rightarrow 4 \\ \text{QuotientSpace}[16, 1] \rightarrow 16 & \text{QuotientSpace}[21, 2] \rightarrow 20 \\ \text{QuotientSpace}[160, 1] \rightarrow 160 & \text{QuotientSpace}[187, 8] \rightarrow 180 \end{array} \right)$$

**Analyze[{"R2c"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[10, 1] \rightarrow 10 & \text{QuotientSpace}[12, 1] \rightarrow 12 \\ \text{QuotientSpace}[96, 1] \rightarrow 96 & \text{QuotientSpace}[110, 3] \rightarrow 108 \\ \text{QuotientSpace}[1332, 1] \rightarrow 1332 & \text{QuotientSpace}[1466, 27] \rightarrow 1440 \end{array} \right)$$

**Analyze[{"R2c", "Descending"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[2, 1] \rightarrow 2 & \text{QuotientSpace}[3, 1] \rightarrow 3 \\ \text{QuotientSpace}[9, 1] \rightarrow 9 & \text{QuotientSpace}[13, 2] \rightarrow 12 \\ \text{QuotientSpace}[63, 1] \rightarrow 63 & \text{QuotientSpace}[82, 8] \rightarrow 75 \end{array} \right)$$

## Other runs

**Analyze[{"R3b", "TC", "R1"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[4, 5] \rightarrow 1 & \text{QuotientSpace}[4, 5] \rightarrow 1 \\ \text{QuotientSpace}[26, 51] \rightarrow 1 & \text{QuotientSpace}[30, 60] \rightarrow 2 \\ \text{QuotientSpace}[267, 777] \rightarrow 2 & \text{QuotientSpace}[297, 844] \rightarrow 4 \end{array} \right)$$

**Analyze[{"R3b", "TC"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[9, 7] \rightarrow 4 & \text{QuotientSpace}[11, 7] \rightarrow 6 \\ \text{QuotientSpace}[64, 97] \rightarrow 7 & \text{QuotientSpace}[75, 103] \rightarrow 13 \\ \text{QuotientSpace}[625, 1501] \rightarrow 12 & \text{QuotientSpace}[700, 1603] \rightarrow 25 \end{array} \right)$$

**Analyze[{"R3b", "TC", "R1", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 3] \rightarrow 0 & \text{QuotientSpace}[1, 3] \rightarrow 0 \\ \text{QuotientSpace}[8, 29] \rightarrow 0 & \text{QuotientSpace}[9, 34] \rightarrow 0 \\ \text{QuotientSpace}[54, 253] \rightarrow 0 & \text{QuotientSpace}[63, 298] \rightarrow 0 \end{array} \right)$$

**Analyze[{"R3b", "TC", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[3, 3] \rightarrow 1 & \text{QuotientSpace}[4, 3] \rightarrow 2 \\ \text{QuotientSpace}[11, 15] \rightarrow 1 & \text{QuotientSpace}[15, 17] \rightarrow 3 \\ \text{QuotientSpace}[70, 185] \rightarrow 1 & \text{QuotientSpace}[85, 205] \rightarrow 4 \end{array} \right)$$

**Analyze[{"R3b", "R1", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 3] \rightarrow 0 & \text{QuotientSpace}[1, 3] \rightarrow 0 \\ \text{QuotientSpace}[12, 29] \rightarrow 1 & \text{QuotientSpace}[13, 33] \rightarrow 1 \\ \text{QuotientSpace}[107, 297] \rightarrow 4 & \text{QuotientSpace}[122, 345] \rightarrow 5 \end{array} \right)$$

**Analyze[{"R3b", "R1"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[4, 4] \rightarrow 2 & \text{QuotientSpace}[4, 4] \rightarrow 2 \\ \text{QuotientSpace}[32, 37] \rightarrow 7 & \text{QuotientSpace}[36, 43] \rightarrow 9 \\ \text{QuotientSpace}[460, 749] \rightarrow 42 & \text{QuotientSpace}[504, 827] \rightarrow 51 \end{array} \right)$$

**Analyze[{"R3b", "R1", "Descending"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 1] \rightarrow 0 & \text{QuotientSpace}[1, 1] \rightarrow 0 \\ \text{QuotientSpace}[3, 3] \rightarrow 1 & \text{QuotientSpace}[4, 4] \rightarrow 1 \\ \text{QuotientSpace}[22, 23] \rightarrow 6 & \text{QuotientSpace}[28, 30] \rightarrow 7 \end{array} \right)$$

**Analyze[{"R3b", "HTC", "R1"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[4, 8] \rightarrow 0 & \text{QuotientSpace}[4, 8] \rightarrow 0 \\ \text{QuotientSpace}[10, 37] \rightarrow 0 & \text{QuotientSpace}[14, 47] \rightarrow 0 \\ \text{QuotientSpace}[75, 485] \rightarrow 0 & \text{QuotientSpace}[97, 587] \rightarrow 0 \end{array} \right)$$

**Analyze[{"R3b", "HTC"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[7, 7] \rightarrow 3 & \text{QuotientSpace}[9, 7] \rightarrow 5 \\ \text{QuotientSpace}[34, 79] \rightarrow 4 & \text{QuotientSpace}[43, 85] \rightarrow 9 \\ \text{QuotientSpace}[209, 877] \rightarrow 5 & \text{QuotientSpace}[252, 961] \rightarrow 14 \end{array} \right)$$

**Analyze[{"R3b", "HTC", "R1", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[1, 3] \rightarrow 0 & \text{QuotientSpace}[1, 3] \rightarrow 0 \\ \text{QuotientSpace}[3, 13] \rightarrow 0 & \text{QuotientSpace}[4, 19] \rightarrow 0 \\ \text{QuotientSpace}[7, 66] \rightarrow 0 & \text{QuotientSpace}[13, 113] \rightarrow 0 \end{array} \right)$$

**Analyze[{"R3b", "HTC", "Round"}] // MatrixForm**

$$\left( \begin{array}{cc} \text{QuotientSpace}[2, 3] \rightarrow 1 & \text{QuotientSpace}[3, 3] \rightarrow 2 \\ \text{QuotientSpace}[5, 11] \rightarrow 1 & \text{QuotientSpace}[8, 13] \rightarrow 3 \\ \text{QuotientSpace}[20, 95] \rightarrow 1 & \text{QuotientSpace}[28, 111] \rightarrow 4 \end{array} \right)$$

In[87]:= **Analyze[2, {"R3b", "HTC"}] // MatrixForm**

Out[87]//MatrixForm=

$$\left( \begin{array}{cc} \text{QuotientSpace}[39, 25] \rightarrow 22 & \text{QuotientSpace}[45, 25] \rightarrow 28 \\ \text{QuotientSpace}[292, 505] \rightarrow 62 & \text{QuotientSpace}[337, 529] \rightarrow 90 \\ \text{QuotientSpace}[2505, 8161] \rightarrow 148 & \text{QuotientSpace}[2842, 8689] \rightarrow 238 \end{array} \right)$$