The Mathematica® Journal

Every Clutter Is a Tree of Blobs

Gus Wiseman

Given a finite vertex set, one can construct every connected spanning hypergraph by first choosing a spanning hypertree, then choosing a blob on each of its edges.

Introduction

If V is a finite vertex set and $E \subseteq 2^V$ is a collection of finite subsets (called edges), none of which is a subset of another, we recursively define the *swell* of E, Swell(E), to be the collection of all sets that either:

1. belong to E
2. are the union of some pair of *overlapping* sets, both already belonging to Swell(E)

For example, if

$$E = \{\{1, 2\}, \{1, 3\}, \{2, 3, 4\}\},$$

then

$$\text{Swell}(E) = \{\{1, 2\}, \{1, 3\}, \{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}.$$

If we also have $\cup(E) \in \text{Swell}(E)$, then the set system E is called a *clutter*. This condition means that (except in the case $|\cup(E)| = 1$) each edge contains at least two vertices, and the hypergraph spanned by the edge set is connected. Here the hypergraph (V, E) spanned by a set of edges E is defined to have vertex set $V = \cup(E)$ and edge set E. (There is no agreed-upon definition of "hypergraph." For some authors it is any set system; for others it is a simplicial complex; for others it is an antichain of sets.)

```
subsetQ[t_, s_] := Or[
  Length[t] === 0,
  And[MemberQ[s, First[t]], subsetQ[Rest[t], s]]
 ]
```

```
stableQ[u_] := stableQ[u, subsetQ]

stableQ[u_, Q_] :=
 Not[Apply[Or, Outer[#1 =!= #2 && Q[#1, #2] &, u, u, 1], {0, 1}]]

swell[c : {{__}?OrderedQ ..}] :=
 Union @@ FixedPointList[
   Union[ReplaceList[#,
       {___, a : {___, x_, ___}, ___, b : {___, x_, ___}, ___} :>
        Union[a, b]]] &, c]

connectedQ[c : {{_, __}?OrderedQ ..}] :=
 MemberQ[swell[c], Union @@ c]

clutterQ[c : {{_, __}?OrderedQ ..}] :=
 And[stableQ[c], connectedQ[c]]

clutterQ[
 clut = {{1, 2}, {1, 3, 4}, {1, 3, 5}, {3, 6, 7}, {6, 7, 8}}]

True
```

Here is a larger clutter.

```
swell[clut]

{{1, 2}, {1, 3, 4}, {1, 3, 5}, {3, 6, 7}, {6, 7, 8},
 {1, 2, 3, 4}, {1, 2, 3, 5}, {1, 3, 4, 5}, {3, 6, 7, 8},
 {1, 2, 3, 4, 5}, {1, 3, 4, 6, 7}, {1, 3, 5, 6, 7},
 {1, 2, 3, 4, 6, 7}, {1, 2, 3, 5, 6, 7}, {1, 3, 4, 5, 6, 7},
 {1, 3, 4, 6, 7, 8}, {1, 3, 5, 6, 7, 8}, {1, 2, 3, 4, 5, 6, 7},
 {1, 2, 3, 4, 6, 7, 8}, {1, 2, 3, 5, 6, 7, 8},
 {1, 3, 4, 5, 6, 7, 8}, {1, 2, 3, 4, 5, 6, 7, 8}}
```

The number of clutters $|C(n)|$ spanning $n = 1, 2, \ldots, 8$ vertices is given by A048143 (oeis.org/A048143),

$$1, 1, 5, 84, 6348, 7\,743\,728, 2\,414\,572\,893\,530, 56\,130\,437\,190\,053\,299\,918\,162.$$

This sequence varies as 2^{2^n}, so the number of digits required roughly doubles with each consecutive term. Our main example is just one of some 56 sextillion members of $C(8)$.

```
normalizeColumns[m_List, aft_Integer] :=
 Module[{allcols, cols, mx, leads, resets},
  allcols = Transpose[m];
  If[Length[allcols] ≤ aft, Return[{m}]];
  cols = Drop[allcols, aft - 1];
  mx = Plus @@@ cols;
  leads = First /@ Position[mx, Max[mx]];
  resets =
   Union[
    Function[par,
      Sort[Transpose[Join[Take[allcols, aft - 1],
         Prepend[Delete[cols, par], cols〚par〛]]]]] /@
     leads];
  Union @@ (normalizeColumns[#, aft + 1] & /@ resets)
 ]

normalizeColumns[m_List] :=
 First[Sort[normalizeColumns[m, 1]]]

clutterToArray[c_] := Transpose[Outer[Count, c, Union @@ c, 1]]

arrayToClutter[m_] :=
 Table[Join @@ Position[m〚All, i〛, 1], {i, Length[m〚1〛]}]

normalizeClutter[c_] :=
 Sort[arrayToClutter[normalizeColumns[clutterToArray[c]]]]
```

This normalizing function is a universal invariant for the species of labeled clutters, meaning two clutters are isomorphic iff they have the same image.

```
normalizeClutter[clut]
```

```
{{1, 4}, {2, 4, 6}, {3, 4, 6}, {5, 7, 8}, {6, 7, 8}}
```

Here is a list of nonisomorphic representatives for all clutters with up to four vertices, corresponding to “unlabeled” clutters. This brute-force enumeration may not work for $n > 4$.

```
allNormalSpanningClutters[n_Integer] :=
  If[n === 1, {{{1}}},
   Union[normalizeClutter /@
     Select[Subsets[Select[Subsets[Range[n]],
        Length[#] > 1 &]],
      And[Union @@ # === Range[n], clutterQ[#]] &]]];
```

```
Table[
 Column[Apply[SequenceForm, allNormalSpanningClutters[n],
   {2}]], {n, 4}]
```

```
                              {1234}
                              {14, 234}
                              {134, 234}
                              {12, 134, 234}
                              {13, 14, 234}
                              {13, 24, 34}
                {123}         {14, 24, 34}
{ {1} , {12} , {13, 23}    ,  {124, 134, 234}          }
                {12, 13, 23}  {12, 13, 14, 234}
                              {12, 13, 24, 34}
                              {14, 23, 24, 34}
                              {123, 124, 134, 234}
                              {13, 14, 23, 24, 34}
                              {12, 13, 14, 23, 24, 34}
```

■ Kernels and Caps in Clutters

A *kernel* of E is a clutter $E|w$ (the restriction of E to edges that are subsets of w) for some $w \in \text{Swell}(E)$.

```
kernels[c : {{_, __}?OrderedQ ..}] :=
 Function[s, Select[c, subsetQ[#, s] &]] /@ swell[c]
```

```
kernels[clut]
```

```
{{{1, 2}}, {{1, 3, 4}}, {{1, 3, 5}},
 {{3, 6, 7}}, {{6, 7, 8}}, {{1, 2}, {1, 3, 4}},
 {{1, 2}, {1, 3, 5}}, {{1, 3, 4}, {1, 3, 5}},
 {{3, 6, 7}, {6, 7, 8}}, {{1, 2}, {1, 3, 4}, {1, 3, 5}},
 {{1, 3, 4}, {3, 6, 7}}, {{1, 3, 5}, {3, 6, 7}},
 {{1, 2}, {1, 3, 4}, {3, 6, 7}}, {{1, 2}, {1, 3, 5}, {3, 6, 7}},
 {{1, 3, 4}, {1, 3, 5}, {3, 6, 7}},
 {{1, 3, 4}, {3, 6, 7}, {6, 7, 8}},
 {{1, 3, 5}, {3, 6, 7}, {6, 7, 8}},
 {{1, 2}, {1, 3, 4}, {1, 3, 5}, {3, 6, 7}},
 {{1, 2}, {1, 3, 4}, {3, 6, 7}, {6, 7, 8}},
 {{1, 2}, {1, 3, 5}, {3, 6, 7}, {6, 7, 8}},
 {{1, 3, 4}, {1, 3, 5}, {3, 6, 7}, {6, 7, 8}},
 {{1, 2}, {1, 3, 4}, {1, 3, 5}, {3, 6, 7}, {6, 7, 8}}}
```

Define $f^*(\{a, b, \ldots, z\}) = \{f(a), f(b), \ldots, f(z)\}$.

A *set partition* $\pi \vdash S$ is a set of disjoint sets with $\cup(\pi) = S$.

Suppose $\pi \vdash E$ is a set partition of E such that each block $T \in \pi$ is a kernel of E (i.e. $\cup(T) \in \mathrm{Swell}(E)$ and $T = E \mid \cup(T)$). Since $S \subseteq T$ would imply $E \mid S \subseteq E \mid T$, it follows that the set of unions $F = \cup^*(\pi)$ is itself a clutter, which we call a *cap* of E.

```
setPartitionsUsing[pile_, span_] := Module[{samples},
  If[Length[pile] === 0,
   If[Length[span] === 0, Return[{{}}], Return[{}]]];
  samples = Select[pile, MemberQ[#, First[span]] &];
  Join @@
   Table[Prepend[#, samples[[i]]] & /@
     setPartitionsUsing[
      Select[pile,
       Length[Intersection[samples[[i]], #]] === 0 &],
      Complement[span, samples[[i]]]],
    {i, 1, Length[samples]}]
]
```

```
clutterPartitions[c : {{_, __}?OrderedQ ..}] :=
 setPartitionsUsing[kernels[c], c]
```

Equivalently, a cap F of E is a clutter satisfying both:

1. $F \subseteq \mathrm{Swell}(E)$
2. every edge of E is a subset of exactly one edge of F

```
caps[c : {{_, __}?OrderedQ ..}] :=
 Sort /@ Apply[Union, clutterPartitions[c], {2}]
```

```
caps[clut]
```

```
{{{1, 2}, {1, 3, 4}, {1, 3, 5}, {3, 6, 7}, {6, 7, 8}},
 {{1, 2}, {1, 3, 4}, {1, 3, 5}, {3, 6, 7, 8}},
 {{1, 2}, {1, 3, 4}, {6, 7, 8}, {1, 3, 5, 6, 7}},
 {{1, 2}, {1, 3, 4}, {1, 3, 5, 6, 7, 8}},
 {{1, 2}, {3, 6, 7}, {6, 7, 8}, {1, 3, 4, 5}},
 {{1, 2}, {1, 3, 4, 5}, {3, 6, 7, 8}},
 {{1, 2}, {1, 3, 5}, {6, 7, 8}, {1, 3, 4, 6, 7}},
 {{1, 2}, {6, 7, 8}, {1, 3, 4, 5, 6, 7}},
 {{1, 2}, {1, 3, 5}, {1, 3, 4, 6, 7, 8}},
 {{1, 2}, {1, 3, 4, 5, 6, 7, 8}},
 {{1, 3, 5}, {3, 6, 7}, {6, 7, 8}, {1, 2, 3, 4}},
 {{1, 3, 5}, {1, 2, 3, 4}, {3, 6, 7, 8}},
 {{6, 7, 8}, {1, 2, 3, 4}, {1, 3, 5, 6, 7}},
 {{1, 2, 3, 4}, {1, 3, 5, 6, 7, 8}},
 {{1, 3, 4}, {3, 6, 7}, {6, 7, 8}, {1, 2, 3, 5}},
```

```
{{1, 3, 4}, {1, 2, 3, 5}, {3, 6, 7, 8}},
{{6, 7, 8}, {1, 2, 3, 5}, {1, 3, 4, 6, 7}},
{{1, 2, 3, 5}, {1, 3, 4, 6, 7, 8}},
{{3, 6, 7}, {6, 7, 8}, {1, 2, 3, 4, 5}},
{{3, 6, 7, 8}, {1, 2, 3, 4, 5}},
{{1, 3, 5}, {6, 7, 8}, {1, 2, 3, 4, 6, 7}},
{{1, 3, 4}, {6, 7, 8}, {1, 2, 3, 5, 6, 7}},
{{6, 7, 8}, {1, 2, 3, 4, 5, 6, 7}},
{{1, 3, 5}, {1, 2, 3, 4, 6, 7, 8}},
{{1, 3, 4}, {1, 2, 3, 5, 6, 7, 8}}, {{1, 2, 3, 4, 5, 6, 7, 8}}}
```

To see that this does *not* establish a partial order of clutters with a vertex set, observe that

- {{1, 2}, {1, 3}, {2, 3}, {3, 4}}
- {{1, 2}, {1, 3}, {2, 3, 4}}
- {{1, 2, 3}, {2, 3, 4}}

is a nontransitive chain of caps. The following is the set of all set partitions of the edge set indices corresponding to each cap of the clutter.

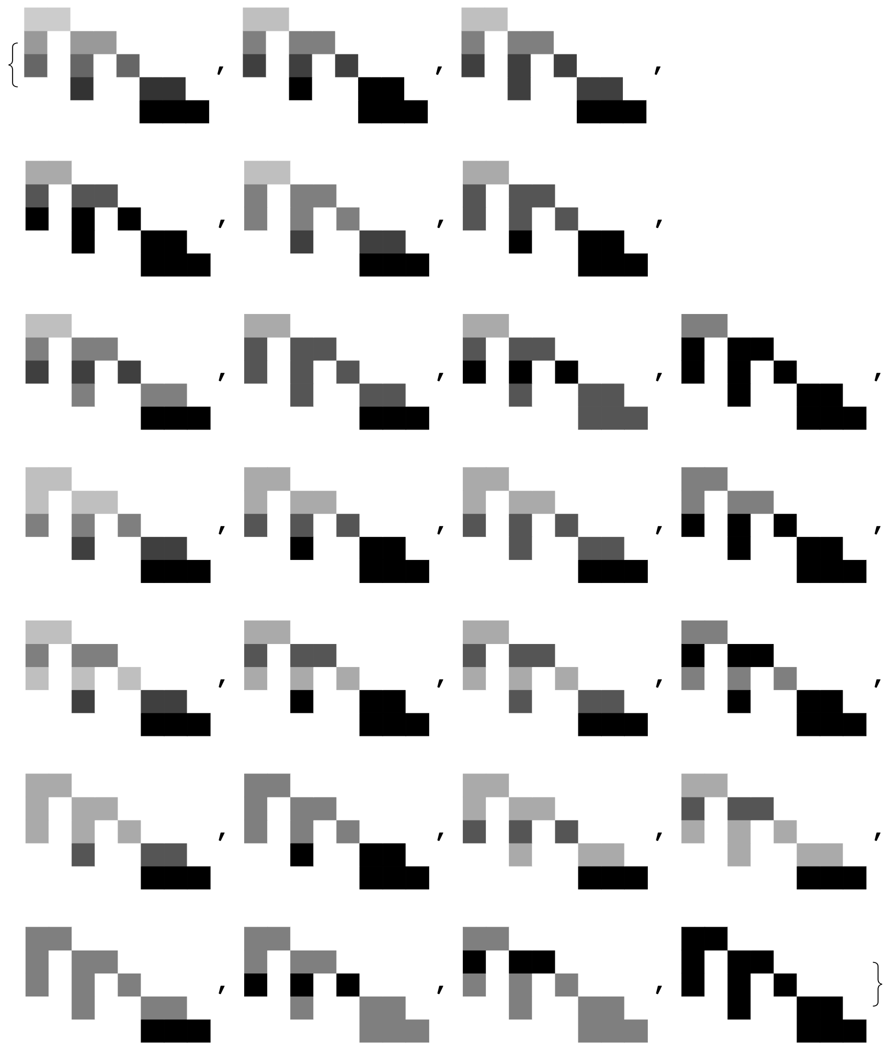
```
clutterPartitions[clut] /.
 Rule @@@ Transpose[{clut, Range[Length[clut]]}]
```

```
{{{1}, {2}, {3}, {4}, {5}}, {{1}, {2}, {3}, {4, 5}},
 {{1}, {2}, {3, 4}, {5}}, {{1}, {2}, {3, 4, 5}},
 {{1}, {2, 3}, {4}, {5}}, {{1}, {2, 3}, {4, 5}},
 {{1}, {2, 4}, {3}, {5}}, {{1}, {2, 3, 4}, {5}},
 {{1}, {2, 4, 5}, {3}}, {{1}, {2, 3, 4, 5}},
 {{1, 2}, {3}, {4}, {5}}, {{1, 2}, {3}, {4, 5}},
 {{1, 2}, {3, 4}, {5}}, {{1, 2}, {3, 4, 5}},
 {{1, 3}, {2}, {4}, {5}}, {{1, 3}, {2}, {4, 5}},
 {{1, 3}, {2, 4}, {5}}, {{1, 3}, {2, 4, 5}},
 {{1, 2, 3}, {4}, {5}}, {{1, 2, 3}, {4, 5}},
 {{1, 2, 4}, {3}, {5}}, {{1, 3, 4}, {2}, {5}},
 {{1, 2, 3, 4}, {5}}, {{1, 2, 4, 5}, {3}},
 {{1, 3, 4, 5}, {2}}, {{1, 2, 3, 4, 5}}}
```

```
clutterPartitionPlot[ptn_] :=
  ArrayPlot[Transpose[clutterToArray[Union @@ ptn]] *
    (Position[ptn, #, {2}, Heads → False][[1, 1]] & /@
      Union @@ ptn), Background → White, Frame → False,
   ImageSize → Length[Union @@ Union @@ ptn] * 10];
```

In these plots of clutter partitions, the filled squares correspond to all pairs of a vertex and an edge such that the vertex belongs to the edge; these squares are then shaded according to which block of the partition the edge belongs to.

```
clutterPartitionPlot /@ clutterPartitions[clut]
```



Trees and Blobs

The *density* of a clutter E is

$$\kappa(E) = \sum(|e| - 1) - \left|\cup(E)\right|,$$

where the sum is over all edges $e \in E$.

```
density[c_] := Total[(Length[#] - 1 &) /@ c] - Length[Union @@ c]
```

A clutter E with two or more edges is a *tree* iff $\kappa(E) = -1$. This is equivalent to the usual definition of a spanning hypertree [1].

```
treeQ[c_] := And[Length[c] ≥ 2, density[c] === -1];
```

A clutter E is a *blob* iff no cap of E is a tree.

```
blobQ[c_] := Apply[And, Composition[Not, treeQ] /@ caps[c]]
```

The trees and blobs among the caps and kernels (respectively) of our running example are as follows.

```
Select[caps[clut], treeQ]
```

```
{{{1, 2}, {1, 3, 4, 5}, {3, 6, 7, 8}},
 {{1, 2}, {1, 3, 4, 5, 6, 7, 8}},
 {{3, 6, 7, 8}, {1, 2, 3, 4, 5}}}
```

```
Select[kernels[clut], blobQ]
```

```
{{{1, 2}}, {{1, 3, 4}}, {{1, 3, 5}}, {{3, 6, 7}},
 {{6, 7, 8}}, {{1, 3, 4}, {1, 3, 5}}, {{3, 6, 7}, {6, 7, 8}}}
```

Suppose a clutter E decomposes into a cap F and corresponding set of kernels $\xi \vdash E$. Then

$$\kappa(E) - \kappa(F) = \sum(\kappa(H) + 1),$$

where the sum is over all $H \in \xi$. In particular, $\kappa(F) \leq \kappa(E)$, and $\kappa(E) = \kappa(F)$ iff every $H \in \xi$ is a tree. Using this simple identity, one easily proves the following.

Lemma

Every kernel (with two or more edges) of a tree is a tree.

Every cap (with two or more edges) of a tree is a tree.

The union of a set of trees whose set of unions is a tree, is a tree.

The following is also straightforward.

Proposition

A clutter E is a tree iff no kernel of E is a blob.

We now come to the main result.

```
maximize[c_] :=
 Complement[c,
  First /@ Select[Tuples[c, 2],
    And[UnsameQ @@ #, subsetQ @@ #] &]]
```

```
treeOfBlobs[c_] := maximize[Select[kernels[c], blobQ]]
```

For our running example, the theorem corresponds to the following decomposition into a tree of blobs.

```
Union @@@ treeOfBlobs[clut](* tree *)
```

```
{{1, 2}, {1, 3, 4, 5}, {3, 6, 7, 8}}
```

```
treeOfBlobs[clut](* blobs *)
```

```
{{{1, 2}}, {{1, 3, 4}, {1, 3, 5}}, {{3, 6, 7}, {6, 7, 8}}}
```

Theorem

Assume E is not a blob. Let $\tau = \tau(E)$ be the subset-maximal kernels of E that are blobs. Then τ is a set partition of E whose set of unions $\cup^*(\tau)$ is a tree.

Proof

First we show that any blob (kernel) is contained within a single branch of any tree (cap). Suppose that $B = E \mid w$ is a kernel of E and is a blob, and that T is a cap of E that is a tree. Let T' be the subtree of T contributing to the set partition $\pi \vdash B$ of *non-empty* intersections $B \cap (E \mid t)$ for each branch $t \in T'$. The set of unions $H = \cup^*(\pi)$ forms a clutter that is obtained from T' by deleting in turn all vertices not in w, a process that weakly decreases density. Let $\sigma \vdash B$ be the set partition comprised of maximal kernels (i.e. connected components) contained in blocks of π. Then $F = \cup^*(\sigma)$ is a cap of B and $\kappa(H) - \kappa(F) = |\sigma| - |\pi|$. Since F is a connected clutter, we have $-1 \leq \kappa(F) \leq \kappa(H) \leq \kappa(T') \leq -1$, and therefore $F = H$. But since B is a blob, F cannot be a tree, hence it must be a maximal cap (viz. $F = \{w\}, \pi = \{B\}$).

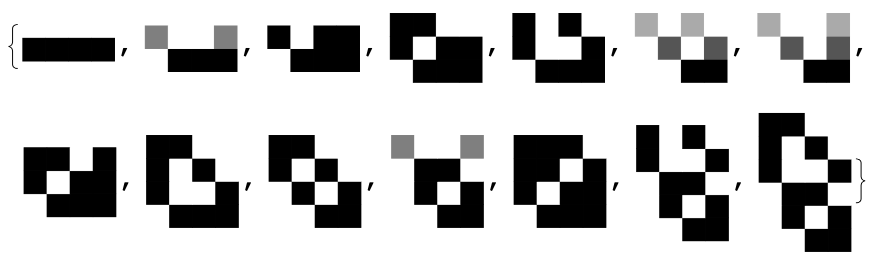
Next we show that $\tau(E) \vdash E$. If any two blobs overlap, both blobs must be contained entirely in whatever branch (of any given tree) contains their intersection. This implies that there is another blob containing their union, and hence that the maximal blobs $\tau(E)$ are disjoint. Since every singleton is also a blob, we conclude that $\cup^*(\tau)$ is a cap of E.

Finally, if any kernel of $\cup^*(\tau)$ were a blob, so would be the restriction of E to its union, contradicting maximality of τ. This proves that the set of unions of τ is a tree. ■

The following are the decompositions $\tau(E)$ for each nonisomorphic clutter E with four vertices.

```
treeOfBlobs /@ allNormalSpanningClutters[4]
clutterPartitionPlot /@ %
```

```
{{{{1, 2, 3, 4}}}, {{{1, 4}}, {{2, 3, 4}}},
 {{{1, 3, 4}, {2, 3, 4}}}, {{{1, 2}, {1, 3, 4}, {2, 3, 4}}},
 {{{1, 3}, {1, 4}, {2, 3, 4}}}, {{{1, 3}}, {{2, 4}}, {{3, 4}}},
 {{{1, 4}}, {{2, 4}}, {{3, 4}}},
 {{{1, 2, 4}, {1, 3, 4}, {2, 3, 4}}},
 {{{1, 2}, {1, 3}, {1, 4}, {2, 3, 4}}},
 {{{1, 2}, {1, 3}, {2, 4}, {3, 4}}},
 {{{1, 4}}, {{2, 3}, {2, 4}, {3, 4}}},
 {{{1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}}},
 {{{1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}}},
 {{{1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}}}}
```



Connected Sets of Kernels

Let $\ker(E)$ be the set of all kernels of E. If $K \subseteq \ker(E)$ is itself a (connected) clutter with vertex set $E = \cup(K)$, then there exists a unique subset-minimal upper bound $m(K) = E \mid \cup(\cup^*(K))$ satisfying both

1. $m(K) \in \ker(E)$
2. $k \subseteq m(K)$ for all $k \in K$

In general, we can only define $m(K)$ uniquely for K a *connected* set of kernels, so m is not strictly a join operation for the poset of subsets $\ker(E) \subseteq 2^E$. But if K is not connected as a clutter, then letting $\pi \vdash K$ be its (maximal) connected components, we say that K is a *connected set of kernels* iff $m^*(\pi)$ is connected as a set of kernels, in which case the join is given by

$$m(K) = m(m^*(\pi)).$$

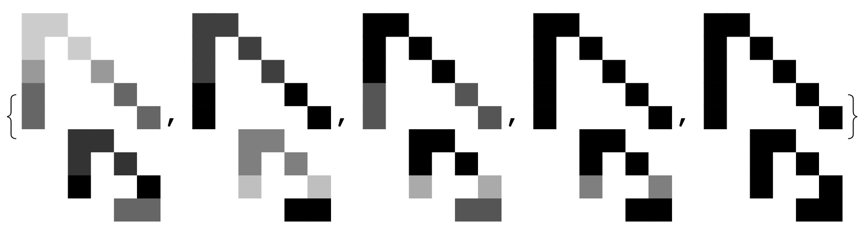
In practice, the verification of connectedness and the computation of m may require several iterations constructing joins of connected components. For example, consider the connected set of kernels.

```
kerset = {{{1, 2}, {1, 3}}, {{1, 2}, {1, 4}},
   {{1, 5}, {1, 6}, {5, 6}}, {{3, 4}, {3, 5}}, {{3, 6}}};
```

It has the following sequence of joins of connected components.

```
Most[FixedPointList[
  Function[s, Select[Union @@ #, subsetQ[#, Union @@ s] &]] /@
    maximize[swell[#]] &, kerset]]
clutterPartitionPlot /@ %
```

```
{{{{1, 2}, {1, 3}}, {{1, 2}, {1, 4}}, {{1, 5}, {1, 6}, {5, 6}},
  {{3, 4}, {3, 5}}, {{3, 6}}}, {{{3, 6}}, {{3, 4}, {3, 5}},
  {{1, 2}, {1, 3}, {1, 4}, {3, 4}}, {{1, 5}, {1, 6}, {5, 6}}},
 {{{3, 6}}, {{1, 5}, {1, 6}, {5, 6}},
  {{1, 2}, {1, 3}, {1, 4}, {1, 5}, {3, 4}, {3, 5}}},
 {{{3, 6}}, {{1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {3, 4},
   {3, 5}, {3, 6}, {5, 6}}}, {{{1, 2}, {1, 3}, {1, 4},
   {1, 5}, {1, 6}, {3, 4}, {3, 5}, {3, 6}, {5, 6}}}}
```



One Problem

If K is a connected set of kernels, we define its *compression* cmp(K) to be the number of iterations in the computation of $m(K)$ by constructing consecutive joins of connected components. For the previous example, we have cmp$(K) = 5$. Although it seems unlikely that cmp is a bounded invariant, we do not know how to construct an example with compression greater than 5.

1. For which positive numbers $n \geq 1$ does there exist a connected set of kernels K such that cmp$(K) = n$?
2. Does there exist an infinite chain $K^1 \subseteq K^2 \subseteq K^3 \subseteq \cdots$ of connected sets of kernels such that cmp$(K^i) <$ cmp(K^{i+1}) for all $i \geq 1$?

Define an invariant $\mu : \ker(E) \to \mathbb{Z}$ by

$$\sum \mu(\xi_1)\, \mu(\xi_2) \cdots \mu(\xi_n) = \delta^{|H|=1}$$

for all $H \in \ker(E)$, where the sum is over all clutter partitions $\ker(E) \supseteq \xi \vdash H$. Here δ^P denotes the indicator function for a proposition P, equal to 1 or 0 depending on whether P is true or false, respectively.

```
clutterMu[c_] := If[Length[c] === 1, 1, 0] -
  Total[
   Times @@@ Map[clutterMu,
     Select[clutterPartitions[c], Length[#] > 1 &], {2}]]
```

```
clutterMu /@ kernels[clut]
```

```
{1, 1, 1, 1, 1, -1, -1, -1, -1, 2,
 -1, -1, 1, 1, 2, 1, 1, -4, -1, -1, -2, 4}
```

```
clutterMu /@ caps[clut]
```

```
{4, -4, -2, 2, -1, 1, -2, 1, 2, -1, -2, 2,
 1, -1, -2, 2, 1, -1, 1, -1, 1, 1, -1, -1, -1, 1}
```

Theorem

For any $H \in \ker(E)$ we have $\mu(H) = \sum(-1)^{|K|}$, where the sum is over all connected sets of kernels spanning H.

Proof

Let $\operatorname{cptn}(E)$ be the set of all clutter partitions $\ker(E) \supseteq \pi \vdash E$. What we have essentially shown above is that $\operatorname{cptn}(E) \subseteq \cup^{-1}(E)$, regarded as a subposet of the lattice of set partitions ordered by refinement, is a lattice. We have the simple enumerative identity

$$\delta^{E=\cup^*(\pi)} = \prod\left(1 - \delta^{H \leq \pi}\right),$$

where the product is over all non-singleton kernels H, here regarded as elements of $\operatorname{cptn}(E)$ whose only non-singleton block is $\{H\}$. Expanding the right-hand side gives

$$\sum(-1)^{|S|}\,\delta^{m(S) \leq \pi},$$

where the sum is over all sets of non-singleton kernels $S \subseteq \ker(E)$, again regarded as lattice elements. Here $m(S) \in \operatorname{cptn}(E)$ is algorithmically the same operation as the connected-join operation on $\ker(E)$. Expanding and factoring accordingly, this becomes

$$\sum\sum(-1)^{|S|}\,\delta^{\sigma=m(S)} = \sum\prod\sum(-1)^{|K|},$$

where the outer sum is over all $\sigma \leq \pi$, the product is over all $H \in \sigma$, and the inner sum is over all connected sets of kernels $K \subseteq \ker(E)$ spanning H. For any kernel $F \in \ker(E)$, define

$$d(F) = \sum\prod\sum(-1)^{|K|},$$

where the outer sum is over all clutter partitions $\sigma \in \text{cptn}(F)$, and where the product and inner sum are as before. Letting π be the set partition of E whose only non-singleton block is $\{F\}$, we have shown that

$$d(F) = \delta^{E=\cup^*(\pi)} = \delta^{|F|=1}.$$

Hence our theorized expansion does indeed satisfy the defining identity of μ. ■

Note that it is sufficient in the preceding theorem and proof to consider only connected sets of *subset-minimal* non-singleton kernels, and it is often practical to do so. The hypergraph $(E, \text{minker}(E))$ whose edges are minimal non-singleton kernels is also of some interest. The well-known Möbius function of a hypergraph is defined on the lattice of connected set partitions, and in this context an element of Swell(minker(E)) may be called a *pseudo-kernel*. In comparison, however, our invariant μ, which is defined on essentially all clutters, seems to be more interesting; we do not know if it has been studied before.

```
clutterMu /@ allNormalSpanningClutters[4]
```

```
{1, -1, -1, -1, 0, 1, 2, -1, 2, 1, 2, -1, 2, 3}
```

■ Additional Considerations

A *semi-clutter* is any anti-chain of subsets $E \subseteq 2^V$. For each finite set S, let $K(S)$ be the set of semi-clutters spanning S. A *species* [2] is an endofunctor on the category of finite sets and bijections, so here we have defined a species of semi-clutters. The compound semi-clutter of a decomposition $R(R_1, \ldots, R_k)$, as defined by Billera [3], is obtained as a disjoint "sum" of Cartesian "products." Interpreted in the language of species theory, this is a certain natural transformation

$$\text{com} : K \odot K \to K,$$

where $\odot$ denotes the composition operation on species, a generalization of composition of exponential formal power series. Let $C(S)$ be the set of (connected) clutters spanning S, let $T(S)$ be the set $\{\{S\}\}$ containing only the maximal clutter on S, and let $P(S)$ be the set of clutters having no expression as a compound of a proper decomposition (i.e. P is the species of "prime" clutters). Billera's main theorem (attributed to Shapley) establishes a unique reduced compound representation, which is itself a species of decompositions

$$\text{com}(T \odot P + P \odot K) = C.$$

From this it is evident that $K = 1 \odot C$ can ultimately be reduced to a nested compound expression using only trivial and prime clutters. Hence the problem of enumerating semi-clutters on a vertex set is reduced to the problem of constructing, for any connected clutter, its "maximal proper committees," which is the nontrivial solution of [2] for the enumeration of prime clutters considered. This is a particularly interesting application of formal species.

References

[1] R. Bacher, "On the Enumeration of Labelled Hypertrees and of Labelled Bipartite Trees." arxiv.org/abs/1102.2708.

[2] A. Joyal, "Une théorie combinatoire des séries formelles," *Advances in Mathematics*, **42**(1), 1981 pp. 1–82. doi:10.1016/0001-8708(81)90052-9.

[3] L. J. Billera, "On the Composition and Decomposition of Clutters," *Journal of Combinatorial Theory*, *Series B*, **11**(3), 1971 pp. 234–245. doi:10.1016/0095-8956(71)90033-5.

G. Wiseman, "Every Clutter Is a Tree of Blobs," *The Mathematica Journal*, 2017.
dx.doi.org/doi:10.3888/tmj.19-7.

About the Author

The author is a former graduate student of pure mathematics at the University of California, Davis. He is interested in categorical technology applied to discrete mathematics.

Gus Wiseman
gus@nafindix.com