Algebraic analysis of complex network structures with `multiplex`

Antonio Rivero Ostoic

University of San Simón

Sunbelt XXXVIII ☆ Utrecht, Netherlands ☆ 28 June 2018

Agenda

Visualization and Algebraic Analysis . . .

1. Introduction (Group structure)

2. Decomposition of a Partially Ordered Semigroup

☛ Computations with `multiplex` and visualization with `multigraph`

First, some terminology...

- **Multiplex** $\leftrightarrow$ **Monoplex** *structure*

  ➠ System with several $\leftrightarrow$ single or collapsed levels in the set of relations

- **multiplex** $\leftrightarrow$ **uniplex** *edge*

  ➠ A relationship with multiple $\leftrightarrow$ single or collapsed levels

- **Multimodal** network

  ➠ Same as Multiplex, but most used with flows or transportation modes

- **Multilevel** network

  ➠ A structure with individual and group levels; i.e. affiliation networks, but *where both level entities can be interrelated*

- **Multilayer** network

  ➠ Cascade structure with multiple subsystems and layers of connectivity

Representations for multiplex networks

Simple networks:

- *(Simple) graphs*, *matrices*

  ➟ for relations between actors

Multiplex networks:

- *Multigraphs*, *arrays*

  ➟ for (types of) relations between actors

- *Cayley graphs*, *tables*

  ➟ for relationships between relations

☛ *Different types of algebraic structures are represented by tables*

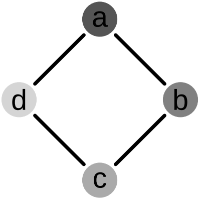
Multiplex networks

Algebraic systems representing multiplex networks:

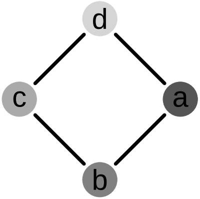
Type of structure	Algebraic object
Elementary	*Group*
Complex	*Semigroup*, *Semiring*, *Lattice*, etc.

Example of elementary group structure

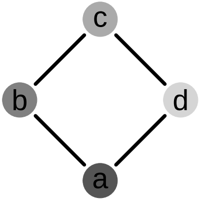
Dihedral group of the square, D_4



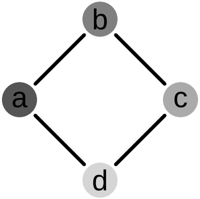
Identity



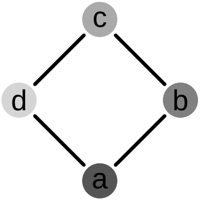
Rotation R_{90}



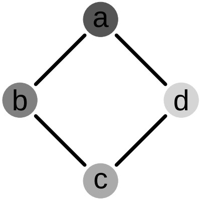
R_{90^2}



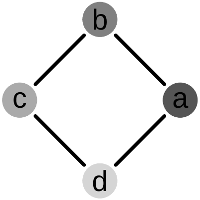
R_{90^3}



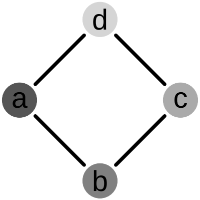
Reflection S_H



S_V



S_D



S_L

Code: Example of elementary group structure

Dihedral group of the square, D_4

```
# CONSTRUCT MATRIX REPRESENTING A SQUARE
R> square <- transf(c("a, b", "b, c", "c, d", "d, a"), type = "toarray")
```

```
# DEFINE THE SCOPE OF THE GRAPH
R> scp <- list(directed = FALSE, cex = 16, vcol = gray(2:5/6), clu = 1:4,
+    lwd = 10, ecol = "black", tcex = 6, pos = 0)
```

```
# IDENTITY
R> multigraph(square, scope = scp)
# ROTATIONS
R> multigraph(square, scope = scp, rot = 90)
R> multigraph(square, scope = scp, rot = 90*2)
R> multigraph(square, scope = scp, rot = 90*3)
# REFLECTIONS
R> multigraph(square, scope = scp, mirrorH = TRUE)
R> multigraph(square, scope = scp, mirrorV = TRUE)
R> multigraph(square, scope = scp, mirrorD = TRUE)
R> multigraph(square, scope = scp, mirrorL = TRUE)
```

Multiplication table

$\circ$	I	R_{90}	R_{90^2}	R_{90^3}	S_H	S_L	S_V	S_D
I	I	R_{90}	R_{90^2}	R_{90^3}	S_H	S_L	S_V	S_D
R_{90}	R_{90}	R_{90^2}	R_{90^3}	I	S_L	S_V	S_D	S_H
R_{90^2}	R_{90^2}	R_{90^3}	I	R_{90}	S_V	S_D	S_H	S_L
R_{90^3}	R_{90^3}	I	R_{90}	R_{90^2}	S_D	S_H	S_L	S_V
S_H	S_H	S_D	S_V	S_L	I	R_{90^3}	R_{90^2}	R_{90}
S_L	S_L	S_H	S_D	S_V	R_{90}	I	R_{90^3}	R_{90^2}
S_V	S_V	S_L	S_H	S_D	R_{90^2}	R_{90}	I	R_{90^3}
S_D	S_D	S_V	S_L	S_H	R_{90^3}	R_{90^2}	R_{90}	I

Code: Constructing the multiplication table

```
# FIRST CREATE THE GENERATORS OF D4 AS PERMUTATION MATRICES
R> SD4 <- transf(list(R = c("2, 1", "3, 2", "4, 3", "1, 4"),
+       S = c("1, 3", "2, 2", "3, 1", "4, 4")), type = "toarray", sort = TRUE)
```

```
, , R

  1 2 3 4
1 0 0 0 1
2 1 0 0 0
3 0 1 0 0
4 0 0 1 0

, , S

  1 2 3 4
1 0 0 1 0
2 0 1 0 0
3 1 0 0 0
4 0 0 0 1
```

Code: Constructing the multiplication table

"Because all groups are semigroups as well..."

```
# THE GROUP STRUCTURE WITH SYMBOLIC FORMAT
R> semigroup(SD4, type = "symbolic")
```

```
...

$st
[1] "R"   "S"   "RR"  "RS"  "SR"  "SS"  "RRR" "RRS"

$S
      R   S  RR  RS  SR  SS RRR RRS
R    RR  RS RRR RRS   S   R  SS  SR
S    SR  SS RRS RRR   R   S  RS  RR
RR  RRR RRS  SS  SR  RS  RR   R   S
RS    S   R  SR  SS  RR  RS RRS RRR
SR  RRS RRR  RS  RR  SS  SR   S   R
SS    R   S  RR  RS  SR  SS RRR RRS
RRR  SS  SR   R   S RRS RRR  RR  RS
RRS  RS  RR   S   R RRR RRS  SR  SS

attr(,"class")
[1] "Semigroup" "symbolic"
```

Group structure in social networks?

- Despite the symmetry, algebraic groups can model human societies

- Some kinship networks from primitive societies have specific rules of marriage & descent that follow the elementary group structure

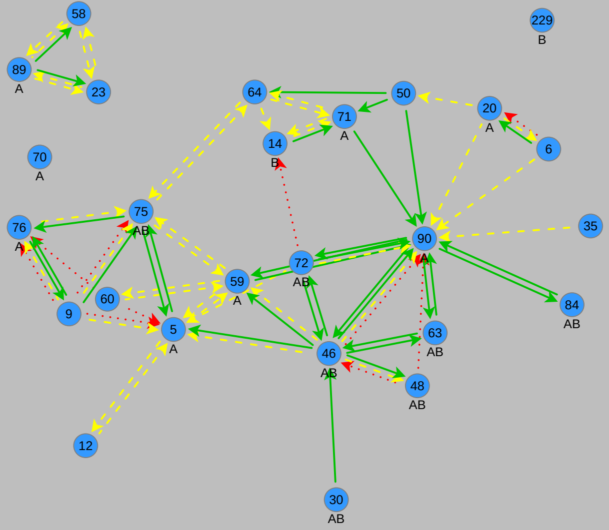
  ➟ Kariera and Arunta from Western Australia are classical examples

- However, since most of multiplex social networks are not symmetric they represent *complex* structures

☛ from a group to a **semigroup** *partially ordered*

Example: Incubator network "A", $\mathscr{X}_A$

Collaboration (green), Friendship (yellow), Competition (red)



Tie interlock

- **Social structure** = Ties between actors
- **Relational structure** = Interrelations between relations
- **Role structure** = Relational system of aggregated relations

☛ we benefit from algebraic structures to represent relational systems

Positional System of Incubator network “A”

After applying Compositional Equivalence with Relational contrast to $\mathscr{X}_A$:

```
# LOAD DATA SET AND RECORD THE IMAGE MATRICES
R> data("incubA")
R> net <- incubA$IM
```

C

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

F

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

K

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

A

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

D

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

G

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

L

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

B

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

Role Structure of $\mathscr{X}_A$

Algebraic constraint: Role table

```
R> semigroup(net)


...
$S
    1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
1   2  2  8  9  9 10  1 10  9 10 17 10  9  9 17  9 17 17 17 17
2   2  2 10  9  9 10  2 10  9 10 17 10  9  9 17  9 17 17 17 17
3   8 10  3 11 11 12  3  8 17 10 11 12 18 17 11 18 17 18 18 17
4  13 14 15  5  5 15  4 19  9 20 11 15 16  9 11 16 17 18 18 17
5  16  9 11  5  5 11  5 18  9 17 11 11 16  9 11 16 17 18 18 17
6   8 10 12 15 11  6  6  8 17 10 11 12 19 20 15 18 17 18 19 20
7   1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
8  10 10  8 17 17 10  8 10 17 10 17 10 17 17 17 17 17 17 17 17
9   9  9 17  9  9 17  9 17  9 17 17 17  9  9 17  9 17 17 17 17
10 10 10 10 17 17 10 10 10 17 10 17 10 17 17 17 17 17 17 17 17
11 18 17 11 11 11 11 11 18 17 17 11 11 18 17 11 18 17 18 18 17
12  8 10 12 11 11 12 12  8 17 10 11 12 18 17 11 18 17 18 18 17
13 14 14 19  9  9 20 13 20  9 20 17 20  9  9 17  9 17 17 17 17
14 14 14 20  9  9 20 14 20  9 20 17 20  9  9 17  9 17 17 17 17
15 19 20 15 11 11 15 15 19 17 20 11 15 18 17 11 18 17 18 18 17
16  9  9 18  9  9 17 16 17  9 17 17 17  9  9 17  9 17 17 17 17
17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
18 17 17 18 17 17 17 18 17 17 17 17 17 17 17 17 17 17 17 17 17
19 20 20 19 17 17 20 19 20 17 20 17 20 17 17 17 17 17 17 17 17
20 20 20 20 17 17 20 20 20 17 20 17 20 17 17 17 17 17 17 17 17

attr(,"class")
[1] "Semigroup" "numerical"
```

Role Structure of $\mathscr{X}_A$

Algebraic constraint: Set of equations

```
R> netst <- strings(net, equat = TRUE, k = 3)
R> netst$equat
```

```
...
$equat
$equat$C
[1] "C"   "CA"  "AC"  "AAC" "CAA" "ACA"

$equat$F
 [1] "F"   "CC"  "FF"  "CF"  "FC"  "FA"  "AF"  "CCC" "FFC" "CFF" "CCF" "FFF"
[13] "AAF" "FCC" "FAA" "CCA" "FFA" "ACC" "AFF" "FCF" "CFC" "AFA" "CAC" "FAF"
[25] "CFA" "CAF" "FCA" "FAC" "ACF" "AFC"

$equat$K
 [1] "K"   "KK"  "KA"  "AK"  "KKK" "AAK" "KAA" "KKA" "AKK" "AKA" "KAK"

$equat$D
[1] "D"   "DA"  "AD"  "AAD" "DAA" "ADA"

$equat$G
 [1] "G"   "DD"  "GG"  "DG"  "GD"  "GA"  "AG"  "DDD" "GGD" "DGG" "DDG" "GGG"
[13] "AAG" "GDD" "GAA" "DDA" "GGA" "ADD" "AGG" "GDG" "DGD" "AGA" "DAD" "GAG"
[25] "DGA" "DAG" "GDA" "GAD" "ADG" "AGD"

$equat$L
 [1] "L"   "LL"  "LA"  "AL"  "LLL" "AAL" "LAA" "LLA" "ALL" "ALA" "LAL"

$equat$A
[1] "A"   "AA"  "AAA"
```

Role Structure of $\mathscr{X}_A$

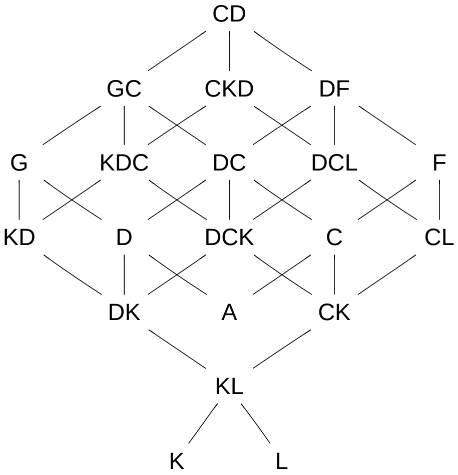
Algebraic constraint: Set of inclusions

```
R> partial.order(netst, type = "strings")


    C F K D G L A CK CD CL KD KL DC DF DK GC CKD KDC DCK DCL
C   1 1 0 0 0 0 0  0  1  0  0  0  1  1  0  1   0   0   0   0
F   0 1 0 0 0 0 0  0  1  0  0  0  0  1  0  0   0   0   0   0
K   1 1 1 1 1 0 0  1  1  1  1  1  1  1  1  1   1   1   1   1
D   0 0 0 1 1 0 0  0  1  0  0  0  1  1  0  1   0   0   0   0
G   0 0 0 0 1 0 0  0  1  0  0  0  0  0  0  1   0   0   0   0
L   1 1 0 1 1 1 0  1  1  1  1  1  1  1  1  1   1   1   1   1
A   1 1 0 1 1 0 1  0  1  0  0  0  1  1  0  1   0   0   0   0
CK  1 1 0 0 0 0 0  1  1  1  0  0  1  1  0  1   1   1   1   1
CD  0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0   0
CL  0 1 0 0 0 0 0  0  1  1  0  0  0  1  0  0   1   0   0   1
KD  0 0 0 0 1 0 0  0  1  0  1  0  0  0  0  1   1   1   0   0
KL  1 1 0 1 1 0 0  1  1  1  1  1  1  1  1  1   1   1   1   1
DC  0 0 0 0 0 0 0  0  1  0  0  0  1  1  0  1   0   0   0   0
DF  0 0 0 0 0 0 0  0  1  0  0  0  0  1  0  0   0   0   0   0
DK  0 0 0 1 1 0 0  0  1  0  1  0  1  1  1  1   1   1   1   1
GC  0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  1   0   0   0   0
CKD 0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  0   1   0   0   0
KDC 0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  1   1   1   0   0
DCK 0 0 0 0 0 0 0  0  1  0  0  0  1  1  0  1   1   1   1   1
DCL 0 0 0 0 0 0 0  0  1  0  0  0  0  1  0  0   1   0   0   1
attr(,"class")
[1] "Partial.Order" "strings"
```

Role Structure of $\mathscr{X}_A$

Algebraic constraint: Set of inclusions (visualization)

```
R> diagram(partial.order(netst, type = "strings"))
```



Decomposition of Relational Structures

Abstract Semigroup

```
# ROLE STRUCTURE OF 𝒳_A
R> S <- semigroup(net, type = "symbolic")
```

```
# DECOMPOSE AND REDUCE WITH CONGRUENCE CLASSES IN THE ABSTRACT SEMIGROUP
R> decomp(S, cngr(S, uniq = TRUE), type = "cc", reduc = TRUE)


...

$IM[[11]]
  A C
A A C
C C C

$IM[[12]]
    C KD KL CD  K  D  L  A
C   C CD  C CD  C CD  C  C
KD CD KD KD CD KD KD KD KD
KL  C KD KL CD KL KD KL KL
CD CD CD CD CD CD CD CD CD
K   C KD KL CD  K KD KL  K
D  CD KD KD CD KD  D KD  D
L   C KD KL CD KL KD  L  L
A   C KD KL CD  K  D  L  A

$ord
 [1] 10  6  8  7  5  6  5  8  2  2  2  8
```

Decomposition of Relational Structures

Partially Ordered Semigroup

```
# PARTIAL ORDER STRUCTURE OF STRINGS
R> PO <- partial.order(netst, type = "strings")


    C F K D G L A CK CD CL KD KL DC DF DK GC CKD KDC DCK DCL
C   1 1 0 0 0 0 0  0  1  0  0  0  1  1  0  1   0   0   0   0
F   0 1 0 0 0 0 0  0  1  0  0  0  0  1  0  0   0   0   0   0
K   1 1 1 1 1 0 0  1  1  1  1  1  1  1  1  1   1   1   1   1
D   0 0 0 1 1 0 0  0  1  0  0  0  1  1  0  1   0   0   0   0
G   0 0 0 0 1 0 0  0  1  0  0  0  0  0  0  1   0   0   0   0
L   1 1 0 1 1 1 0  1  1  1  1  1  1  1  1  1   1   1   1   1
A   1 1 0 1 1 0 1  0  1  0  0  0  1  1  0  1   0   0   0   0
CK  1 1 0 0 0 0 0  1  1  1  0  0  1  1  0  1   1   1   1   1
CD  0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  0   0   0   0   0
CL  0 1 0 0 0 0 0  0  1  1  0  0  0  1  0  0   1   0   0   1
KD  0 0 0 0 1 0 0  0  1  0  1  0  0  0  0  1   1   1   0   0
KL  1 1 0 1 1 0 0  1  1  1  1  1  1  1  1  1   1   1   1   1
DC  0 0 0 0 0 0 0  0  1  0  0  0  1  1  0  1   0   0   0   0
DF  0 0 0 0 0 0 0  0  1  0  0  0  0  1  0  0   0   0   0   0
DK  0 0 0 1 1 0 0  0  1  0  1  0  1  1  1  1   1   1   1   1
GC  0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  1   0   0   0   0
CKD 0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  0   1   0   0   0
KDC 0 0 0 0 0 0 0  0  1  0  0  0  0  0  0  1   1   1   0   0
DCK 0 0 0 0 0 0 0  0  1  0  0  0  1  1  0  1   1   1   1   1
DCL 0 0 0 0 0 0 0  0  1  0  0  0  0  1  0  0   1   0   0   1
attr(,"class")
[1] "Partial.Order" "strings"
```

Factorization

Partially Ordered Semigroup

☛ Function `fact` performs the full factorization option from PACNET

```
# FACTORIZATION BY INDUCED INCLUSIONS
R> POSii <- fact(S, PO)
```

```
# PARTITION—RELATIONS
R> POSpr <- pi.rels(POSii, PO, po.incl = TRUE)
```

```
# DECOMPOSITION WITH MEET—COMPLEMENTS
R> Sfac <- decomp(S, POSpr, type = "mc", reduc = TRUE)
```

Factorization of the Partially Ordered Semigroup

Induced inclusions

```
R> POSii$ii


$`2, 1`
 [1] "10, 1"  "10, 13" "10, 16" "10, 18" "10, 19" "10, 8"  "14, 13" "14, 16"
 [9] "17, 16" "17, 18" "2, 1"   "2, 13"  "2, 16"  "20, 13" "20, 16" "20, 18"
[17] "20, 19" "9, 16"

$`4, 1`
 [1] "11, 1"  "11, 10" "11, 13" "11, 14" "11, 19" "11, 2"  "11, 20" "11, 8"
 [9] "13, 2"  "14, 2"  "15, 1"  "15, 10" "15, 2"  "15, 8"  "16, 14" "16, 2"
[17] "17, 10" "17, 14" "17, 2"  "17, 20" "18, 10" "18, 14" "18, 2"  "18, 20"
[25] "19, 10" "19, 2"  "20, 10" "20, 2"  "4, 1"   "4, 2"   "5, 13"  "5, 14"
[33] "5, 2"   "9, 14"  "9, 2"

$`5, 1`
 [1] "11, 1"  "11, 10" "11, 13" "11, 14" "11, 19" "11, 2"  "11, 20" "11, 8"
 [9] "13, 2"  "14, 2"  "15, 1"  "15, 10" "15, 2"  "15, 8"  "16, 14" "16, 2"
[17] "17, 10" "17, 14" "17, 2"  "17, 20" "18, 10" "18, 14" "18, 2"  "18, 20"
[25] "19, 10" "19, 2"  "20, 10" "20, 2"  "4, 1"   "4, 2"   "5, 1"   "5, 13"
[33] "5, 14"  "5, 2"   "9, 14"  "9, 2"

...
```

Factorization of the Partially Ordered Semigroup

Atoms

```
R> POSii$at
```

```
$`11, 14`
[1] "11, 14" "11, 20" "17, 14" "17, 20" "18, 14" "18, 20"

$`10, 16`
[1] "10, 16" "10, 18" "17, 16" "17, 18" "20, 16" "20, 18"

$`1, 17`
[1] "1, 17"  "13, 17" "14, 17" "16, 17" "2, 17"  "4, 17"  "5, 17"  "7, 17"
[9] "9, 17"
```

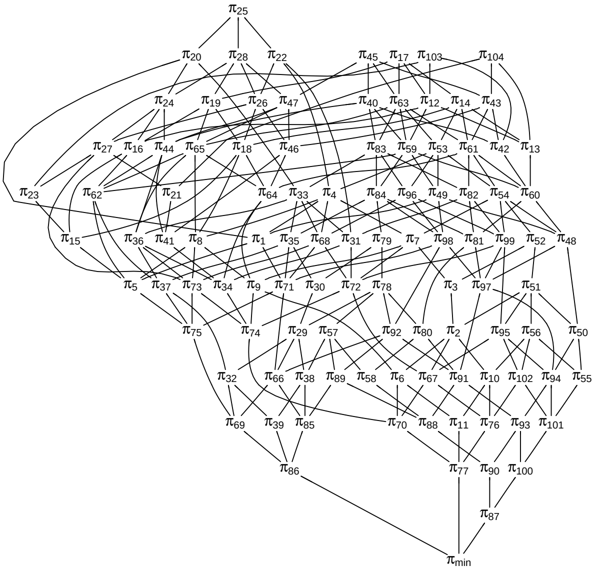
Factorization of the Partially Ordered Semigroup

Meet-Complements of the Atoms

```
R> POSii$mc


[[1]]
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1  1 1 1 1 1 0 0 1 1  1  1  1  1  1  1  1  1  1  1  1
2  1 1 1 1 1 0 0 1 1  1  1  1  1  1  1  1  1  1  1  1
3  1 1 1 1 1 0 0 1 1  1  1  1  1  1  1  1  1  1  1  1
4  0 0 0 1 1 0 0 0 1  0  1  0  1  1  1  1  1  1  1  1
5  0 0 0 0 1 0 0 0 1  0  1  0  0  0  0  1  1  1  0  0
6  1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
7  1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
8  1 1 1 1 1 0 0 1 1  1  1  1  1  1  1  1  1  1  1  1
9  0 0 0 0 1 0 0 0 1  0  1  0  0  0  0  1  1  1  0  0
10 1 1 1 1 1 0 0 1 1  1  1  1  1  1  1  1  1  1  1  1
11 0 0 0 0 1 0 0 0 1  0  1  0  0  0  0  1  1  1  0  0
12 1 1 1 1 1 0 0 1 1  1  1  1  1  1  1  1  1  1  1  1
13 0 0 0 1 1 0 0 0 1  0  1  0  1  1  1  1  1  1  1  1
14 0 0 0 1 1 0 0 0 1  0  1  0  1  1  1  1  1  1  1  1
15 0 0 0 1 1 0 0 0 1  0  1  0  1  1  1  1  1  1  1  1
16 0 0 0 0 1 0 0 0 1  0  1  0  0  0  0  1  1  1  0  0
17 0 0 0 0 1 0 0 0 1  0  1  0  0  0  0  1  1  1  0  0
18 0 0 0 0 1 0 0 0 1  0  1  0  0  0  0  1  1  1  0  0
19 0 0 0 1 1 0 0 0 1  0  1  0  1  1  1  1  1  1  1  1
20 0 0 0 1 1 0 0 0 1  0  1  0  1  1  1  1  1  1  1  1

[[2]]
...
```

Lattice of Congruence Relations

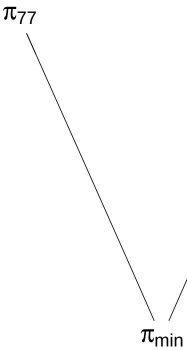
Factorization of the Partially Ordered Semigroup



(Relational) Role Interlock

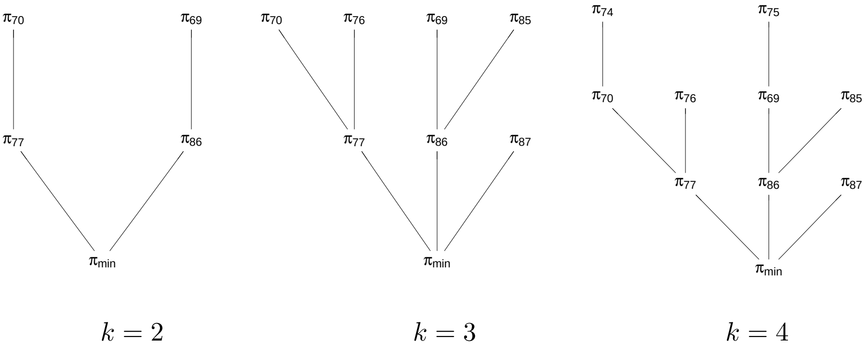
Factorization of the Partially Ordered Semigroup

```
# POTENTIAL ATOMS AS THE K—SHORTEST INDUCED INCLUSIONS IN THE FACTORIZATION
POSpatm <- fact(S, PO, patm = TRUE, K = 1)$patm

# PLOT HASSE DIAGRAM OF SELECTED π—RELATIONS WITH THE PARTIAL ORDER
diagram(partial.order(POSpr, type = "pi.rels", po.incl = TRUE, sel = POSpatm))
```



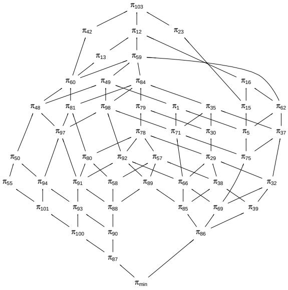
(Relational) Role Interlock

Factorization of the Partially Ordered Semigroup

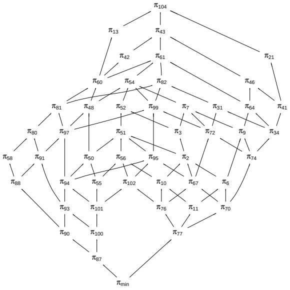


Meet-complements of the Atoms

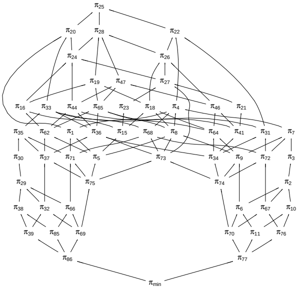
We are interested in the maximal elements in the (complete) lattice



π_{77}



π_{86}



π_{87}

Factorization with Meet-Complements of the Atoms

Clustering of Semigroup elements

```
R> Sfac$clu
```

```
$clu
$clu[[1]]
  C   F   K   D   G   L   A  CK  CD  CL  KD  KL  DC  DF  DK  GC CKD KDC DCK DCL
  1   1   1   2   3   4   4   1   3   1   3   1   2   2   2   3   3   3   2   2

$clu[[2]]
  C   F   K   D   G   L   A  CK  CD  CL  KD  KL  DC  DF  DK  GC CKD KDC DCK DCL
  1   2   3   4   4   4   3   1   2   2   4   4   1   2   4   1   2   1   1   2

$clu[[3]]
  C   F   K   D   G   L   A  CK  CD  CL  KD  KL  DC  DF  DK  GC CKD KDC DCK DCL
  1   1   2   1   1   2   1   2   1   2   2   2   1   1   2   1   2   2   2   2
```

Logics of Interlock

Aggregated Image Matrices

```
R> Sfac$IM
```

```
[[1]]
  C D G L
C C G G C
D D G G D
G G G G G
L C D G L

[[2]]
  C F K D
C F F C F
F F F F F
K C F K D
D C F D D

[[3]]
  C K
C C K
K K K
```

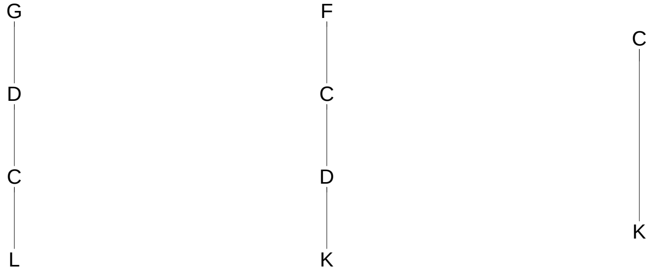
Logics of Interlock

Aggregated Partial Orders

```
R> Sfac$PO

[[1]]
  C D G L
C 1 1 1 0
D 0 1 1 0
G 0 0 1 0
L 1 1 1 1

[[2]]
  C F K D
C 1 1 0 0
F 0 1 0 0
K 1 1 1 1
D 1 1 0 1

[[3]]
  C K
C 1 0
K 1 1
```

Logics of Interlock

Aggregated Partial Orders (Visualization)



π_{77} π_{86} π_{87}

Thanks for your attention!

`github.com/mplex`