Algebraic analysis of complex social networks

· **Workshop 8** ·

Antonio Rivero Ostoic
School of Culture and Society

Networks 2021 Conference ☆ 24 June 2021

AARHUS
UNIVERSITY

Agenda

1. Introduction (plotting multigraphs)
   ➟ Example 1: Monastery novices

2. Elementary structures
   ➟ Example 2: Dihedral group

3. Group structure in social networks
   ➟ Example 3: Kariera kinship

4. Multiplex and signed networks
   ➟ Example 4: Florentine families
   ➟ Example 5: Incubator network A

5. Affiliation and multilevel networks
   ➟ Example 6: Group of Twenty (valued)

1. Introduction

Plotting multigraphs

Example 1: Monastery novices

‘multiplex’ *for computations of multiple networks in* ***R***

Package ‘multiplex’

August 28, 2013

Type Package

Title Analysis of Multiple Social Networks with Algebra

Version 1.0

Depends R (>= 3.0.1)

Date 2013-08-28

Author J. Antonio Rivero Ostoic

Maintainer Antonio Rivero Ostoic <multiplex@post.com>

Description multiplex - Analysis of Multiple Social Networks with Algebra is a package for the study of social systems made of different types of relationships. It is possible to create and manipulate multivariate network data with different formats, and there are effective ways available to treat multiple networks with routines that combine algebraic systems like the partially ordered semigroup or the semiring structure together with the relational bundles occurring in different types of multivariate network data sets.

License GPL-3

Suggests Rgraphviz

Encoding latin1

Collate
'as.semigroup.R' 'as.strings.R' 'bundle.census.R' 'bundles.R''cngr.R' 'convert.R' 'cph.R'

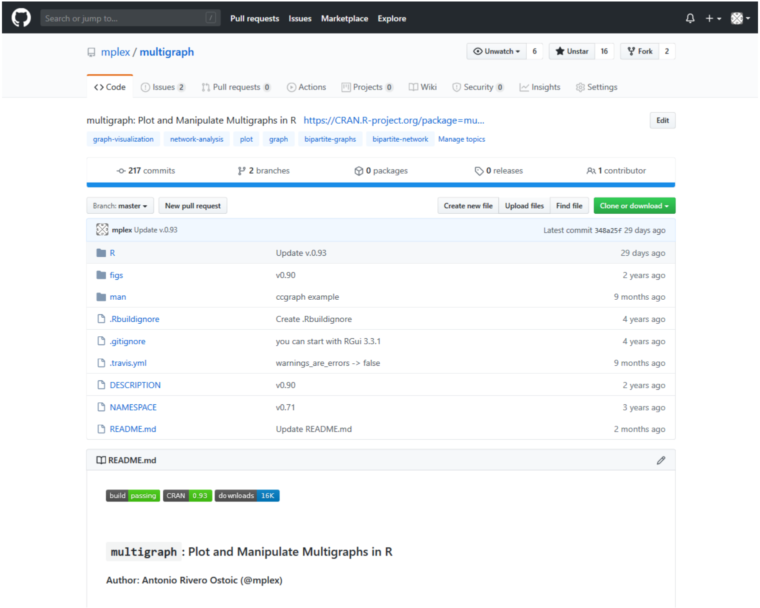
NeedsCompilation no

Repository CRAN

Date/Publication 2013-08-28 13:53:11

R topics documented:

‘multigraph’ *to depict multiplex networks in* ***R***



Getting started: Installation & loading packages

After download **R** and install (with IDE Rstudio if desired):

```r
# install the packages from CRAN
install.packages("multiplex", "multigraph")
# or their beta versions from GitHub
devtools::install_github("mplex/multiplex", ref = "beta")
devtools::install_github("mplex/multigraph", ref = "beta")
```

```r
# load packages
library("multigraph")
# Loading required package: multiplex
```

Representing network data

There are different ways to represent network data in **R**

$(1, 2)$

```r
transf("1, 2")
```

```
  1 2
1 0 1
2 0 0
```

```r
multigraph("1, 2", cex = 18, lwd = 20, rot = -90, pos = 0, vedist = -2)
```

```r
scp <- list(cex = 18, lwd = 20, rot = -90, pos = 0, vedist = -2)
multigraph("1, 2", scope = scp)
```

Representing network data

Undirected

$\{1, 2\}$

```r
matrix(c(0,1,1,0), nrow = 2, ncol = 2)
```

```
     [,1] [,2]
[1,]    0    1
[2,]    1    0
```

```r
multigraph("1, 2", directed = FALSE, scope = scp)
```

Multiplex

$(1, 2); (2, 1)$

```
, , 1

  1 2
1 0 1
2 0 0

, , 2

  1 2
1 0 0
2 1 0
```

```r
multigraph(list("1, 2", "2, 1"), scope = scp, ecol = 1, bwd = .7)
```

Multiplex

$(1, 2); (2, 1)$

```
, , 1

  1 2
1 0 1
2 0 0

, , 2

  1 2
1 0 0
2 1 0
```

```
net <- list("1, 2", "2, 1")
multigraph(net, scope = scp, ecol = 1, bwd = .7, swp = TRUE)
```

Monastery novices: Directed, multiplex, signed, valued, and longitudinal

```
# read Sampson Monastery dataset as Ucinet DL file
samp <- read.dl("http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/sampson.dat")
```

```
# what types of tie the network has?
dimnames(samp)[[3]]
```

```
[1] "SAMPLK1" "SAMPLK2" "SAMPLK3" "SAMPDLK" "SAMPES"  "SAMPDES" "SAMPIN"  "SAMPNIN" "SAMPPR"  "SAMNPR"
```

➠ "like T1-T3", "dislike", "esteem", "disesteem", "influence" (pos/neg), "praise" (pos/neg)

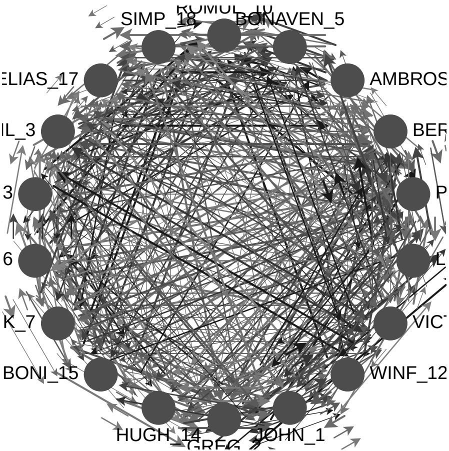
```
# plot Monastery novices network as valued multigraph (default)
multigraph(samp, valued = TRUE)
```

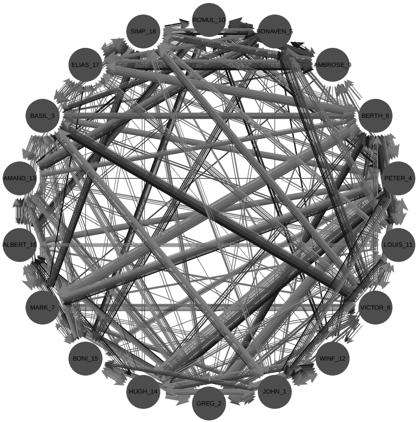
```
# plot valued network with customized values
multigraph(samp, valued = TRUE, bwd = .1, pos = 0, fsize = 6)
```

(Sampson, 1969)

Monastery novices network plot

multigraph circular





Monastery novices: Bundle patterns

```
# enumeration of bundle class types
bundle.census(samp)
```

```
      BUNDLES NULL  ASYMM RECIP T.ENTR T.EXCH MIXED FULL
TOTAL     134   19     20     1     37      8    68    0
```

```
# bundle patterns in the Monastery novices network
summaryBundles(bundles(samp))
```

```
                                                                              Bundles
Asym1                                                  ->{SAMPLK1} (WINF_12, BONAVEN_5)
Asym2                                                   ->{SAMPLK1} (BASIL_3, ROMUL_10)
...                                                                                 ...
Asym20                                                   ->{SAMNPR} (AMAND_13, SIMP_18)
Recp                                                   <->{SAMPLK3} (BONI_15, VICTOR_8)
Tent1                           ->{SAMPDLK} ->{SAMPDES} ->{SAMNPR}  (ALBERT_16, ELIAS_17)
Tent2                ->{SAMPDLK} ->{SAMPDES} ->{SAMPNIN} ->{SAMNPR}  (ALBERT_16, PETER_4)
...                                                                                 ...
```

Monastery novices: Define a system & plot

```
# recall network types of tie
dimnames(samp)[[3]]
```

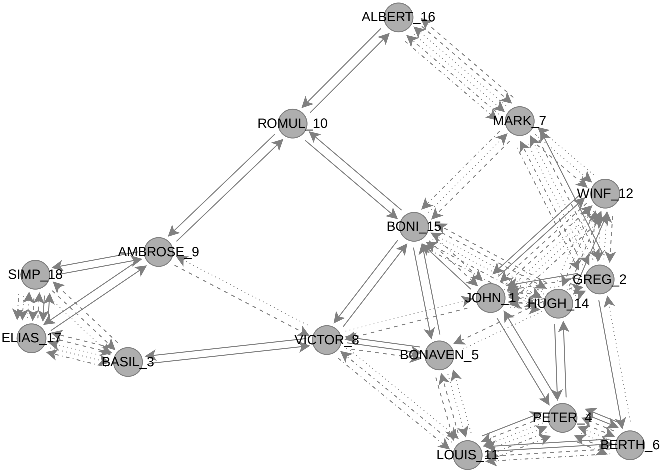
➟ "like T1-T3", "dislike", "esteem", "disesteem", "influence" (pos/neg), "praise" (pos/neg)

```
# extract system of strong bonds having positive ties
sampsb <- rel.sys(samp[,,c(3,5,7,9)], type = "toarray", bonds = "strong")
```

```
# define a new scope
scps <- list(fsize = 8, pos = 0, vcol = "#AEAEAE", vcol0 = "#808080",
             ecol = "#808080", bwd = .4, rot = 145, mirrorY = TRUE)

# plot graph with a reproducible force-directed layout
multigraph(sampsb, layout = "force", seed = 12, scope = scps)
```

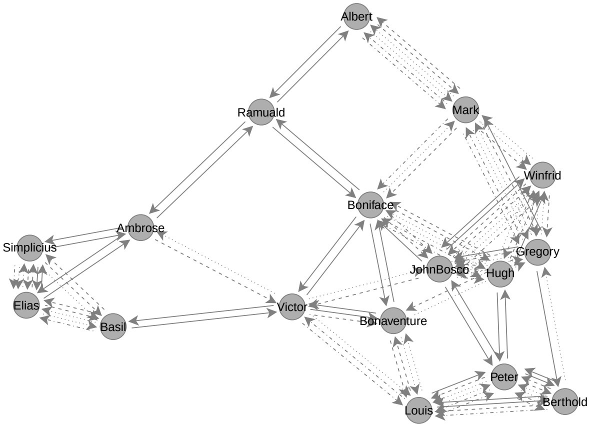
System of strong bonds

Monastery novices network



System of strong bonds: Customized node labels

```
monks <- c("Ramuald","Bonaventure","Ambrose","Berthold","Peter",
+           "Louis","Victor","Winfrid","JohnBosco","Gregory","Hugh",
+           "Boniface","Mark","Albert","Basil","Elias","Simplicius")

multigraph(sampsb, layout = "force", seed = 12, scope = scps, lbs = monks)
```



2. Elementary structures

Example 2: Dihedral groups

Typology of multiple network structures

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*

Algebraic representation of multiplex networks

Typology

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

Group: Elementary structure

A *group* is an algebraic structure with an *element set* and an endowed *operation*:

$$\langle\, G,\ \cdot\, \rangle$$

That for all a, b, c, and a neutral element $e \in G$ satisfies axioms:

$$\begin{aligned} \text{Identity:}\quad & a \cdot e = e \cdot a = a \\ \textit{Inversion}\text{:}\quad & a \cdot a^{-1} = a^{-1} \cdot a = e \\ \text{Associativity:}\quad & (a \cdot b) \cdot c = a \cdot (b \cdot c) \\ \text{Closure:}\quad & a \cdot b \in G \qquad \text{(for all}\quad a, b) \end{aligned}$$

Group structure by permutations

Theorem (Cayley)

All of group theory can be found in permutations.

➡ we focus on permutation symmetry

A *permutation* operator is represented by a *permutation matrix*

➡ having one entry in each row and in each column, and 0 elsewhere

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

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

Group Structures

Definition (Permutation Group on X)

The *permutation group on* X is the set of all permutations S_X on X

Definition (Symmetric Group of order n, S_n)

The *symmetric group* on a n-element set $\{1, 2, \ldots, n\}$ is the set of all permutations with $n!$ bijections σ, $S_n = \{\sigma_1, \sigma_2, \ldots, \sigma_{n!}\}$.

- If $X = \{1, 2, \ldots, n\}$ then $S_X = S_n$
  - ➟ the symmetric groups on n-elements are permutation groups

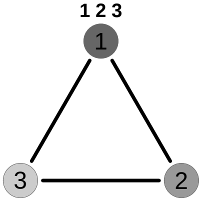
Definition (Dihedral Group of degree n, D_n, $n > 2$)

The set of all permutations which are symmetries on a regular n-sided polygon and the composition operation $\circ$ makes the *dihedral group* $(D_n, \circ)$.

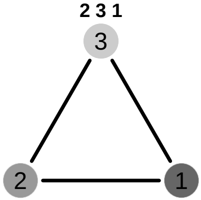
- the order of a dihedral group is twice its degree

Group of symmetries of the equilateral triangle (Dihedral group, D_3)

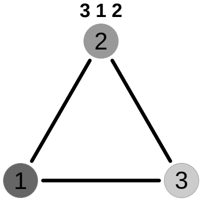
Rotations



R_0

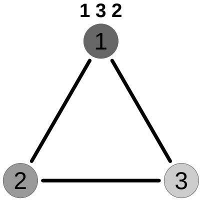


R_{120}

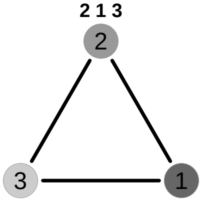


R_{240}

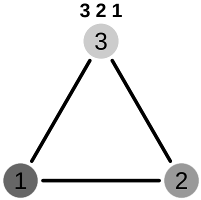
Reflections



S_V



S_D



S_L

Dihedral group, D_3

Cayley table

$\circ$	R_0	R_{120}	R_{240}	S_V	S_D	S_L
R_0	R_0	R_{120}	R_{240}	S_V	S_D	S_L
R_{120}	R_{120}	R_{240}	R_0	S_D	S_L	S_V
R_{240}	R_{240}	R_0	R_{120}	S_L	S_V	S_D
S_V	S_V	S_L	S_D	R_0	R_{240}	R_{120}
S_D	S_D	S_V	S_L	R_{120}	R_0	R_{240}
S_L	S_L	S_D	S_V	R_{240}	R_{120}	R_0

Generators of D_3

as permutation matrices

```
# define generators as permutation matrices with a lexicographic order
D3 <- transf(list(F = c("1, 3","2, 1","3, 2"), G = c("1, 1","2, 3","3, 2")),
+     type = "toarray", sort = TRUE)
```

```
, , F

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

, , G

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

String relations in D_3

Function `strings()` allows finding *word tables* in the group structure

```
strings(D3)
```

```
$wt
, , F          , , FF          , , GF

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

, , G          , , FG          , , GG

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

Equations in group structure, D_3 $(k = 3)$

With argument `equat`, we can find the group equations with the identity

```r
strings(D3, equat = TRUE, k = 3)
```

```
$equat
$equat$F
[1] "F"   "GGF" "FGG"

$equat$G
[1] "G"   "GGG" "FGF"

$equat$FF
[1] "FF"  "GFG"

$equat$FG
[1] "FG"  "GFF"

$equat$GF
[1] "GF"  "FFG"

$equat$GG
[1] "GG"  "FFF"


$equate
$equate$e
[1] "e"   "GG"  "FFF"
```

Group structure, D_3

Function `semigroup()` allows finding the group structure

➟ since "any group is a semigroup as well"

```
D3S <- semigroup(D3)
```

```
...

$st
[1] "F"  "G"  "FF" "FG" "GF" "GG"

$S
  1 2 3 4 5 6
1 3 4 6 5 2 1
2 5 6 4 3 1 2
3 6 5 1 2 4 3
4 2 1 5 6 3 4
5 4 3 2 1 6 5
6 1 2 3 4 5 6

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

Group structure, D_3

symbolic format

```
# semigroup structure with symbolic format
semigroup(D3, type = "symbolic")$S
```

```
    F  G FF FG GF GG
F  FF FG GG GF  G  F
G  GF GG FG FF  F  G
FF GG GF  F  G FG FF
FG  G  F GF GG FF FG
GF FG FF  G  F GG GF
GG  F  G FF FG GF GG
```

Permutation of the group structure, D_3

`perm()` for rearrangement of elements' group structure in `D3S`

```
D3S <- perm(D3S$S, clu = c(2,4,3,5,6,1))
```

```
  6 1 3 2 4 5
6 6 1 3 2 4 5
1 1 3 6 4 5 2
3 3 6 1 5 2 4
2 2 5 4 6 3 1
4 4 2 5 1 6 3
5 5 4 2 3 1 6
```

This comes from the string labels where `GG` is the identity element

```
..
$st
[1] "F"  "G"  "FF" "FG" "GF" "GG"
...
```

Depiction of group structure: Cayley graph

Definition (Cayley graph)

The *Cayley graph* Γ of a group G with respect to a generating set $C \subseteq G$:

$$\Gamma = \Gamma(G, C).$$

- G is the node set in Γ
- A generator $c \in C$ connects two nodes $a, b \in G$ whenever $b = ca$

  ➟ i.e. all pairs of the form $(a, c \cdot b)$ make the edge set in Γ

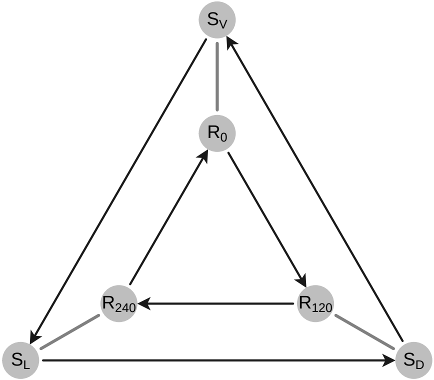
Cayley colour graph

Example (Cayley graph, integers under addition $\mathbb{Z}_2$)

```
  e x
e e x
x x e

e=ee  => solid loop
e=xx  => solid loop

x=ex  => dashed arc
x=xe  => dashed arc
```

Dihedral group, D_3

Cayley graph



Depiction of the group structure, D_3

Cayley table

Relabeling of elements in group structure with `as.semigroup()`

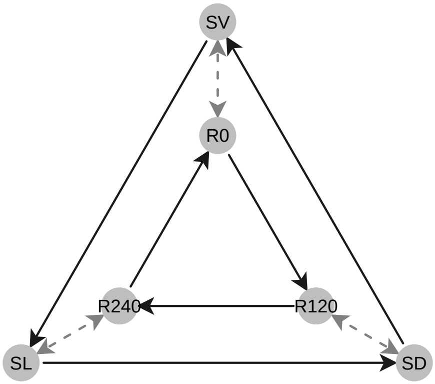
```
D3S <- as.semigroup(D3S, gens = c(2, 4),
+     lbs = c("R0", "R120", "R240", "SV", "SD", "SL"))
```

```
...
$st
[1] "R0"   "R120" "R240" "SV"   "SD"   "SL"

$gens
[1] "R120" "SV"

$S
       R0 R120 R240   SV   SD   SL
R0     R0 R120 R240   SV   SD   SL
R120 R120 R240   R0   SD   SL   SV
R240 R240   R0 R120   SL   SV   SD
SV     SV   SL   SD   R0 R240 R120
SD     SD   SV   SL R120   R0 R240
SL     SL   SD   SV R240 R120   R0

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

Depiction of the group structure, D_3

Cayley graph

```
# plot Cayley colour graph with a 2-radii concentric layout
scpD3 <- list(cex = 7, lwd = 3, pos = 0, col = 8, fsize = 16)
ccgraph(D3S, conc = TRUE, nr = 2, scope = scpD3)
```



3. Group structure in social networks

Example 3: Kariera kinship

Kariera society kinship system and group structure

- Despite the symmetry, algebraic groups can model human societies

- Some primitive societies like the *Kariera* from Western Australia have kinship networks that follow the rules of a group structure

  ➟ where *primitive* means "first of its class"

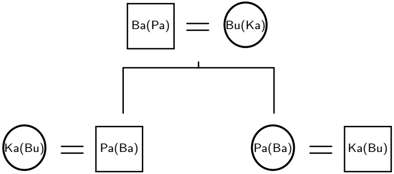
- The Karieras have (had?) four clans with specific rules of marriage & descent: *Banaka*, *Burung*, *Karimera*, and *Palyeri*.

  ➟ data collected by Radcliffe-Brown, analysed by White (1963)

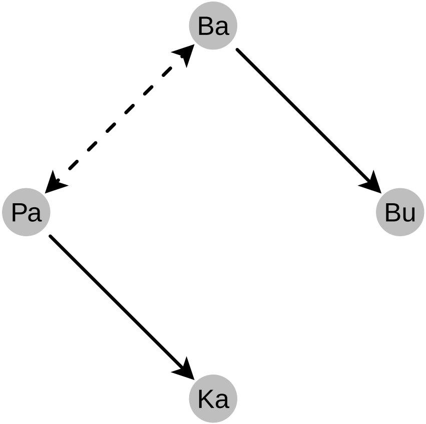
Kariera rules for marriage & descent (I)

Clans: Banaka (Ba), Burung (Bu), Karimera (Ka), Palyeri (Pa)

Two types of descent rules among Banaka and Palyeri (ego male)



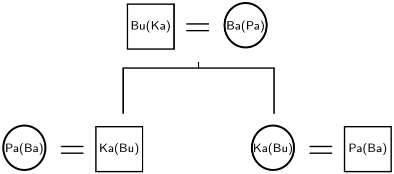
square: male, circle: female



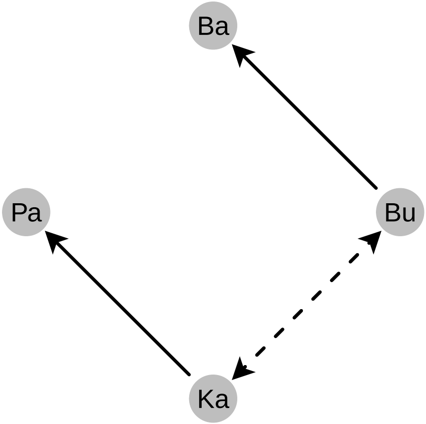
Kariera rules for marriage & descent (II)

Clans: Banaka (Ba), Burung (Bu), Karimera (Ka), Palyeri (Pa)

Two types of descent rules among Burung and Karimera (ego male)



square: male, circle: female



Parallel-cousins marriages in kinship networks

identifiers, F for male and G for female, are with right multiplication

$\mathsf{FG} = \mathsf{GG}$ $\qquad$ $\mathsf{GF} = \mathsf{FF}$

(a) Matrilineal $\qquad$ (b) Patrilineal

Cross-cousins marriages in kinship networks

identifiers, F for male and G for female, are with right multiplication

FG = GF   FF = GG

(a) Matrilineal   (b) Patrilineal

Permutation matrices for marriage & descent

Kariera kinship system

```
# create permutation matrices for marriage & descent rules
kks <- transf(list(F = c("1, 2", "3, 4", "2, 1", "4, 3"),
+     G = c("1, 4", "2, 3", "3, 2", "4, 1")))
```

```
, , F

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

, , G

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

Group structure as multiplication table

Kariera kinship system

The *multiplication table* reflects the group structure of the clan system

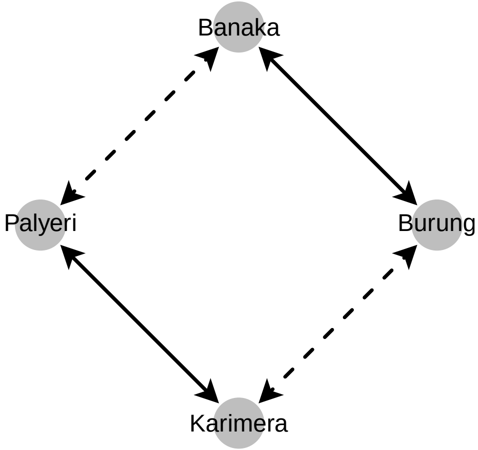
```
# Group structure with a symbolic format
semigroup(kks, type = "symbolic")
```

```
$dim
[1] 4
...

$ord
[1] 4

$st
[1] "F"  "G"  "FF" "FG"

$S
    F  G FF FG
F  FF FG  F  G
G  FG FF  G  F
FF  F  G FF FG
FG  G  F FG FF

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

Rules of marriage & descent

Kariera kinship system

```
# visualize marriage & descent rules in the Kariera
multigraph(kks, scope = scpD3, ecol = 1, collRecip = TRUE,
+     lbs = c("Banaka", "Burung", "Karimera", "Palyeri"))
```



Set of equations

to identify cross- and parallel-cousins marriages

The *set of equations* to detect allowed marriage types by commutation

```
# the equations allows finding marriage types in 'kks'
strings(kks, equat = TRUE)
```

```
...
$st
[1] "F"  "G"  "FF" "FG"

$equat
$equat$FF
[1] "FF" "GG"

$equat$FG
[1] "FG" "GF"


$equate
$equate$e
[1] "e"  "FF" "GG"
```

☛ *Both cross-cousins marriages are permitted in the Kariera*

Algebraic constraints in group structures

Two *algebraic constraints* for the analysis of the elementary structures:

- *Multiplication table* with relations between the different types of tie
- *Set of equations* among different types of tie

☛ *Complex structures have additional algebraic constraints*

$4a$. Multiplex networks

Example 4: Florentine families

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

Semigroup

- The algebraic structure of *semigroup* is made of a set of elements with an attached associative operation

$$\langle\, S,\ \circ \,\rangle$$

  - S as underlying set, closed under the operation
  - $\circ$ as the binary operation on an ordered pair, i.e. $\circ\colon S \times S \to S$ that, for all $x, y, z \in S$ satisfies the associative law:

$$x \circ (y \circ z) \;=\; (x \circ y) \circ z$$

- In a *semigroup of relations* $S(R)$ that represents the relational structure, x and y are *generators*, and $x \circ y$ a *compound*

➟ elements in $S(R)$ are unique representative *strings* of these

Hierarchy of relations: Partial order structure

Complex structures with lack of symmetry

- A *partially ordered semigroup* is S with a partial order
- A *partial order* is defined by an *inclusion* relation $\leq$ among $x, y \in S$ with the rule:

$$S^{\leq}_{x,y} = \begin{cases} 1 & \text{iff relation } x \text{ is contained in relation } y \\ 0 & \text{otherwise} \end{cases}$$

where "contained" implies that all ties in x are also occurring in y

Issues with the semigroup structure

- Modelling a multiple network by $S(R)$ typically results in a quite large structure, even if the system is small

- An important task is to reduce complexity of the network

  ➟ this is done by grouping different classes of actors

- *Blockmodeling* is an effective way to reduce the network and keeping the essential structure of the system

➥ *Positional Analysis*

☛ But it needs to preserve the network multiplicity of ties

Equivalence types in positional analysis

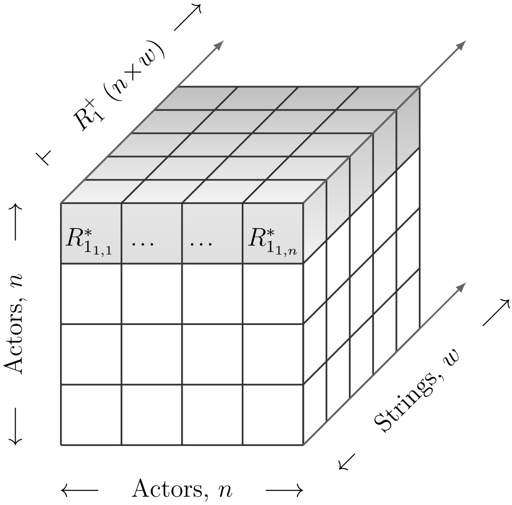
Each *type* of graph homomorphism (a structure-preserving mapping) induces to a particular kind of equivalence

➟ which represents a system of *positions* and *roles* of the network

- Equivalences from a *global perspective*:
  - Structural (Lorrain & White, 1971)
  - Automorphic (Winship & Mandel, 1983; Everett, 1985)
  - Regular (Sailer [Boyd], 1978; White & Reitz, 1983)
  - Generalized (Batagelj et al, 1992; Doreian et al, 1994)

- Equivalences from a *local perspective*:
  - Local Role (Winship & Mandel, 1983; Mandel, 1983)
  - *Compositional* (Breiger & Pattison, 1986; Mandel, 1978)

Compositional equivalence: Relation-Box



Compositional equivalence: Person hierarchies

Person Hierarchies

- Builds on the ordering among the actors' Role Relations in a particular *relation plane* (shadow part in the Relation-Box)

- All perceived inclusions in R_l^+ represents the *person hierarchy* H_l defined for $l, i, j \in X$ and relation x as:

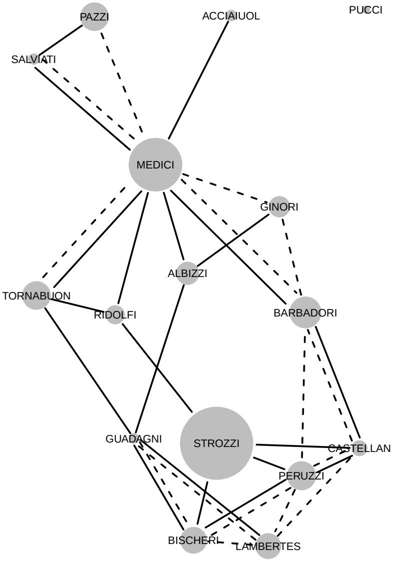
$$H_{l_{ij}} = \begin{cases} 1 & \text{iff } \; R^*_{l_{xi}} \leq R^*_{l_{xj}} \\ 1 & \text{iff } \; R^*_{l_{xi}} = R^*_{l_{xj}} \\ 0 & \text{iff } \; R^*_{l_{xi}} \nleq R^*_{l_{xj}} \\ 0 & \text{iff } \; \sum R^*_{l_{xi}} = 0 \end{cases}$$

- A *cumulated person hierarchy* matrix is based on the union of all person hierarchies with transitive closure

  ➟ the establishment of roles and positions are from the perspectives of individual actors, but it also considers common relational features

Compositional equivalence: Florentine families

Undirected network, solid: marriage, dashed: business

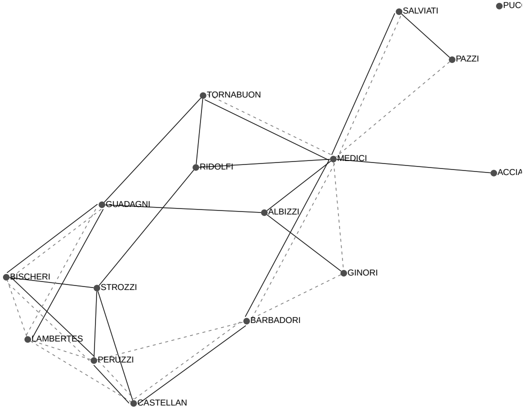


Florentine families network

(Padgett & Ansell, 1993)

```
# Florentine families data set as a Ucinet DL file
flf <- read.dl(file = "http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/padgett.dat")
```

```
multigraph(flf, directed = FALSE, layout = "force", seed = 1)
```



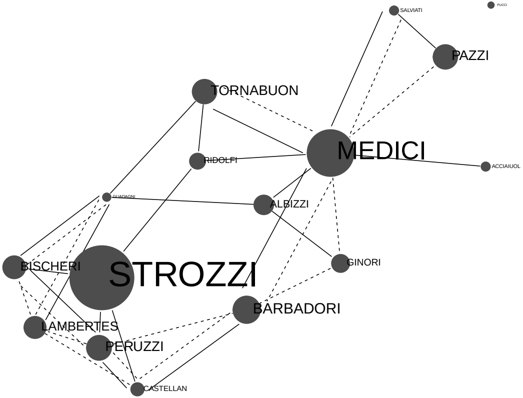
Florentine families network

```r
# Actor attributes
flfa <- read.dl(file = "http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/padgw.dat")
flfa <- flfa[order(rownames(flfa)), ]
```

```
          WEALTH #PRIORS #TIES
ACCIAIUOL     10      53     2
ALBIZZI       36      65     3
BARBADORI     55       0    14
BISCHERI      44      12     9
CASTELLAN     20      22    18
GINORI        32       0     9
GUADAGNI       8      21    14
LAMBERTES     42       0    14
MEDICI       103      53    54
PAZZI         48       0     7
PERUZZI       49      42    32
PUCCI          3       0     1
RIDOLFI       27      38     4
SALVIATI      10      35     5
STROZZI      146      74    29
TORNABUON     48       0     7
```

Florentine families network

```
# plotting with actor attributes
multigraph(flf, directed = FALSE, layout = "force", seed = 1, ecol = 1,
           cex = flfa[,1])
```



Florentine families network

```r
# inspect the network relational system
rel.sys(flf, bonds = "full")$incl
```

```
[1] "BARBADORI" "BISCHERI"  "CASTELLAN" "GUADAGNI"  "LAMBERTES" "MEDICI"    "PERUZZI"
[8] "SALVIATI"  "TORNABUON"
```

```r
# who is not linked at both levels
rel.sys(flf, bonds = "full")$excl
```

```
[1] "ACCIAIUOL" "ALBIZZI"   "GINORI"    "PAZZI"     "PUCCI"     "RIDOLFI"   "STROZZI"
```

Compositional equivalence: Relation-Box

Florentine families

```
# function to construct the Relation-Box
formals("rbox")
```

```
$w


$transp
[1] FALSE

$smpl
[1] FALSE

$k
[1] 3

$tlbs
```

Compositional equivalence: Cumulated person hierarchy

Florentine families

`cph()` serves to construct the network cumulated person hierarchy

```
# input must be a "Rel.Box" class object
cph(rbox(flf))
```

```
           ACCIAIUOL ALBIZZI BARBADORI BISCHERI CASTELLAN GINORI GUADAGNI LAMBERTES MEDICI PAZZI
ACCIAIUOL          1       1         1        1         1      1        1         1      1     1
ALBIZZI            1       1         1        1         1      1        1         1      1     1
BARBADORI          1       1         1        1         1      1        1         1      1     1
BISCHERI           1       1         1        1         1      1        1         1      1     1
CASTELLAN          1       1         1        1         1      1        1         1      1     1
GINORI             1       1         1        1         1      1        1         1      1     1
GUADAGNI           1       1         1        1         1      1        1         1      1     1
LAMBERTES          1       1         1        1         1      1        1         1      1     1
MEDICI             1       1         1        1         1      1        1         1      1     1
PAZZI              1       1         1        1         1      1        1         1      1     1
PERUZZI            1       1         1        1         1      1        1         1      1     1
PUCCI              0       0         0        0         0      0        0         0      0     0
RIDOLFI            1       1         1        1         1      1        1         1      1     1
SALVIATI           1       1         1        1         1      1        1         1      1     1
STROZZI            1       1         1        1         1      1        1         1      1     1
TORNABUON          1       1         1        1         1      1        1         1      1     1
attr(,"class")
[1] "Partial.Order" "CPH"
```

(Extract)

Compositional equivalence: Cumulated person hierarchy

Florentine families

```r
cph(rbox(flf, k = 4))
```

```
          ACCIAIUOL ALBIZZI BARBADORI BISCHERI CASTELLAN GINORI GUADAGNI LAMBERTES MEDICI PAZZI
ACCIAIUOL         1       1         1        1         1      1        1         1      1     1
ALBIZZI           1       1         1        1         1      1        1         1      1     1
BARBADORI         1       1         1        1         1      1        1         1      1     1
BISCHERI          1       1         1        1         1      1        1         1      1     1
CASTELLAN         1       1         1        1         1      1        1         1      1     1
GINORI            1       1         1        1         1      1        1         1      1     1
GUADAGNI          1       1         1        1         1      1        1         1      1     1
LAMBERTES         1       1         1        1         1      1        1         1      1     1
MEDICI            1       1         1        1         1      1        1         1      1     1
PAZZI             1       1         1        1         1      1        1         1      1     1
PERUZZI           1       1         1        1         1      1        1         1      1     1
PUCCI             0       0         0        0         0      0        0         0      0     0
RIDOLFI           1       1         1        1         1      1        1         1      1     1
SALVIATI          1       1         1        1         1      1        1         1      1     1
STROZZI           1       1         1        1         1      1        1         1      1     1
TORNABUON         1       1         1        1         1      1        1         1      1     1
attr(,"class")
[1] "Partial.Order" "CPH"
```

(Extract)

Compositional equivalence: Cumulated person hierarchy

Florentine families

```
# test objects for exact equality
identical(cph(rbox(flf, k = 3)), cph(rbox(flf, k = 4)))
```

```
[1] TRUE
```

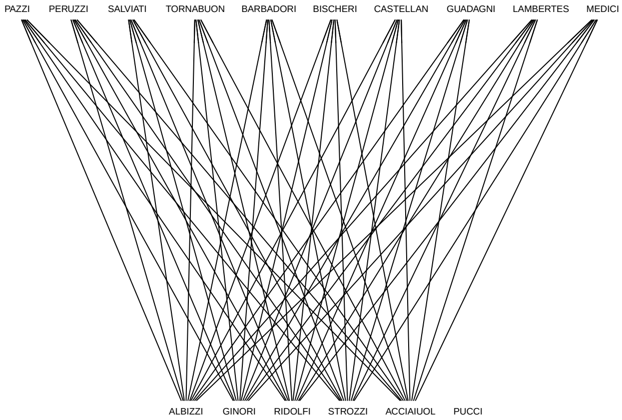
```
identical(cph(rbox(flf, k = 4)), cph(rbox(flf, k = 5)))
```

```
[1] FALSE
```

Visualization of the poset

Florentine families

```r
# CPH is a poset
diagram(cph(rbox(flf, k = 5)))
```



Compositional equivalence: Positional analysis

Florentine families

```
# levels in the ploted hasse diagram
diagram.levels(cph(rbox(flf, k = 5)))
```

```
         2        2          1         1          1       2         1          1       1      1
1 ACCIAIUOL ALBIZZI BARBADORI BISCHERI CASTELLAN GINORI GUADAGNI LAMBERTES MEDICI PAZZI
        1     3       2         1        2          1
1 PERUZZI PUCCI RIDOLFI SALVIATI STROZZI TORNABUON
```

```
# obtain the clustering with 'perm' argument
diagram.levels(cph(rbox(flf, k = 5)), perm = TRUE)$clu
```

```
[1] 2 2 1 1 1 2 1 1 1 1 1 3 2 1 2 1
```

☛ However, levels in the plotted Hasse diagram are not always the best criteria for classifying the actors

Compositional equivalence: Positional analysis

Florentine families

```r
# first record the clustering vector
flfclu <- diagram.levels(cph(rbox(flf, k = 5)), perm = TRUE)$clu

# apply clustering to produce a positional system with function reduc()
flfps <- reduc(flf, clu = flfclu)
```

```
, , PADGM

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

, , PADGB

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

Compositional equivalence: Role structure

Florentine families

```r
# the semigroup of the positional system in default format
semigroup(flfps)
```

```
$dim
[1] 3

$gens

...

$ord
[1] 2

$st
[1] "PADGM" "PADGB"

$S
  1 2
1 1 1
2 1 1

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

Compositional equivalence with actor attributes

For a given attribute defined in α, and for $i = x_1, x_2, \ldots, x_n$, attribute information is analyzed in relational terms where pair of vectors are element of an indexed matrix A^α as:

$$a_{ij}^{\alpha} =; \delta_{ij},$$

Here

$$c_i = \begin{cases} 1 & \text{if the corresponding attribute is tied to actor } i \\ 0 & \text{otherwise.} \end{cases}$$

And δ_{ij} is defined for nodes $i, j = x_1, x_2, \ldots, x_n$ in X by the Kronecker delta function as:

$$\delta_{ij} = \begin{cases} 1 & \text{for } \; i = j \\ 0 & \text{for } \; i \neq j. \end{cases}$$

☛ That is, A^α is a *diagonal matrix*.

Actor attributes in relational structures

Florentine families

```
          WEALTH #PRIORS #TIES
ACCIAIUOL     10      53     2
ALBIZZI       36      65     3
BARBADORI     55       0    14
BISCHERI      44      12     9
CASTELLAN     20      22    18
GINORI        32       0     9
GUADAGNI       8      21    14
LAMBERTES     42       0    14
MEDICI       103      53    54
PAZZI         48       0     7
PERUZZI       49      42    32
PUCCI          3       0     1
RIDOLFI       27      38     4
SALVIATI      10      35     5
STROZZI      146      74    29
TORNABUON     48       0     7
```

```r
# read.srt() transforms data frames into arrays
read.srt(flfa, attr = TRUE, rownames = TRUE)

# split rich from very rich actors and bind arrays
fw <- dichot(read.srt(flfa, attr = TRUE, rownames = TRUE)[,,1], c = 40)
flfw <- zbind(flf, fw)
```

Compositional equivalence: CPH with actor attributes

Florentine families

```r
# test objects for exact equality
identical(cph(rbox(flfw, k = 2)), cph(rbox(flfw, k = 3)))
```

```
[1] TRUE
```

```r
identical(cph(rbox(flfw, k = 3)), cph(rbox(flfw, k = 4)))
```

```
[1] TRUE
```

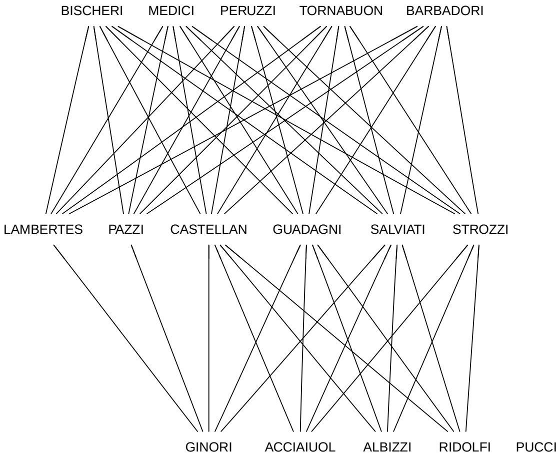
```r
identical(cph(rbox(flfw, k = 4)), cph(rbox(flfw, k = 5)))
```

```
[1] FALSE
```

Hasse Diagram of CPH with actor attributes

Florentine families

```r
diagram(cph(rbox(flfw, k = 5)))
```



Positional analysis with actor attributes

Florentine families

```r
# positional system with the clustering info of the hasse diagram
flfwclu <- diagram.levels(cph(rbox(flfw, k = 5)), perm = TRUE)$clu
flfwps <- reduc(flfw, clu = flfwclu)
```

```
, , PADGM

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

, , PADGB

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

, , 3

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

Algebraic constraint: Role table

Florentine families role structure with actor attributes

```r
# semigroup of role relations with costumized labels
semigroup(flfwps, type = "symbolic", lbs = c("M", "B", "W"))$S

# or even better...
dimnames(flfwps)[3][[1]] <- c("M", "B", "W")
semigroup(flfwps, type = "symbolic")$S
```

```
     M  B   W  MW  BW WM WB WMW
M    M  M  MW  MW  MW  M  M  MW
B    M  M  BW  MW  MW  M  M  MW
W   WM WB   W WMW WMW WM WB WMW
MW   M  M  MW  MW  MW  M  M  MW
BW   M  M  BW  MW  MW  M  M  MW
WM  WM WM WMW WMW WMW WM WM WMW
WB  WM WM WMW WMW WMW WM WM WMW
WMW WM WM WMW WMW WMW WM WM WMW
```

Algebraic constraint: Set of equations

Florentine families role structure with actor attributes

```
# function strings() serves to find equations among relations
flfwst <- strings(flfwps, equat = TRUE, k = 3)$equat
```

```
$M
 [1] "M"   "MM"  "BB"  "MB"  "BM"  "MMM" "BBM" "MBB" "MMB" "BBB" "BMM" "BMB" "MBM" "MWM"
[15] "BWB" "MWB" "BWM"

$W
[1] "W"   "WW"  "WWW"

$MW
[1] "MW"  "MWW" "MMW" "BBW" "MBW" "BMW"

$BW
[1] "BW"  "BWW"

$WM
[1] "WM"  "WWM" "WMM" "WBB" "WMB" "WBM"

$WB
[1] "WB"  "WWB"

$WMW
[1] "WMW" "WBW"
```

Algebraic constraint: Partial ordering

Florentine families role structure with actor attributes

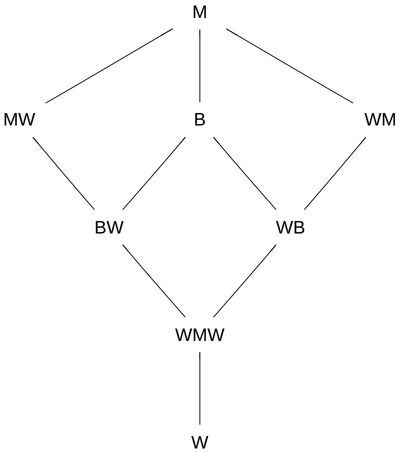
```r
# partial ordering of string relations
partial.order(flfwst, type = "strings")
```

```
    M B W MW BW WM WB WMW
M   1 0 0  0  0  0  0   0
B   1 1 0  0  0  0  0   0
W   1 1 1  1  1  1  1   1
MW  1 0 0  1  0  0  0   0
BW  1 1 0  1  1  0  0   0
WM  1 0 0  0  0  1  0   0
WB  1 1 0  0  0  1  1   0
WMW 1 1 0  1  1  1  1   1
attr(,"class")
[1] "Partial.Order" "strings"
```

Hasse diagram for the partial order of string relations

Florentine families role structure with actor attributes

```r
diagram(partial.order(flfwst, type = "strings"))
```



More on modeling complex networks with a partially ordered semigroup

Journal of Statistical Software

February 2020, Volume 92, Issue 11. *doi: 10.18637/jss.v092.i11*

Algebraic Analysis of Multiple Social Networks with multiplex

J. Antonio Rivero Ostoic
Aarhus University

Abstract

multiplex is a computer program that provides algebraic tools for the analysis of multiple network structures within the R environment. Apart from the possibility to create and manipulate multivariate data representing multiplex, signed, and two-mode networks, this package offers a collection of functions that deal with algebraic systems – such as the partially ordered semigroup, and balance or cluster semirings – their decomposition, and the enumeration of bundle patterns occurring at different levels of the network. Moreover, through Galois derivations between families of the pairs of subsets in different domains it is possible to analyze affiliation networks with an algebraic approach. Visualization of multigraphs, different forms of bipartite graphs, inclusion lattices, Cayley graphs is supported as well with related packages.

Keywords: social network analysis, relational algebra, graph visualization, R.

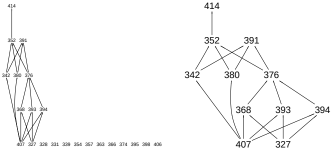


Figure 3: CPH of Incubator C with and without incomparable elements in the poset.

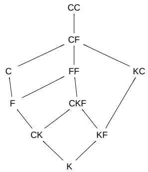


Figure 5: Hierarchy of string relations in the role structure of `netC`.

$4b.$ Signed networks

Example 5: Incubator network A

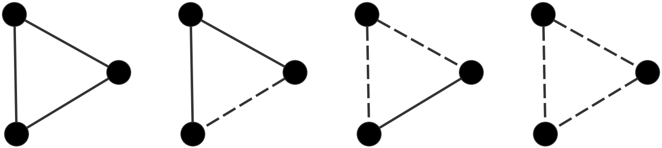
Structural Balance

- Simmel (1950) studied "conflict as a mechanism for integration" in triadic relations
- Heider (1958) developed the *Structural Balance* theory as a special cases of transitivity
- Structural Balance theory applies to networks to see whether the system has an inherent equilibrium or not

  "all positive ties within groups; all negative ties between groups"

Structural Balance

- A balanced structure is represented by a *signed network*, which is a special case of multiplex network



- Paths in signed graphs are positive when they have an even number of negative edges; otherwise negative

☛ *extension*: a path/semipath is ambivalent iff contains at least one ambivalent edge

Structures in Balance theory

balanced $\rightarrow$ **clusterable** $\rightarrow$ '*weak*' clusterable

(Cartwright & Harary, 1956) (Davis, 1967)

$\circ$	p	n
p	p	n
n	n	p

Classical

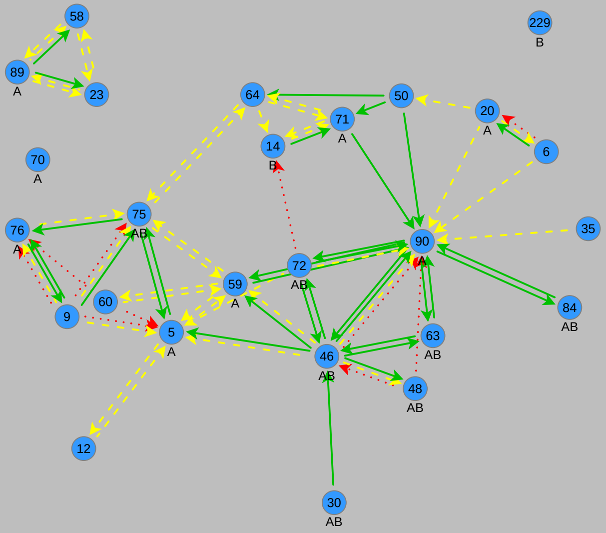
$\circ$	p	n	a
p	p	n	a
n	n	a	a
a	a	a	a

Extended

p $\rightarrow$ positive n $\rightarrow$ negative a $\rightarrow$ ambivalent

Incubator network "A"

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



Incubator network A

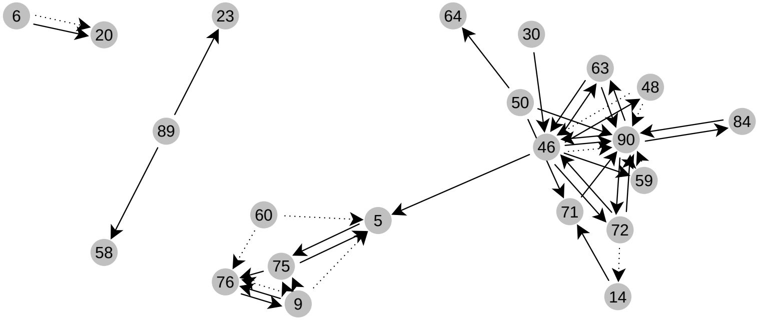
```r
# incubator network A dataset and structure
data("incA")
str(incA)
```

```
List of 5
 $ net  : num [1:26, 1:26, 1:5] 0 0 0 0 0 0 0 0 0 1 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:26] "5" "6" "9" "12" ...
  .. ..$ : chr [1:26] "5" "6" "9" "12" ...
  .. ..$ : chr [1:5] "C" "F" "K" "A" ...
 $ atnet:List of 1
  ..$ : num [1:5] 0 0 0 1 1
 $ IM   : num [1:4, 1:4, 1:7] 1 1 1 0 0 1 0 0 1 0 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : NULL
  .. ..$ : NULL
  .. ..$ : chr [1:7] "C" "F" "K" "D" ...
 $ atIM : num [1:7] 0 0 0 0 0 0 1
 $ ...
```

```r
# cooperation and competition ties in 'incA' without isolated actors
netA <- rm.isol(incA$net[,,c(1,3)])
```

Signed structure in Incubator network A

```r
# plot signed multigraph
scpA <- list(ecol = 1, vcol = "#C0C0C0", cex = 3, fsize = 8, pos = 0, bwd = .5)
multigraph(netA, scope = scpA, signed = TRUE, layout = "force", seed = 9)
```



Signed Network *C and K in Incubator A*

```
# signed() creates a "Signed" class object from 2 matrices
netAsg <- signed(netA)
```

```
$val
[1] p o n a

$s
   5 6 9 14 20 23 30 46 48 50 58 59 60 63 64 71 72 75 76 84 89 90
5  o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  p  o  o  o  o
6  o o o o  a  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
9  n o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  a  a  o  o  o
14 o o o o  o  o  o  o  o  o  o  o  o  o  o  p  o  o  o  o  o  o
20 o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
23 o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
30 o o o o  o  o  o  p  o  o  o  o  o  o  o  o  o  o  o  o  o  o
46 p o o o  o  o  o  o  p  o  o  p  o  p  o  o  p  o  o  o  o  a
48 o o o o  o  o  o  n  o  o  o  o  o  o  o  o  o  o  o  o  o  n
50 o o o o  o  o  o  o  o  o  o  o  o  o  p  p  o  o  o  o  o  p
58 o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
59 o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  p
60 n o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  n  o  o  o
63 o o o o  o  o  o  p  o  o  o  o  o  o  o  o  o  o  o  o  o  p
64 o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
71 o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  p
72 o o o n  o  o  o  p  o  o  o  o  o  o  o  o  o  o  o  o  o  p
75 p o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  p  o  o  o
76 o o p o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o
84 o o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  p
89 o o o o  o  p  o  o  o  o  p  o  o  o  o  o  o  o  o  o  o  o
90 o o o o  o  o  o  p  o  o  o  p  o  p  o  o  p  o  o  p  o  o

attr(,"class")
[1] "Signed"
```

Semiring

Algebraic structure

A *semiring* is an object set endowed with a pair operations, multiplication and addition, together with two neutral elements:

$$\langle\, Q,\ +,\ \cdot\,,\ 0,\ 1\,\rangle$$

properties:

- closed, associative, and commutative under addition
- multiplication distributes over addition, i.e. for all $\mathsf{p}, \mathsf{n}, \mathsf{a} \in Q$:

$$\mathsf{p}\ \cdot\ (\mathsf{n} + \mathsf{a}) = (\mathsf{p}\ \cdot\ \mathsf{n}) + (\mathsf{p}\ \cdot\ \mathsf{a}) \quad \text{and} \quad (\mathsf{p} + \mathsf{n})\ \cdot\ \mathsf{a} = (\mathsf{p}\ \cdot\ \mathsf{a}) + (\mathsf{n}\ \cdot\ \mathsf{a})$$

☛ *Semirings help us to evaluate the relational system in terms of balance theory by looking at paths and semipaths*

Semiring operations

$\cdot$	o	n	p	a
o	o	o	o	o
n	o	p	n	a
p	o	n	p	a
a	o	a	a	a

$+$	o	n	p	a
o	o	n	p	a
n	n	n	a	a
p	p	a	p	a
a	a	a	a	a

Balance

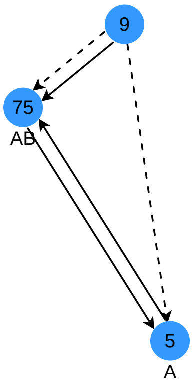
$\cdot$	o	n	p	a	q
o	o	o	o	o	o
n	o	q	n	n	q
p	o	n	p	a	q
a	o	n	a	a	q
q	o	q	q	q	q

$+$	o	n	p	a	q
o	o	n	p	a	q
n	n	n	a	a	n
p	p	a	p	a	p
a	a	a	a	a	a
q	q	n	p	a	q

Clustering

Balance semiring *(Signed triad)*

```
# 2-Paths  (9, 75)
"n, p" "o, a" "a, o"

# multiplication
"n" "o" "o"

# addition
n
```



	5	9	75
5	o	o	p
9	n	o	a
75	p	o	o

t^α

	5	9	75
5	p	o	o
9	a	o	n
75	o	o	p

t^α paths, $k > 1$

	5	9	75
5	p	a	a
9	a	a	n
75	a	n	a

t^α semipaths, $k = 2$

	5	9	75
5	a	a	a
9	a	a	a
75	a	a	a

t^α semipaths, $k > 2$

Semiring function

```r
# arguments in function semiring()
formals("semiring")
```

```
$x


$type
c("balance", "cluster")

$symclos
[1] TRUE

$transclos
[1] TRUE

$k
[1] 2

$lbs
```

Semiring structures

```
# balance semiring 2-paths (deafult)
semiring(netAsg, type = "balance")

# 3-paths
semiring(netAsg, type = "balance", k = 3)

# 2-semipaths
semiring(netAsg, type = "balance", symclos = FALSE)
# ...
```

```
# cluster semiring 2-paths (deafult)
semiring(netAsg, type = "cluster")

# 3-paths
semiring(netAsg, type = "cluster", k = 3)

# 2-semipaths
semiring(netAsg, type = "cluster", symclos = FALSE)
# ...
```

Checking for equilibrium in Balance semiring

```
identical(
+     semiring(netAsg, type = "balance", k = 3)$Q,
+     semiring(netAsg, type = "balance", k = 2)$Q )
```

```
[1] FALSE
```

```
identical(
+     semiring(netAsg, type = "balance", k = 3)$Q,
+     semiring(netAsg, type = "balance", k = 4)$Q )
```

```
[1] FALSE
```

```
identical(
+     semiring(netAsg, type = "balance", k = 4)$Q,
+     semiring(netAsg, type = "balance", k = 5)$Q )
```

```
[1] TRUE
```

Checking for equilibrium in Cluster semiring

```
identical(
+     semiring(netAsg, type = "cluster", k = 3)$Q,
+     semiring(netAsg, type = "cluster", k = 2)$Q )
```

```
[1] FALSE
```

```
identical(
+     semiring(netAsg, type = "cluster", k = 3)$Q,
+     semiring(netAsg, type = "cluster", k = 4)$Q )
```

```
[1] FALSE
```

```
identical(
+     semiring(netAsg, type = "cluster", k = 4)$Q,
+     semiring(netAsg, type = "cluster", k = 5)$Q )
```

```
[1] TRUE
```

Weak balance structure with semipaths

```r
# balance with length four
QnetA <- semiring(netAsg, type = "balance", k = 4)

perm(QnetA$Q, clu = c(1,6,1,2,6,6,3,2,2,3,6,2,4,2,5,2,2,1,1,2,6,2))
```

```
   5 9 75 76 14 46 48 59 63 71 72 84 90 30 50 60 64 6 20 23 58 89
5  a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
9  a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
75 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
76 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
14 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
46 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
48 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
59 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
63 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
71 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
72 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
84 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
90 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
30 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
50 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
60 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
64 a a a  a  a  a  a  a  a  a  a  a  a  a  a  a  a  o o  o  o  o
6  o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  a o  o  o  o
20 o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o a  o  o  o
23 o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  p  p  o
58 o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  p  p  o
89 o o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  p
```

Weak balance structure with paths

```r
# balance with length four
QnetA <- semiring(netAsg, type = "balance", symclos = FALSE, k = 4)

perm(QnetA$Q, clu = c(1,6,1,2,6,6,3,2,2,3,6,2,4,2,5,2,2,1,1,2,6,2))
```

```
   5 9 75 76 14 46 48 59 63 71 72 84 90 30 50 60 64 6 20 23 58 89
5  a a  a  a  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
9  a a  a  a  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
75 a a  a  a  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
76 a a  a  a  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
14 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
46 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
48 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
59 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
63 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
71 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
72 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
84 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
90 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
30 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
50 a a  a  a  a  a  a  a  a  a  a  a  a  o  o  o  o o  o  o  o  o
60 a a  a  a  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
64 o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
6  o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
20 o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
23 o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
58 o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
89 o o  o  o  o  o  o  o  o  o  o  o  o  o  o  o  o o  o  o  o  o
```

Main component of Incubator A

weak balance structure

```
# comps() finds components and isolates
comps(netA)
```

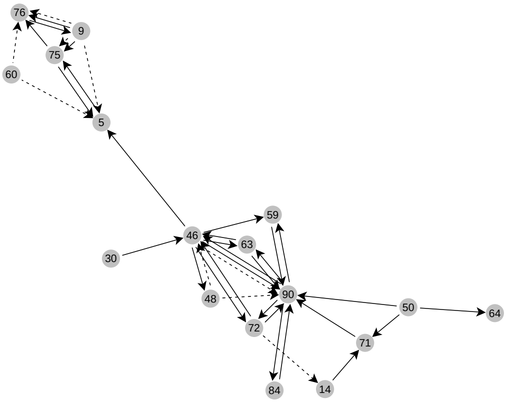
```
$com
$com[[1]]
 [1] "5"  "50" "59" "60" "63" "64" "71" "72" "75" "76" "84" "90" "9"  "14" "30" "46" "48"

$com[[2]]
[1] "58" "89" "23"

$com[[3]]
[1] "6"  "20"


$isol
character(0)
```

```
# extract the 17 ties from main component in 'netA'
com <- comps(netA)$com[[1]]
# select in component two types of tie and actor attributes
nsA <- rel.sys(incA$net[,,c(1,3,4:5)], type = "toarray", sel = com)
```

Main component of Incubator A

weak balance structure

```
# plot network relations 'C' and 'K' in main component
multigraph(nsA, layout = "force", seed = 123, scope = scpA)
```



Factions with weak balance structure

paths main component of Incubator A

```
# scope with factions
scpAc <- list(lty = c(1,3), clu = c(1,1,2,3,2,2,3,2,4,2,5,2,2,1,1,2,2),
+             vcol = c("blue","red","green","orange","peru"), alpha = .5)
```

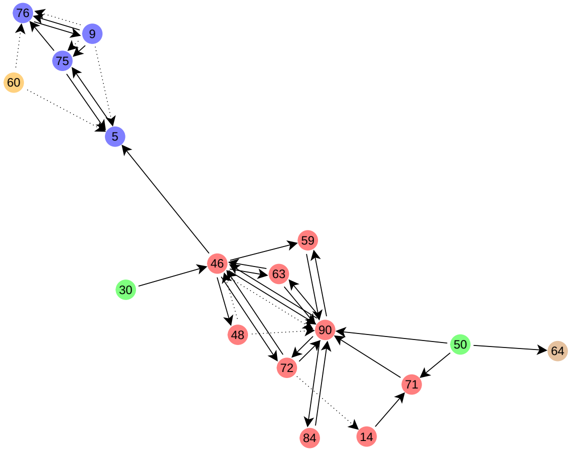
```
c(scpAc,scpA)
```

```
$lty
[1] 1 3

$clu
 [1] 1 1 2 3 2 2 3 2 4 2 5 2 2 1 1 2 2

$vcol
[1] "blue"   "red"    "green"  "orange" "peru"

$alpha
[1] 0.5

$ecol
[1] 1

$vcol
[1] "#C0C0C0"

...
```

Factions with weak balance structure

paths main component of Incubator A

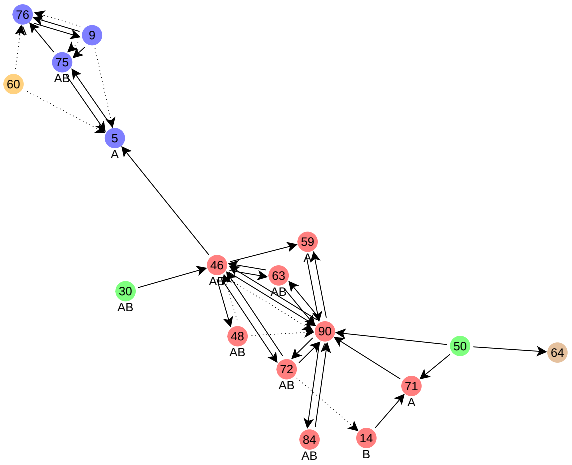
```
# plot combining scopes
multigraph(nsA, layout = "force", seed = 123, scope = c(scpA, scpAc))
```



Social influence through comparison

weak balance structure with paths

```
multigraph(nsA, att = nsA[,,3:4], layout = "force", seed = 123,
+     scope = c(scpA, scpAc))
```



$5a.$ Affiliation networks

Example 6: Group of Twenty

Affiliation networks

- Ties between two sets of entities represent two-mode, bipartite, or *affiliations networks*

  ➟ like the duality between *"people and groups"*, *"person and events"*, *"actors and their attributes"*

- In a 2-mode matrix data the domain and the codomain are not equal

  ➟ serves to represent affiliations networks

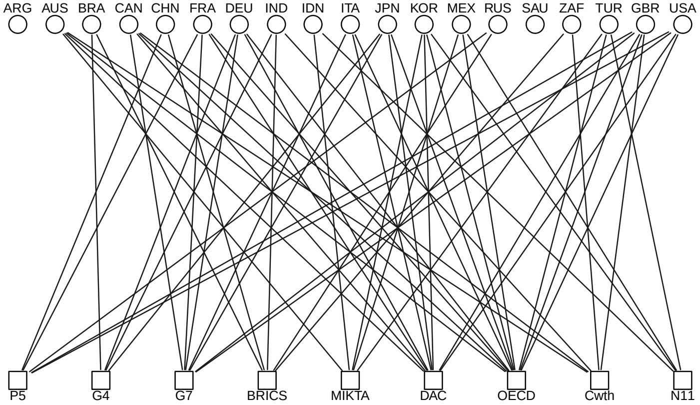
Group of Twenty (G20) affiliation network

```r
G20 <- data.frame(
         P5    = c(0,0,0,0,1,0,1,1,0,0,0,0,0,0,1,0,0,1,0),
         G4    = c(0,0,1,0,0,1,0,0,0,1,0,1,0,0,0,0,0,0,0),
         G7    = c(0,0,0,1,0,1,1,1,0,0,1,1,0,0,0,0,0,1,0),
         BRICS = c(0,0,1,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,1),
         MITKA = c(0,1,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0),
         DAC   = c(0,1,0,1,0,1,1,1,0,0,1,1,1,0,0,0,0,1,0),
         OECD  = c(0,1,0,1,0,1,1,1,0,0,1,1,1,1,0,0,1,1,0),
         Cwth  = c(0,1,0,1,0,0,0,1,0,1,0,0,0,0,0,0,0,0,1),
         N11   = c(0,0,0,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0) )
rownames(G20) <- c("ARG","AUS","BRA","CAN","CHN","DEU","FRA","GBR","IDN","IND",
                   "ITA","JPN","KOR","MEX","RUS","SAU","TUR","USA","ZAF")
```

```
    P5 G4 G7 BRICS MITKA DAC OECD Cwth N11
ARG  0  0  0     0     0   0    0    0   0
AUS  0  0  0     0     1   1    1    1   0
BRA  0  1  0     1     0   0    0    0   0
CAN  0  0  1     0     0   1    1    1   0
CHN  1  0  0     1     0   0    0    0   0
DEU  0  1  1     0     0   1    1    0   0
FRA  1  0  1     0     0   1    1    0   0
GBR  1  0  1     0     0   1    1    1   0
IDN  0  0  0     0     1   0    0    0   1
IND  0  1  0     1     0   0    0    1   0
ITA  0  0  1     0     0   1    1    0   0
JPN  0  1  1     0     0   1    1    0   0
KOR  0  0  0     0     1   1    1    0   1
MEX  0  0  0     0     1   0    1    0   1
RUS  1  0  0     1     0   0    0    0   0
SAU  0  0  0     0     0   0    0    0   0
TUR  0  0  0     0     1   0    1    0   1
USA  1  0  1     0     0   1    1    0   0
ZAF  0  0  0     1     0   0    0    1   0
```

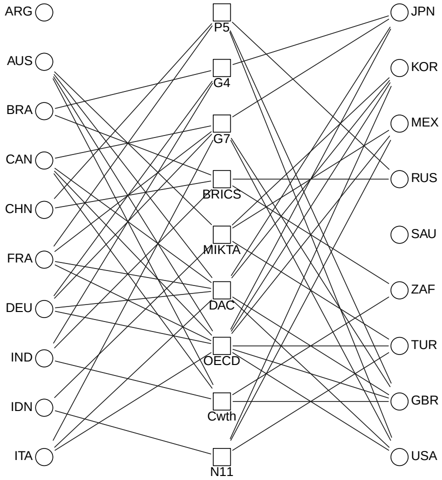
G20 Countries (affiliation network)

```r
# bipartite graph of 'G20'
bmgraph(G20, rot = 90, mirrorX = TRUE)
```



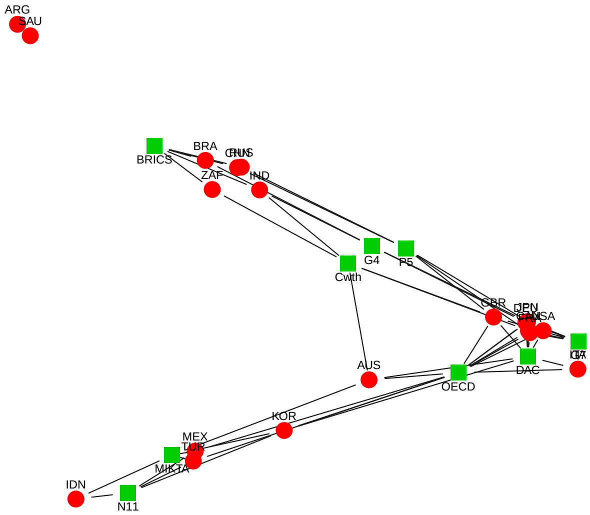
G20 Countries (affiliation network)

```
# bipartite graph with three columns
bmgraph(G20, layout = "bip3", cex = 3, fsize = 10)
```



G20 Countries (affiliation network)

```
# correspondence analysis plot
bmgraph(G20, layout = "CA", rot = 99, vcol = 2:3, pch = c(19, 15), jitter = .1)
```



Formal Concept Analysis (Ganter & Wille, 1996)

algebraic approach

- *Formal concept analysis* is an analytical framework for the study of affiliation networks

- Elements in the domain and codomain are called *objects* and *attributes* resp.

- The set of objects G and the set of attributes M are associated with an incident relation $I \subseteq G \times M$ in a *formal context*

- The *formal concept* of a formal context is a pair of sets of maximally contained objects A and attributes B

  ➟ (i.e. maximal rectangles in the formal context)

A and B are said to be the *extent* and *intent* of the formal concept

Galois Derivations

- A *Galois derivation* between G and M is defined for any subsets $A \subseteq G$ and $B \subseteq M$ by

$$
\begin{aligned}
A' &= m \in M \mid (g, m) \in I \quad \text{(for all } g \in A) \\
B' &= g \in G \mid (g, m) \in I \quad \text{(for all } m \in B)
\end{aligned}
$$

– A' is the set of attributes common to all the objects in the intent

– B' the set of objects possessing the attributes in the extent

```
formals("galois")
```

```
$x


$labeling
c("full", "reduced")
```

Galois derivations in G20

```r
galois(G20)
```

```
$P5
[1] "CHN, FRA, GBR, RUS, USA"

$G4
[1] "BRA, DEU, IND, JPN"

$`DAC, G7, OECD`
[1] "CAN, DEU, FRA, GBR, ITA, JPN, USA"

$BRICS
[1] "BRA, CHN, IND, RUS, ZAF"

$MIKTA
[1] "AUS, IDN, KOR, MEX, TUR"

$`DAC, OECD`
[1] "AUS, CAN, DEU, FRA, GBR, ITA, JPN, KOR, USA"

$OECD
[1] "AUS, CAN, DEU, FRA, GBR, ITA, JPN, KOR, MEX, TUR, USA"

$Cwth
[1] "AUS, CAN, GBR, IND, ZAF"

$`MIKTA, N11`
[1] "IDN, KOR, MEX, TUR"

$`BRICS, Cwth, DAC, G4, G7, MIKTA, N11, OECD, P5`
character(0)

...
```

Galois derivations in G20 – Reduced labeling

```r
g20gc <- galois(G20, labeling = "reduced")
```

```
$reduc
$reduc$P5
character(0)

$reduc$G4
character(0)

$reduc$G7
[1] "ITA"

$reduc$BRICS
character(0)

$reduc$MIKTA
character(0)

$reduc$DAC
character(0)

$reduc$OECD
character(0)

$reduc$Cwth
character(0)

$reduc$N11
[1] "IDN"

$reduc[[10]]
character(0)

$reduc[[11]]
[1] "FRA, USA"

$reduc[[12]]
[1] "CHN, RUS"

$reduc[[13]]
[1] "GBR"

$reduc[[14]]
[1] "DEU, JPN"

$reduc[[15]]
[1] "BRA"

$reduc[[16]]
[1] "IND"

$reduc[[17]]
[1] "CAN"

$reduc[[18]]
[1] "ZAF"

$reduc[[19]]
[1] ""

$reduc[[20]]
character(0)

$reduc[[21]]
[1] "AUS"

$reduc[[22]]
character(0)

$reduc[[23]]
[1] "KOR"

$reduc[[24]]
[1] "MEX, TUR"

$reduc[[25]]
[1] "ARG, SAU"
```

Partial ordering of the Concepts

A *hierarchy* of concepts is given by the sub–superconcept relation

$$(A, B) \leq (A_2, B_2) \quad \Leftrightarrow \quad A_1 \subseteq A_2 \qquad (\Leftrightarrow \quad B_1 \subseteq B_2)$$

Concept lattice of the context

- built from the hierarchy structure of concepts
- The greatest lower bound of the meet and the least upper bound of the join are defined for an index set T as

$$\bigwedge_{t \in T} (A_t, B_t) = \Big(\bigcap_{t \in T} A_t, \ (\bigcup_{t \in T} B_t)'' \Big)$$

$$\bigvee_{t \in T} (A_t, B_t) = \Big((\bigcup_{t \in T} A_t)'', \ \bigcap_{t \in T} B_t \Big)$$

Partial order of concepts

```r
# construct hierarchy of concepts
g20gcpo <- partial.order(g20gc, type = "galois")
```

```
                {P5} {} {G4} {} {G7} {ITA} {BRICS} {} {MIKTA} {} {DAC} {} {OECD} {} {Cwth} {}
{P5} {}               1       0          0          0           0         0          0          0
{G4} {}               0       1          0          0           0         0          0          0
{G7} {ITA}            0       0          1          0           0         1          1          0
{BRICS} {}            0       0          0          1           0         0          0          0
{MIKTA} {}            0       0          0          0           1         0          0          0
{DAC} {}              0       0          0          0           0         1          1          0
{OECD} {}             0       0          0          0           0         0          1          0
{Cwth} {}             0       0          0          0           0         0          0          1
{N11} {IDN}           0       0          0          0           1         0          0          0
10                    1       1          1          1           1         1          1          1
{} {FRA, USA}         1       0          1          0           0         1          1          0
{} {CHN, RUS}         1       0          0          1           0         0          0          0
{} {GBR}              1       0          1          0           0         1          1          1
{} {DEU, JPN}         0       1          1          0           0         1          1          0
{} {BRA}              0       1          0          1           0         0          0          0
{} {IND}              0       1          0          1           0         0          0          1
{} {CAN}              0       0          1          0           0         1          1          1
{} {ZAF}              0       0          0          1           0         0          0          1
19                    0       0          0          0           1         1          1          0
20                    0       0          0          0           1         0          1          0
{} {AUS}              0       0          0          0           1         1          1          1
22                    0       0          0          0           0         1          1          1
{} {KOR}              0       0          0          0           1         1          1          0
{} {MEX, TUR}         0       0          0          0           1         0          1          0
{} {ARG, SAU}         0       0          0          0           0         0          0          0
```

(Extract)

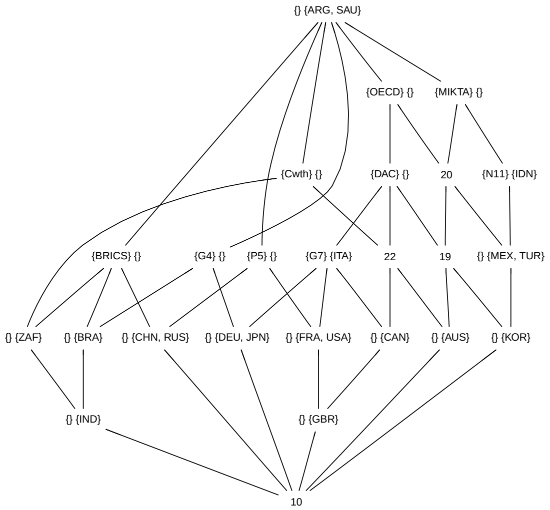
Galois derivations and partial ordering

```r
# structure of g20gc object created with a reduced labeling
str(g20gc)
```

```
List of 2
 $ full :List of 25
  ..$ P5                                    : chr "CHN, FRA, GBR, RUS, USA"
  ..$ G4                                    : chr "BRA, DEU, IND, JPN"
  ..$ DAC, G7, OECD                         : chr "CAN, DEU, FRA, GBR, ITA, JPN, USA"
  ..$ BRICS                                 : chr "BRA, CHN, IND, RUS, ZAF"
  ..$ MIKTA                                 : chr "AUS, IDN, KOR, MEX, TUR"
  ..$ DAC, OECD                             : chr "AUS, CAN, DEU, FRA, GBR, ITA, JPN,
  ..$ OECD                                  : chr "AUS, CAN, DEU, FRA, GBR, ITA, JPN,
  ..$ Cwth                                  : chr "AUS, CAN, GBR, IND, ZAF"
  ..$ MIKTA, N11                            : chr "IDN, KOR, MEX, TUR"
  ..$ BRICS, Cwth, DAC, G4, G7, MIKTA, N11, OECD, P5: chr(0)
...

..- attr(*, "class")= chr [1:2] "Galois" "full"
 $ reduc:List of 25
  ..$ P5   : chr(0)
  ..$ G4   : chr(0)
  ..$ G7   : chr "ITA"
  ..$ BRICS: chr(0)
  ..$ MIKTA: chr(0)
  ..$ DAC  : chr(0)
  ..$ OECD : chr(0)
  ..$ Cwth : chr(0)
  ..$ N11  : chr "IDN"
  ..$      : chr(0)
...
```

Concept lattice of the context

```
# plot hierarchy of concepts as lattice diagram
diagram(g20gcpo)
```



Order Filters and Order Ideals

formal definition

- Let $(P, \leq)$ be an ordered set, and a, b are elements in P
- A non-empty subset U [resp. D] of P is an upset [resp. downset] called a *order filter* [resp. *order ideal*] if, for all $a \in P$ and $b \in U$ [resp. D]

  $$b \leq a \quad \text{implies} \quad a \in U \qquad\qquad [\ \text{resp. } a \leq b \quad \text{implies} \quad a \in D\ ]$$

- The upset $\uparrow x$ formed for all the upper bounds of $x \in P$ is called a *principal order filter* generated by x
- Dually, $\downarrow x$ is a *principal order ideal* with all the lower bounds of $x \in P$

☛ order filters and order ideals not coinciding with P are called *proper*

Order Filters and Order Ideals

```
# find principal order filters in the partial order context
formals("fltr")
```

```
$x


$PO


$rclos
[1] TRUE

$ideal
[1] FALSE
```

Principal Order Filters

```r
# principal order filter of first concept in g20gcpo
fltr(1, g20gcpo)
```

```
$`1`
[1] "{P5} {}"

$`25`
[1] "{} {ARG, SAU}"
```

```r
# another option is to use intent labels of different concepts
fltr(c("P5", "BRICS"), g20gcpo)
```

```
$`1`
[1] "{P5} {}"

$`4`
[1] "{BRICS} {}"

$`25`
[1] "{} {ARG, SAU}"
```

Principal Order Ideals

```r
# principal order ideal of the first concept in g20gcpo
fltr("P5", g20gcpo, ideal = TRUE)
```

```
$`1`
[1] "{P5} {}"

$`10`
[1] "10"

$`11`
[1] "{} {FRA, USA}"

$`12`
[1] "{} {CHN, RUS}"

$`13`
[1] "{} {GBR}"
```

$5b.$ Multilevel networks

Example 6: Group of Twenty

Network tie interlock concepts

- *Social structure*: in *simple* networks, configuration made of ties between actors
- *Positional system*: in *multilevel* networks, reduced structures of actors and events
- *Relational structure*: in *multiplex* networks, configuration made of interrelations between relations
- *Role structure*: relational system of aggregated relations

➠ *use of algebraic objects to represent relational and role structures*

➠ *apply relational and role structure notions to multilevel networks*

Multilevel network

A formal definition

A *multilevel network* X^M for vertex sets N (domain), M (codomain), and edge sets E

$$X^M = \langle\, N,\ M,\ E_N,\ E_M,\ E_{N\times M}\,\rangle$$

- An *affiliation* network is $X^B = \langle N, M, E_{N\times M}\rangle$, where $E_N, E_M = \varnothing$
- A *valued* network $X^V = \langle N, E, V\rangle$ with V for weights
- *Multiplex* systems add R for types of E

Constructing G20 affiliation network

```r
# 'G20' is a data frame of G20 countries formal context
load("./data/G20.rda")
```

```
    P5 G4 G7 BRICS MITKA DAC OECD Cwth N11
ARG  0  0  0     0     0   0    0    0   0
AUS  0  0  0     0     1   1    1    1   0
BRA  0  1  0     1     0   0    0    0   0
CAN  0  0  1     0     0   1    1    1   0
CHN  1  0  0     1     0   0    0    0   0
DEU  0  1  1     0     0   1    1    0   0
FRA  1  0  1     0     0   1    1    0   0
GBR  1  0  1     0     0   1    1    1   0
IDN  0  0  0     0     1   0    0    0   1
IND  0  1  0     1     0   0    0    1   0
ITA  0  0  1     0     0   1    1    0   0
JPN  0  1  1     0     0   1    1    0   0
KOR  0  0  0     0     1   1    1    0   1
MEX  0  0  0     0     1   0    1    0   1
RUS  1  0  0     1     0   0    0    0   0
SAU  0  0  0     0     0   0    0    0   0
TUR  0  0  0     0     1   0    1    0   1
USA  1  0  1     0     0   1    1    0   0
ZAF  0  0  0     1     0   0    0    1   0
```

Plot bipartite graph of G20 affiliation network

```r
# event clustering information
ec <- c(1, 1, 2, 0, 1, 2, 1, 1, 1)

# actor clustering (IMF economic classification of countries)
ac <- c(0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0)
ac <- replace(ac, ac==0, "Emerging")
ac <- replace(ac, ac==1, "Advanced")
```

```
 [1] "Emerging" "Advanced" "Emerging" "Advanced" "Emerging" "Advanced" "Advanced"
 [8] "Advanced" "Emerging" "Emerging" "Advanced" "Advanced" "Advanced" "Emerging"
[15] "Emerging" "Emerging" "Emerging" "Advanced" "Emerging
```

```r
# bipartite graph with a vertical layout with clustering information
bmgraph(G20, layout = "bipc", clu = list(ac, ec), cex = 4)
```

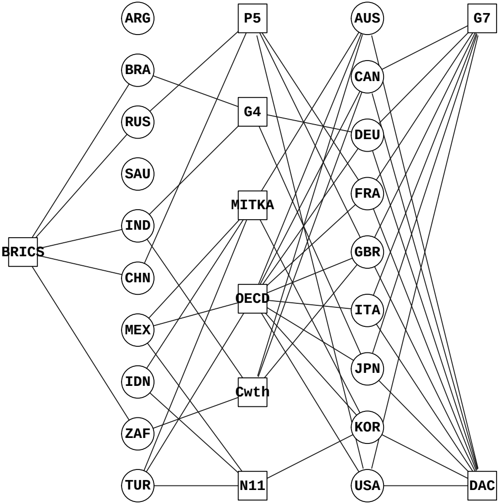
- Plot as a “clustered” bipartite graph
- Clustering information is given as a list since it is for two domains

Group of Twenty (G20) affiliations

clustered bipartite graph

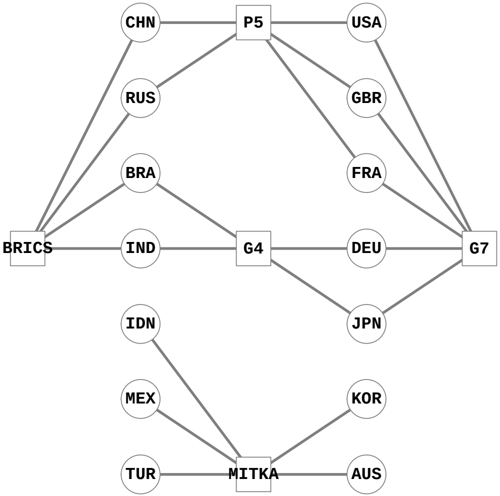
circles: *actors* in N

squares: *events* in M



G20 with non-overlapping "bridge" organisations

clustered bipartite graph



Classes of actors in G20 with “bridges”

1. G7
2. BRICS
3. MITKA

AUS	BRA	CHN	DEU	FRA	GBR	IDN	IND	JPN	KOR	MEX	RUS	TUR	USA
3	2	2	1	1	1	3	2	1	3	3	2	3	1

Constructing G20 affiliation network with bridges

```
# Option: P5 G4 MITKA none
acb <- factor(ac, levels = c("P5", "G4", "MITKA", "none"))
acb[which(G20[,1]==1)] <- "P5"   ; acb[which(G20[,2]==1)] <- "G4"
acb[which(G20[,5]==1)] <- "MITKA"; acb[which(is.na(acb))] <- "none"
```

```
[1] none MITKA G4 none P5 G4 P5 P5 MITKA G4 none G4 MITKA MITKA P5 none MITKA P5 none
Levels: P5 G4 MITKA none
```

```
# bridge organisations
bridges <- which(acb!="none")
```

```
# extract from G20 only countries affiliated to bridge organisations
G20b <- G20[bridges, c(1:5)]
```

Plot binomial projection of bridged affiliation network

```
bmgraph(G20b, layout = "force", seed = 321)
```

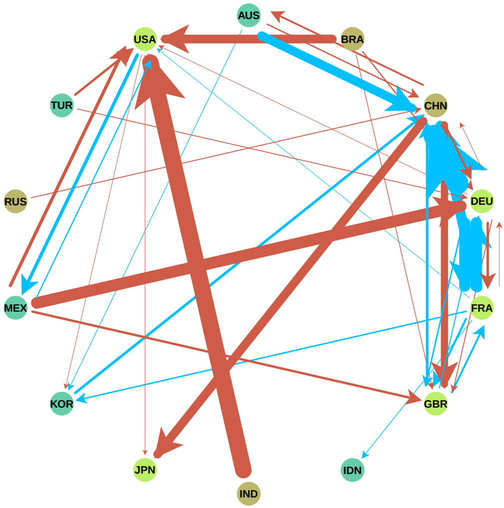
Plotting skeleton G20 trade network with bridges

```r
# load G20net skeleton trade network data
load(file = "./data/G20net.rda")

# extract of the G20 m&h multiplex network (exclude)
G20Bnet <- G20net[bridges, bridges, ]

# define scopes and clustering information
club <- c(3, 2, 2, 1, 1, 1, 3, 2, 1, 3, 3, 2, 3, 1)
scp <- list(cex = 4, pos = 0, fsize = 8, fstyle = "bold")
scpb <- list(vcol = c("#BCEE68","#BDB76B","#66CDAA"),
             ecol = c("#00BFFF","#CD5B45"))
```

```r
# plot graph combining different types of scopes
multigraph(G20Bnet, valued = TRUE, scope = c(scp, scpb), clu = club)
```

G20b Trade network X^{V}_{G20b}

valued skeleton after triangle inequality



Directed edges E_N: blue for fresh milk (R_1), red for honey (R_2)

Multilevel structure of G20 with bridges

co-membership

Functions `mlvl()` and `mlgraph()` allow constructing and plotting multilevel structures

```
# co-membership network matrix
G20Bcn <- mlvl(y = G20B, type = "cn")

# plot valued graph of co-membership network (default circular layout)
mlgraph(G20Bcn, valued = TRUE)
```

However, function `multigraph()` supports multilevel networks as well

```
# valued graph with co-membership values
multigraph(G20Bcn, valued = TRUE, values = TRUE, undRecip = TRUE)
```

Multilevel structure of G20 with bridges

actors co-affiliation

```
# multilevel with co-affiliation of actors
G20Bcn2 <- mlvl(x = G20Bnet, y = G20B, type = "cn2")
```

```
...

$lbs
$lbs$dm
 [1] "AUS" "BRA" "CHN" "DEU" "FRA" "GBR" "IDN" "IND" "JPN" "KOR" "MEX" "RUS" "TUR" "USA"

$lbs$cdm
[1] "P5"    "G4"    "G7"    "BRICS" "MITKA"


$modes
[1] "1M" "1M" "2M"

attr(,"class")
[1] "Multilevel" "cn2"
```

```
# plot multilevel structure
mlgraph(G20Bcn2)
```

Clustering information & actors co-affiliation

multilevel structure of g20b with bridges

Additional clustering information for events and `club` still for actors

```
# clustering information with events (may not needed in future)
club2 <- list(club, rep(1, ncol(G20B)) )
```

```
[[1]]
 [1] 3 2 2 1 1 1 3 2 1 3 3 2 3 1

[[2]]
[1] 1 1 1 1 1
```

```
# plot multilevel network as circular layout and updated clustering info
mlgraph(G20Bcn2, valued = TRUE, scope = c(scp, scpb), clu = club2)
```

Multilevel structure with binomial projection

```
# multilevel with binomial projection
G20Bbp <- mlvl(x = G20Bnet, y = G20B, type = "bpn")


$mlnet
, , M

      AUS BRA   CHN  DEU  FRA  GBR IDN IND JPN KOR  MEX RUS TUR USA P5 G4 G7 BRICS MITKA
AUS     0   0  5829    0    0    0   0   0   0 412    0   0   0   0  0  0  0     0     0
BRA     0   0     0    0    0    0   0   0   0   0    0   0   0   0  0  0  0     0     0
CHN     0   0     0    0    0    0   0   0   0   0    0   0   0   0  0  0  0     0     0
DEU     0   0 14417    0 7358  928   0   0   0   0    0   0   0   0  0  0  0     0     0
FRA     0   0  6714 7553    0 1345 443   0   0 921    0   0   0 254  0  0  0     0     0
GBR     0   0  1734  641 1170    0   0   0   0   0    0   0   0   0  0  0  0     0     0
IDN     0   0     0    0    0    0   0   0   0   0    0   0   0   0  0  0  0     0     0

...


, , 3

      AUS BRA CHN DEU FRA GBR IDN IND JPN KOR MEX RUS TUR USA P5 G4 G7 BRICS MITKA
AUS     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  0     0     1
BRA     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  1  0     1     0
CHN     0   0   0   0   0   0   0   0   0   0   0   0   0   0  1  0  0     1     0
DEU     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  1  1     0     0
FRA     0   0   0   0   0   0   0   0   0   0   0   0   0   0  1  0  1     0     0
GBR     0   0   0   0   0   0   0   0   0   0   0   0   0   0  1  0  1     0     0
IDN     0   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  0     0     1

...
```

Multilevel structure with binomial projection

scope and clustering

Define additional scopes to handle events

```
# shapes and color of vertices for actors and events
scpm <- list(ecol = c("blue","red","orange"), pch = c(21,15), vcol0 = 8,
             vcol = c("#BCEE68","#BDB76B","#66CDAA","#838B8B","#FF7F00"))

# four classes of actors as factor with explicit levels
acc <- factor(ac, levels = c("A-G7", "Advanced", "E-BRICS", "Emerging"))
acc[which(G20[,3]==1)] <- "A-G7"; acc[which(G20[,4]==1)] <- "E-BRICS"

# clustering information for multilevel structure
cluml <- list(acc[bridges], rep(1, nrow(G20B)))
```

```
[[1]]
 [1] Advanced E-BRICS  E-BRICS  A-G7     A-G7     A-G7     Emerging E-BRICS  A-G7     Advanced
[11] Emerging E-BRICS  Emerging A-G7
Levels: A-G7 Advanced E-BRICS Emerging

[[2]]
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Multilevel structure with binomial projection

plotting

Multilevel network with binomial projection (updated clustering information)

```
# plot with default circular layout and recycling scope
mlgraph(G20Bbp, scope = c(scp, scpm), clu = cluml)
```

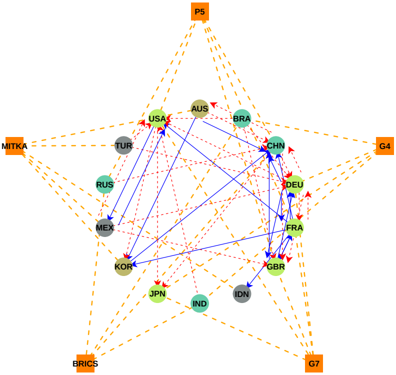
```
# clustering information of the two domains in 'G20Bbp'
nr <- c(rep(1,nrow(G20B)), rep(2,ncol(G20B)))


# [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
```

Plot multilevel graph with binomial projection

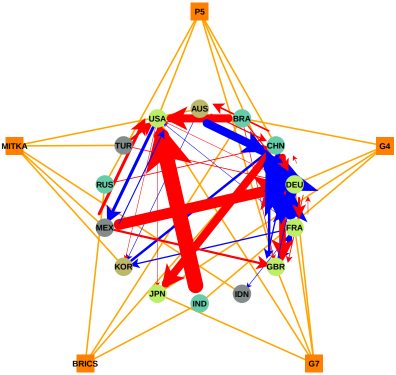
```
# concentric layout with two radii and recycled scopes
mlgraph(G20Bbp, layout = "conc", nr = nr, scope = c(scp,scpm), clu = cluml)
mlgraph(G20Bbp, layout = "conc", nr = nr, scope = c(scp,scpm), clu = cluml,
        valued = TRUE)
```

Multilevel structure with binomial projection

concentric layout



valued



Positional analysis of the Multilevel structure

Functions `reduc()` to reduce array structures

```
# positional system actors with clustering information
PSG20Ba <- reduc(G20Bnet, valued = TRUE, lbs = c("G7.C","BRICS.C","MITKA.C"),
                  clu = club)
```

```
, , M

        G7.C BRICS.C MITKA.C
G7.C    19249   22865    3538
BRICS.C     0       0       0
MITKA.C   752    7572     412

, , H

        G7.C BRICS.C MITKA.C
G7.C     3050     296     293
BRICS.C 29577     584    1187
MITKA.C 13477     860       0
```

```
# Positional system events ('row' option for two-mode networks)
PSG20Be <- reduc(G20B, valued = TRUE, lbs = c("G7.C","BRICS.C","MITKA.C"),
                  clu = club, row = TRUE)
```

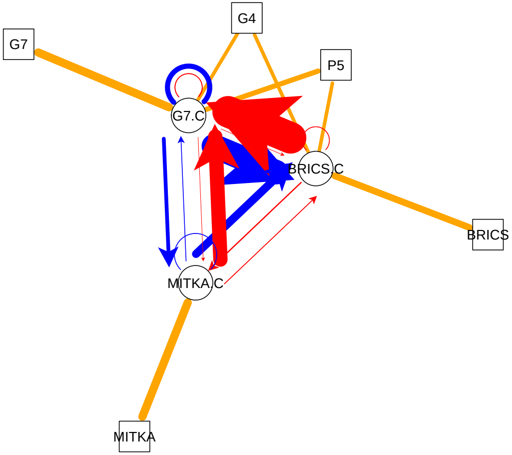
Multilevel structure of positional system for G20B

```
# multilevel structure with binomial projection and symmetric co-domain
PSG20Bbp <- mlvl(x = PSG20Ba, y = PSG20Be, type = "bpn", symCdm = TRUE)


$mlnet
, , m

         G7.C BRICS.C MITKA.C P5 G4 G7 BRICS MITKA
G7.C    19249   22865    3538  0  0  0     0     0
BRICS.C     0       0       0  0  0  0     0     0
MITKA.C   752    7572     412  0  0  0     0     0
P5          0       0       0  0  0  0     0     0
G4          0       0       0  0  0  0     0     0

 ...

, , 3

        G7.C BRICS.C MITKA.C P5 G4 G7 BRICS MITKA
G7.C       0       0       0  3  2  5     0     0
BRICS.C    0       0       0  2  2  0     4     0
MITKA.C    0       0       0  0  0  0     0     5
P5         3       2       0  0  0  0     0     0
G4         2       2       0  0  0  0     0     0
G7         5       0       0  0  0  0     0     0
BRICS      0       4       0  0  0  0     0     0
MITKA      0       0       5  0  0  0     0     0

...
```

Valued multilevel structure of positional system for G20B

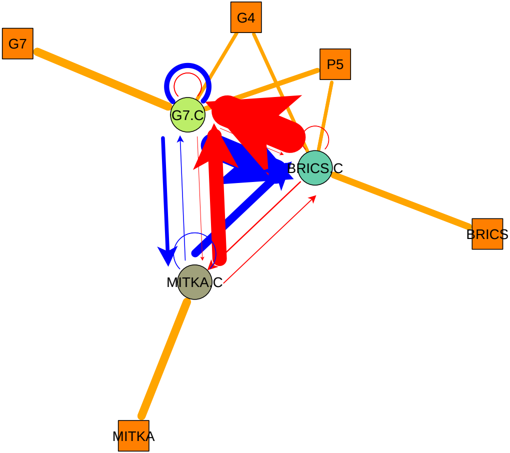
edge colors are for milk, honey, and affiliation ties

```
# plot multilevel graph of positional system
mlgraph(PSG20Bbp, valued = TRUE, layout = "force", seed = 1, cex = 6, pos = 0,
        ecol = c("blue","red","orange"), fsize = 11)
```

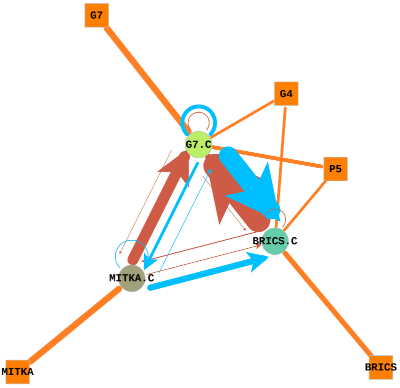


Valued multilevel structure of positional system for G20B

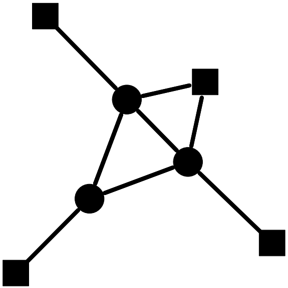
```
# define scopes multilevel positional system and plot
scpps <- list(cex=6, ecol=c("blue","red","orange"), pos=0, fsize=11)
scpps2 <- list(vcol=c("#BCEE68","#66CDAA","#A0A17B","#FF7F00"),
              clu=c(1:3, rep(4,5)))
mlgraph(PSG20Bbp, layout="force", seed=1, valued=TRUE, scope=c(scpps,scpps2))
```



Positional systems of X_{G20b}^{M}



valued multilevel multigraph



skeleton with Structural equivalence applied

Multilevel positional system

Image matrices for X^{M}_{G20b} made with X^{V}_{G20b} and class affiliations in X^{B}_{G20}

	G7.C	BRICS.C	MITKA.C
G7.C	19249	22865	3538
BRICS.C	0	0	0
MITKA.C	752	7572	412

Fresh Milk

	G7.C	BRICS.C	MITKA.C
G7.C	3050	296	293
BRICS.C	29577	584	1187
MITKA.C	13477	860	0

Honey

	P5	G4	G7	BRICS	MITKA
G7.C	3	2	5	0	0
BRICS.C	2	2	0	4	0
MITKA.C	0	0	0	0	5

Affiliation of classes to bridge organizations

Algebraic analysis of multilevel configuration

Role structure of G20 with bridges

G20 Countries network is a multilevel, multiplex and valued structure

➟ complex organisational network

☛ Multilevel version of G20 Countries network with bridges is represented as X^{M}_{G20b}

Partially ordered semigroup of X^{M}_{G20b} positional system

Cayley graph

Generators

m: milk

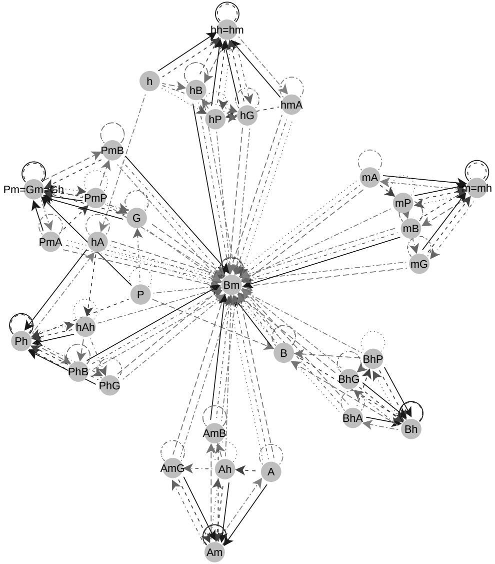
h: honey

G: G7

B: BRICS

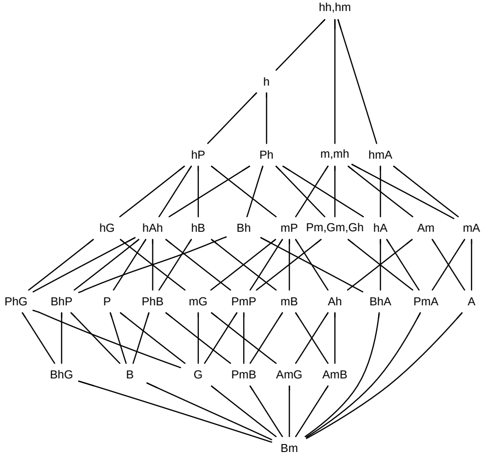
A: MITKA

P: P5 and G4



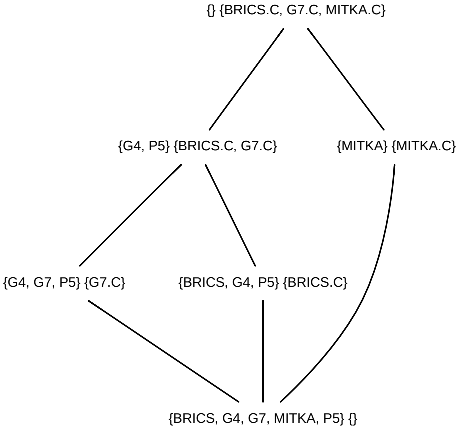
Partially ordered semigroup of X^{M}_{G20b} positional system

Inclusion diagram



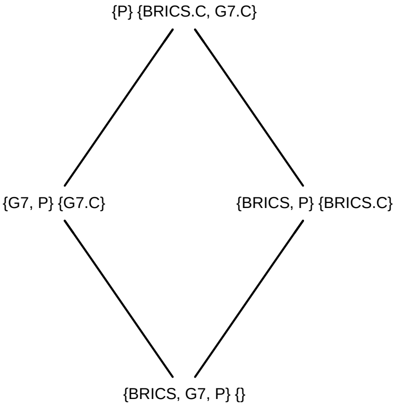
Concept diagram of X_{G20b}^{M} positional system

full labeling

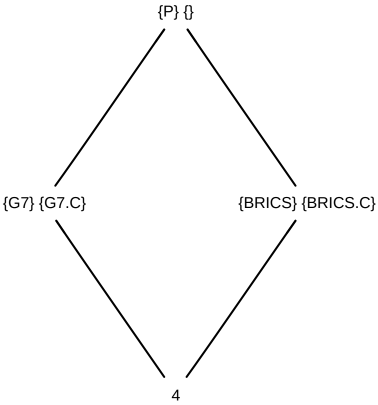


Concept diagrams of X^{M}_{G20b}

with a two element positional system



full labeling



reduced labeling

+ Example 6c: G20 Trade network

Relational structure (valued)

Many-valued context

Relational structure in valued multiplex networks

- Assignments to labeled valued paths with the $\max - \min$ composition
  - ➟ minimum value of sending/receiving scores in nodes
  - ➟ then the maximum of these values

- For the G20 Trade network of milk (M) & honey (H)

$$\mathsf{M} \circ \mathsf{H} = \max_k \{\min\big(w_{\mathsf{M}}(i,k),\ w_{\mathsf{H}}(k,j)\big)\} = \mathsf{MH}$$

☛ `multiplex` supports valued networks

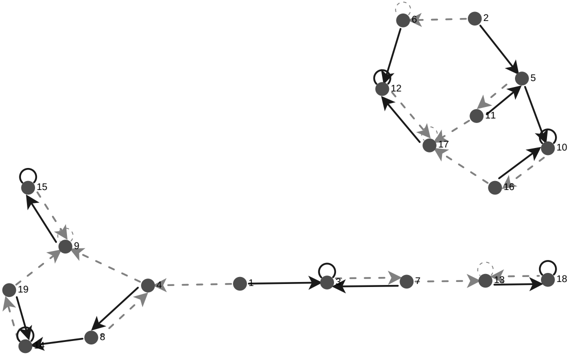
Relational structure in valued multiplex networks

```
load(file = "./data/vnet.rda")
```

```
vnet

, , M

      G7 BRICS MITKA
G7    19    23     4
BRICS  0     0     0
MITKA  1     8     0

, , H

      G7 BRICS MITKA
G7     3     0     0
BRICS 30     1     1
MITKA 13     1     0
```

Relational structure in valued multiplex networks

```r
# relational structure in valued network
semigroup(vnet, valued = TRUE)

# or pipeing to plot semigroup with max-min product
require("magrittr")
vnet %>% semigroup(valued = TRUE) %>% ccgraph(rot = 90, cex = 2, lwd = 2)
```



Many-valued context
(Wille, 1982)

- A *many-valued context*, $\mathbb{K}^V$ is defined as

$$\mathbb{K}^V = (\, G,\ M,\ W,\ I \,)$$

  - ➟ G is the object set
  - ➟ M is the many-valued attributes
  - ➟ W are "weights" or attribute values
  - ➟ I is the incidence relation.

☛ the context is said to be an k*-valued context* when W has k elements

Many-valued context of G20 network

economic and socio-demographic indicators

```r
# four-valued context
load(file = "./data/G20mv.rda")
```

```
G20mv

    T  N  G  H  P  A
ARG vl vl l  l  vl l
AUS vl vl vh vh vl h
BRA vl l  l  l  l  h
CAN l  l  vh vh vl h
CHN vh vh vl vl vh h
DEU h  h  vh vh l  vl
FRA l  l  h  h  vl vl
GBR l  l  h  vh vl vl
IDN vl vl vl vl h  l
IND l  l  vl vl vh l
ITA l  l  h  h  vl vl
JPN h  h  h  h  l  vl
KOR l  vl h  h  vl vl
MEX l  vl vl l  l  l
RUS l  vl l  l  l  vh
SAU vl vl l  h  vl l
TUR vl vl vl l  vl vl
USA vh vh vh vh h  h
ZAF vl vl vl vl vl vl
```

Conceptual scaling

many-valued context of G20 network

	very-high	*high*	*low*	*very-low*
very-high	1	0	0	0
high	0	1	0	0
low	0	0	1	0
very-low	0	0	0	1

$\leq$*very-high*	$\leq$*high*	$\geq$*low*	$\geq$*very-low*
1	1	0	0
0	1	0	0
0	0	1	0
0	0	1	1

Nominal $\mathbb{N}_n = \langle n, n, = \rangle$

Inter-ordinal $\mathbb{I}_n = \langle n, n, \leq \mid n, n, \geq \rangle$

Function `cscl()` from `multiplex`

```
# where 'scl' is a scaling matrix
cscl(G20mv, scl, sep = ".")
```

☛ then plot Concept lattice of output, and apply order filters and order ideals

Pathfinder semiring and Triangle inequality

one-mode valued networks

For the analysis of adjacency matrices:

- *Pathfinder semiring* for symmetric valued relations

  ➟ matrix reflects the "proximity" between pairs of network members

- *Triangle inequality* for asymmetric valued ties

  ➟ then *salient* structure of valued network

Use functions `pfvn()` and `ti()` from `multiplex`

References

- Sampson, FC *A novitiate in a period of change: An experimental and case study of social relationships*. Cornell University. 1969
- White, HC *An Anatomy of Kinship: Mathematical models for structures of cumulated roles*. Prentice-Hall. 1963
- Pattison, PE *Algebraic Models for Social Networks*. Cambridge University Press. 1993
- Ganter, B and R Wille *Formal Concept Analysis – Mathematical Foundations*. Springer. 1996
- Ostoic, JAR *Algebraic Analysis of Social Networks*. Wiley. 2021
- Padgett, JF and CK Ansell 'Robust action and the rise of the Medici, 1400–1434,' *American Journal of Sociology* **98**(6), 1259–1319. 1993
- R Development Core Team, ***R**: A language and environment for statistical computing*, version 3.6.3

Thanks!

Research programme activities at the Department of History and Classical Studies, AU

Center for Digital History Aarhus – CEDHAR

github.com/mplex

jaro@cas.au.dk

(Q & A)