

A combinatorial view on derived codes

Relinde Jurrius
(joint work with Philippe Cara)

University of Neuchâtel, Switzerland

Combinatorics 2016
May 30, 2016

Codes and lattices

Linear code k -dim subspace of $\text{GF}(q)^n$

Arrangement of hyperplanes n -tuple of hyperplanes in $\text{GF}(q)^k$

Projective system n -tuple of points in $\text{PG}(k - 1, q)$.

- One-to-one correspondence between equivalence classes.
- Independent of choice of generator matrix, so notation: \mathcal{A}_C or \mathcal{P}_C .

Codes and lattices

Lattice: poset with *join* (smallest upper bound), $x \vee y$
and *meet* (greatest lower bound) $x \wedge y$

Geometric lattice:

- atomic: every element is join of rank-1 elements (*atoms*)
- semimodular: rank function with $r(x \vee y) + r(x \wedge y) \leq r(x) + r(y)$
- no infinite chains

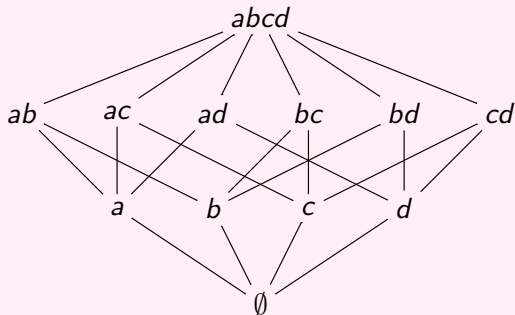
Codes and lattices

Projective system: atoms = points, elements = spans

Hyperplane arrangement: atoms = hyperplanes, elements = intersections

Example

$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$$



Derived code

- Start with $[n, k]$ code.
- Consider the projective system \mathcal{P}_C .
- Look at all hyperplanes spanned by $k - 1$ points of \mathcal{P}_C .
(Ignore $k - 1$ points that span spaces of lower dimension.)
- Remove (multiple) copies of hyperplanes.
- These hyperplanes form an arrangement \mathcal{A} .
- The *derived code* $D(C)$ is the code such that $\mathcal{A} = \mathcal{A}_{D(C)}$.

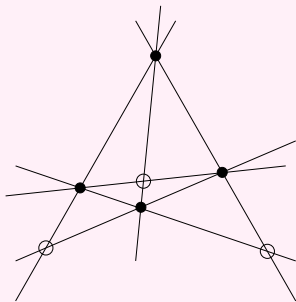
Derived code

Example

$$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & 0 & 1 & -1 & 1 \\ 0 & -1 & 1 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 \end{pmatrix}$$

C

$D(C)$



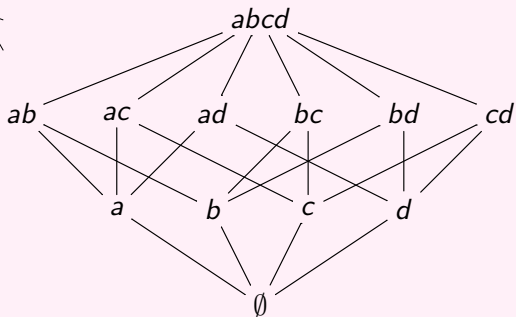
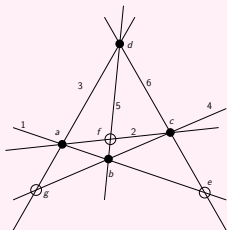
Derived code

From a lattice point of view, this is how we make a derived code:

- Start with $[n, k]$ code.
- Consider the geometric lattice \mathcal{L} of \mathcal{P}_C .
- Turn it upside down: \mathcal{L}^{op} .
- Add extra elements above atom level such that
 - $\mathcal{L}^{op} \hookrightarrow \mathcal{L}(D(C))$ in the “right” way;
 - $\mathcal{L}(D(C))$ is a geometric lattice.

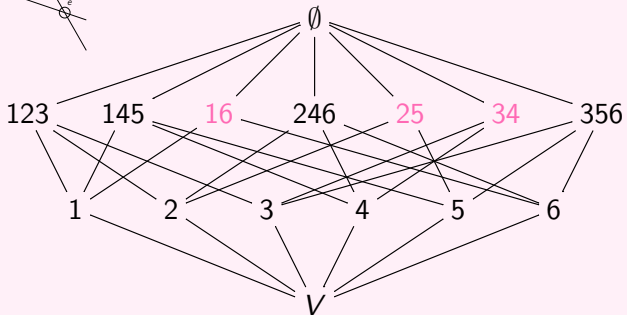
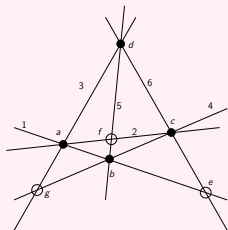
Derived code

Example



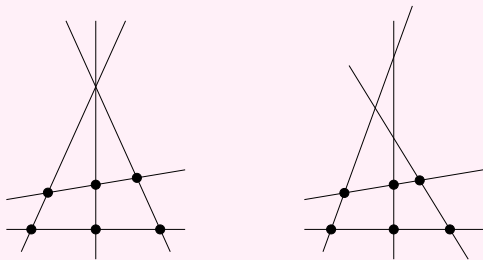
Derived code

Example



Derived code

Example



Codes with equal geometric lattices may have different derived codes!

Results

We answered two open questions:

- When is $C \cong D(C)$?
- Can we define a *derived lattice*, by taking the derived arrangement “as general as possible”?

When is $C \cong D(C)$?

We need: $\mathcal{L}(C) = \mathcal{L}(C)^{op} = \mathcal{L}(D(C))$ and \mathcal{L}^{op} is a geometric lattice.

Theorem

The following are equivalent:

- $C \cong D(C)$
- $r(x \vee y) + r(x \wedge y) = r(x) + r(y)$
- \mathcal{P}_C contains all points of some $\text{PG}(k - 1, q)$
- C is the q -ary simplex code

Derived lattice

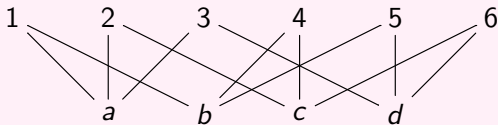
How to create the *derived lattice* $D(\mathcal{L})$:

- Start with a geometric lattice \mathcal{L} .
- The atoms of $D(\mathcal{L})$ are the co-atoms of \mathcal{L} .
- For all subsets I of atoms:
 - If $r(\mathcal{L}) - r(\bigwedge I) \leq |I|$ in \mathcal{L} , then $\bigvee I = (\bigwedge I)^{op}$ in $D(\mathcal{L})$.
 - If $r(\mathcal{L}) - r(\bigwedge I) > |I|$ in \mathcal{L} , then $\bigvee I$ is a new element in $D(\mathcal{L})$ with $r^*(\bigvee I) = |I|$.
- Partial ordering: $\bigvee I \leq \bigvee J$ iff $I \subseteq J$.

Rank function: $r^*(\bigvee I) = \min\{r(\mathcal{L}) - r(\bigwedge I), |I|\}$

Derived lattice

Example



$$I = \{1, 2\}$$

$$3 - r(a) = 3 - 1 = 2$$

$$2 \leq 2$$

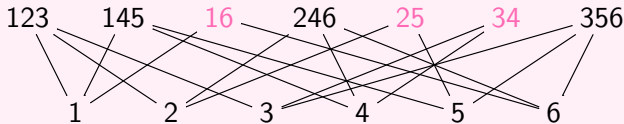
→ no new element

$$I = \{1, 6\}$$

$$3 - r(0) = 3 - 0 = 3$$

$$3 > 2$$

→ new element



$$I = \{1, 5, 6\}$$

$$3 - r(0) = 3 - 0 = 3$$

$$3 \leq 3$$

→ no new element

Derived lattice

Steps of the proof:

- $D(\mathcal{L})$ is a lattice
- $\mathcal{L}^{op} \hookrightarrow D(\mathcal{L})$ in the “right” way
- $D(\mathcal{L})$ is a geometric lattice

Difficult part: semimodularity $r^*(x \vee y) + r^*(x \wedge y) \leq r^*(x) + r^*(y)$

Thank you for your attention.