A Discrete Signal Processing Framework for Meet/Join Lattices

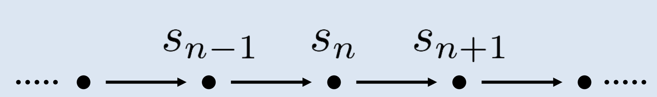
with Applications to Hypergraphs and Trees

ETH zürich

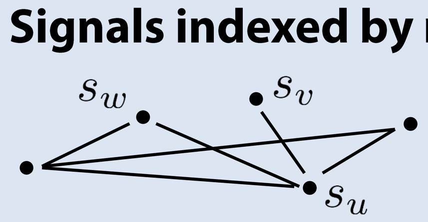
Markus Püschel, Computer Science, ETH Zürich

Goal

Classical DSP: Signals indexed by time



Graph DSP: Signals indexed by nodes of graphs



New Discrete Lattice SP:

Signals indexed by a meet/join lattice

- Shift
- Convolution/filtering
- Fourier transform
- Frequency response

Derivation: Algebraic signal processing (ASP)

shift ⟶ convolution ⟶ Fourier transform etc.

Meet/Join Lattices

Meet Semilattice

L a finite set with partial order $\leq$: for all $a, b, c \in L$,

- $a \leq a$
- $a \leq b, b \leq a$ implies $a = b$
- $a \leq b, b \leq c$ implies $a \leq c$

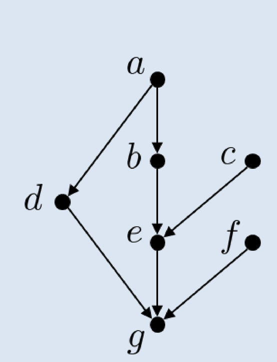
Meet operation: $a \wedge b$ is largest element $\leq a$ and $\leq b$

Join Semilattice

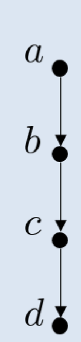
Analogous with $a \vee b$, smallest element $\geq a$ and $\geq b$

Lattice: Meet + Join Semilattice

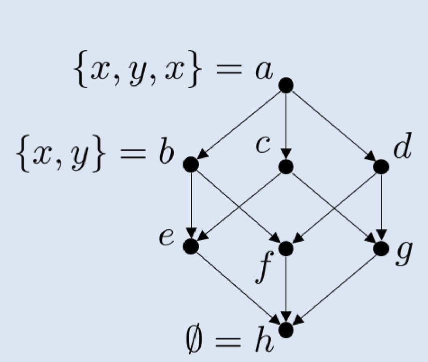
Examples



some meet semilattice



total order lattice



subset lattice of 3-element set {x,y,z}
$\wedge = \cap$

Visualization as graph (a special type)

Meet/join SL has always a minimal/maximal element

Discrete-Lattice SP

Assume a semilattice L with n elements

Signal

$$\mathbf{s} = (s_x)_{x \in L} \in \mathbb{R}^n$$

Shifts by $q \in L$

$$(s_x)_{x \in L} \mapsto (s_{x \wedge q})_{x \in L}$$

Basic (generating) shifts = meet-irreducible elements

Convolution/filtering Filter $\mathbf{h} = (h_q)_{q \in L}$

$$\mathbf{h} * \mathbf{s} = \left(\sum_{q \in L} h_q s_{x \wedge q}\right)_{x \in L}$$

linear, shift(s)-invariant

Pure frequencies/frequency response

eigenvectors $\mathbf{f}^y = (\iota_{y \leq x})_{x \in L}, \quad y \in L$

characteristic function

eigenvalues $\overline{h}_y = \sum_{q \in L,\ y \leq q} h_q$

Fourier transform (Discrete lattice transform)

$$\mathrm{DLT}_L^{-1} = (\mathbf{f}^y)_{y \in L}$$

$$\widehat{s}_y = \sum_{x \leq y} \mu(x, y) s_x$$

Moebius function
$$\mu(x,x) = 1, \qquad \text{for } x \in L$$
$$\mu(x,y) = -\sum_{x \leq z < y} \mu(x,z), \qquad x \neq y$$

Inversion formula provided by lattice theory

Convolution theorem

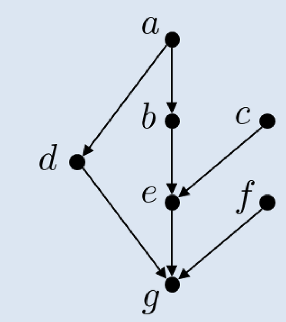
$$\widehat{\mathbf{h} * \mathbf{s}} = \overline{\mathbf{h}} \odot \widehat{\mathbf{s}}$$

Discrete-lattice SP versus discrete-time SP

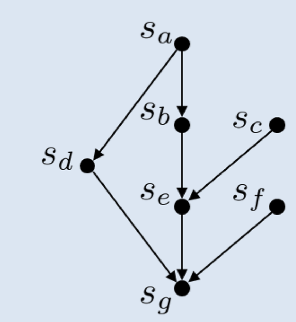
Concept	DLSP	DTSP
Signal	$(s_x)_{x \in L}$	$(s_k)_{k \in [n]}$
Filter	$(h_q)_{q \in L}$	$(h_m)_{m \in [n]}$
Basic shifts	$(s_{x \wedge b})_{x \in L},\ b$ meet irred.	$(s_{k-1})_{k \in [n]}$
Convolution	$\sum_{q \in L} h_q s_{x \wedge q}$	$\sum_{0 \leq m < n} h_m s_{k-m}$
Pure frequency	$(\iota_{y \leq x})_{x \in L},\ y \in L$	$\frac{1}{n}(\omega_n^{-k\ell})_{k \in [n]},\ \ell \in [n]$
Fourier transform	$\widehat{s}_y = \sum_{x \leq y} \mu(x,y) s_x$	$\widehat{s}_\ell = \sum_{k \in [n]} \omega_n^{k\ell} s_k$

Example

Signal



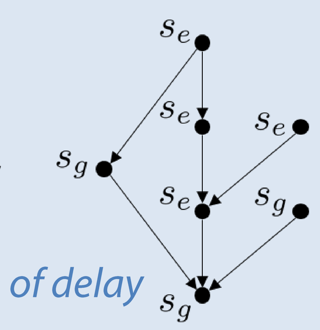
lattice



signal on lattice

Shifts by e

$$(s_{x \wedge e})_{x \in L}$$



a form of delay

$$\begin{bmatrix} 0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1 \end{bmatrix}$$

shift matrix for *e*

Basic shifts: a, b, c, d, f

E.g., $e = a \wedge c$ is not meet-irreducible

Pure frequencies

$$\mathrm{DLT}_L^{-1} = (\mathbf{f}^y)_{y \in L} = \begin{bmatrix} 1&1&0&1&1&0&1 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&1&0&1 \\ 0&0&0&1&0&0&1 \\ 0&0&0&0&1&0&1 \\ 0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&1 \end{bmatrix}$$

Fourier transform (Discrete lattice transform)

$$\mathrm{DLT}_L = \begin{bmatrix} 1&-1&0&-1&0&0&1 \\ 0&1&0&0&-1&0&0 \\ 0&0&1&0&-1&0&0 \\ 0&0&0&1&0&0&-1 \\ 0&0&0&0&1&0&-1 \\ 0&0&0&0&0&1&-1 \\ 0&0&0&0&0&0&1 \end{bmatrix}$$

diagonalizes all shifts and all filters

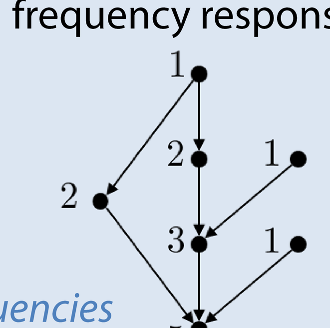
Example

$$\mathrm{DLT}_L \cdot \begin{bmatrix} 0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1 \end{bmatrix} \cdot \mathrm{DLT}_L^{-1} = \mathrm{diag}(0,0,1,0,1,0,1,0)$$

Example: Low-pass filter

$h = a + c + d + f$
(sum of basic shifts)

frequency response



"low" frequencies are amplified

Possible Applications

Set functions (ICASSP 2018, more to come)

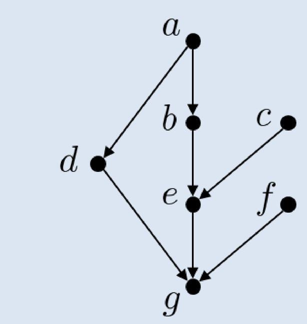
Set function = signal on lattice of subsets

$$\mathbf{s} = (s_A)_{A \subseteq N}$$

N a finite set, $\cap = \wedge$

Signals on graphs representing lattices (e.g., trees)

These have upper triangular adjacency matrices



$$\begin{bmatrix} 0&1&0&1&0&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1 \end{bmatrix}$$

adjacency matrix (graph shift)

Graph DSP	Lattice DSP
Leverages algebraic graph theory	Leverages algebraic lattice theory
One shift captures adjacency structure	Multiple shifts captures lattice structure
No diagonalizing Fourier transform	Fourier transform always exists

Signals on hypergraphs

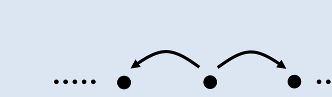
$H = (V, E, \mathbf{s} : E \to \mathbb{R}),\ E \subseteq 2^V$, approximately closed under $\cap$

Algebraic Signal Processing (ASP)

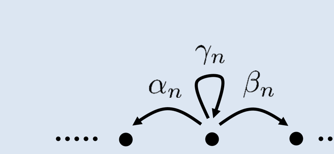
Framework to generalize standard, linear, time DSP using insights from abstract algebra

ASP is constructive: DSP frameworks are derived from shift definition as shown here

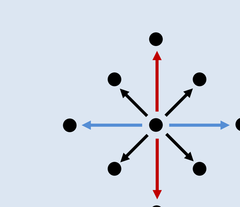
DSP frameworks derived to date:



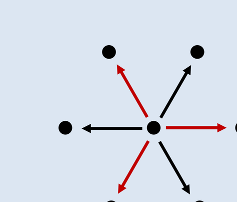
1D space shift (yields DCTs/DST)



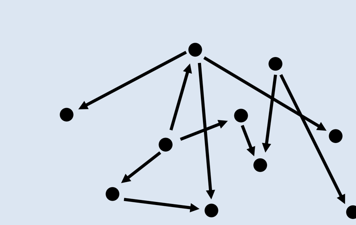
1D generic next neighbor shift



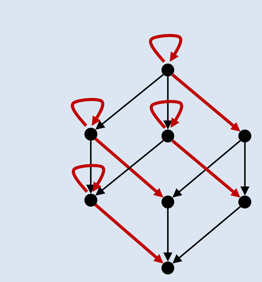
2D quincunx shifts



2D hexagonal shifts



graph shift (yields graph DSP)



set shifts

http://www.ece.cmu.edu/~smart/
https://acl.inf.ethz.ch/research/asp/