Königsberg bridges

Network Analysis
Structure of Networks 1
Subnetworks

Vladimir Batagelj

University of Ljubljana

ECPR Summer School, July 30 – August 16, 2008

Faculty of Social Sciences, University of Ljubljana

version: August 11, 2008 / 03 : 56

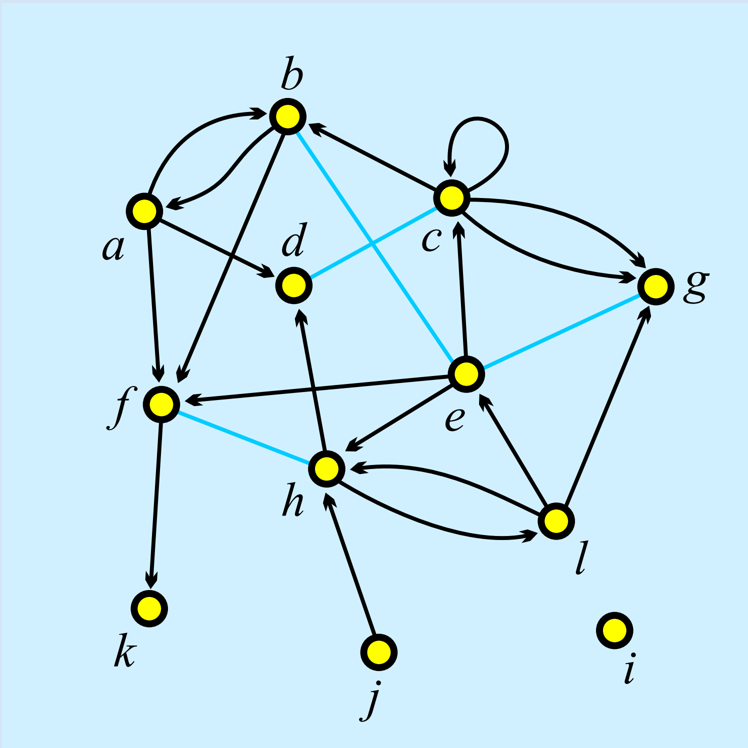
Outline

Approaches to large networks

In analysis of a *large* network (several thousands or millions of vertices, the network can be stored in computer memory) we can't display it in its totality; also there are only few algorithms available.

To analyze a large network we can use statistical approach or we can identify smaller (sub) networks that can be analyzed further using more sophisticated methods.

Degrees



degree of vertex v, $\deg(v)$ = number of lines with v as end-vertex;

indegree of vertex v, $\text{indeg}(v)$ = number of lines with v as terminal vertex (end-vertex is both initial and terminal);

outdegree of vertex v, $\text{outdeg}(v)$ = number of lines with v as initial vertex.

initial vertex $v \Leftrightarrow \text{indeg}(v) = 0$

terminal vertex $v \Leftrightarrow \text{outdeg}(v) = 0$

$$n = 12,\ m = 23,\ \text{indeg}(e) = 3,\ \text{outdeg}(e) = 5,\ \deg(e) = 6$$

$$\sum_{v\in\mathcal{V}} \text{indeg}(v) = \sum_{v\in\mathcal{V}} \text{outdeg}(v) = |\mathcal{A}| + 2|\mathcal{E}|,\ \sum_{v\in\mathcal{V}} \deg(v) = 2|\mathcal{L}| - |\mathcal{E}_0|$$

Statistics

Input data

- numeric → **`vector`**
- ordinal → **`permutation`**
- nominal → **`clustering`** (partition)

Computed properties

global: number of vertices, edges/arcs, components; maximum core number, . . .

local: degrees, cores, indices (betweeness, hubs, authorities, . . .)

inspections: partition, vector, values of lines, . . .

Associations between computed (structural) data and input (measured) data.

… Statistics

The global computed properties are reported by **`Pajek`**'s commands or can be seen using the **`Info`** option. In *repetitive* commands they are stored in vectors.

The local properties are computed by **`Pajek`**'s commands and stored in vectors or partitions. To get information about their distribution use the **`Info`** option.

As an example, let us look at The Edinburgh Associative Thesaurus network. The EAT is a network of word association as collected from subjects (students). The weight on the arcs is the count of word associations.

```
File/Network/Read  eatRS.net
Info/Network/General
```

It has 23219 vertices and 325624 arcs (564 loops); number of lines with value=1 is 227481.

… Statistics

To identify the vertices with the largest degree:

```
Net/Partitions/Degree/All
Partition/Make vector
Info/Vector  +10
```

The largest degrees have the vertices:

	vertex	deg	label
1	12720	1108	ME
2	12459	1074	MAN
3	8878	878	GOOD
4	18122	875	SEX
5	13793	803	NO
6	13181	799	MONEY
7	23136	732	YES
8	15080	723	PEOPLE
9	13948	720	NOTHING
10	22973	716	WORK

Statistics / Pajek and R

Pajek (0.89 and higher) supports interaction with statistical program R and the use of other external programs as tools (menu `Tools`).

In **Pajek** we determine the degrees of vertices and submit them to R

```
info/network/general
Net/Partitions/Degree/All
Partition/Make Vector
Tools/Program R/Send to R/Current Vector
```

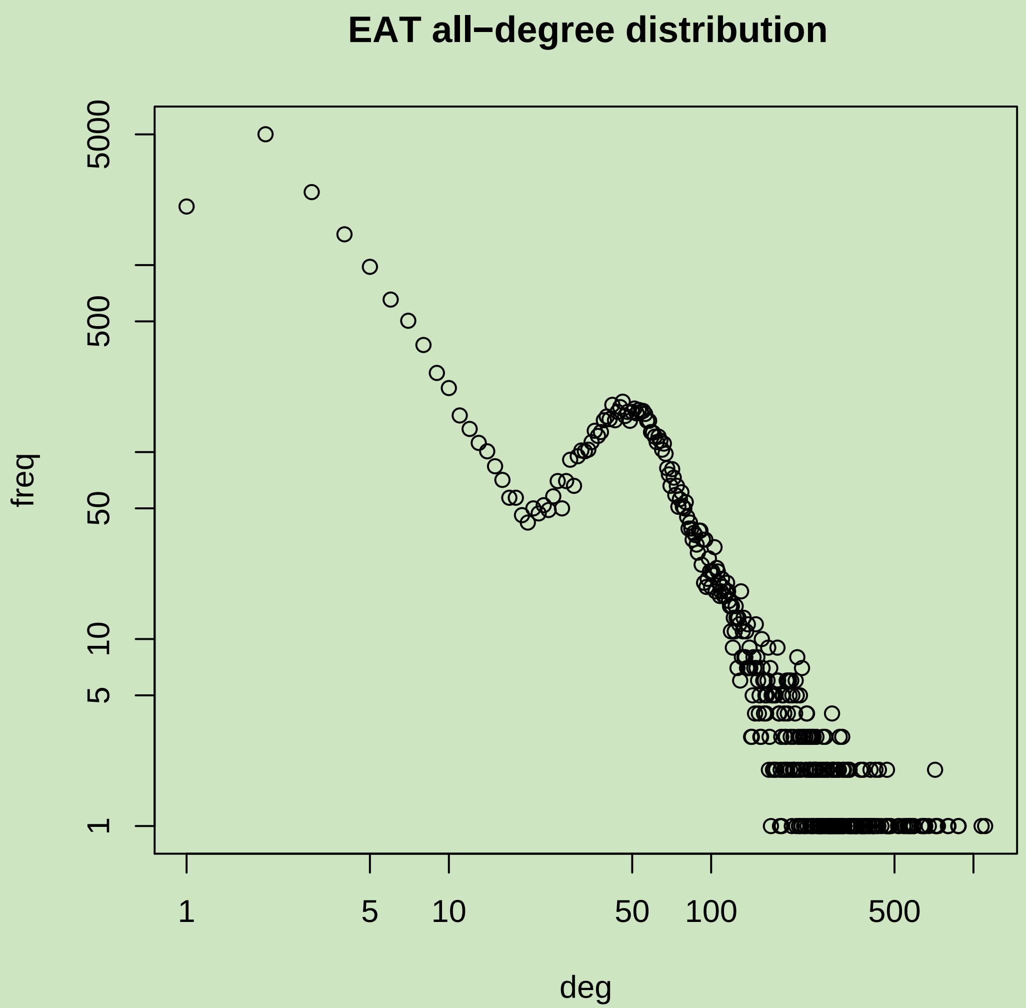
In R we determine their distribution and plot it

```
summary(v2)
t <- tabulate(v2)
c <- t[t>0]
i <- (1:length(t))[t>0]
plot(i,c,log='xy',main='degree distribution',
  xlab='deg',ylab='freq')
```

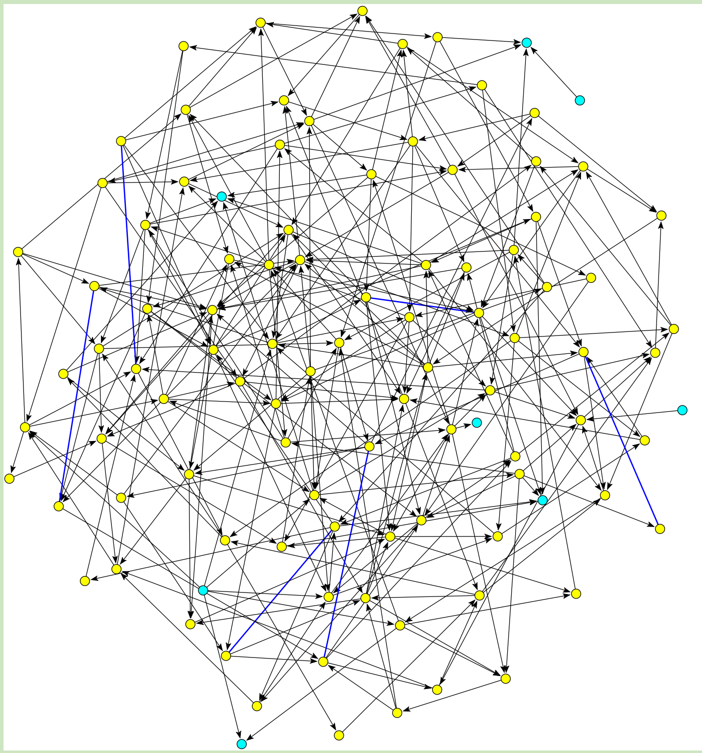
The obtained picture can be saved with `File/Save as` in selected format (PDF or PS for LaTeX; Windows metafile format for inclusion in Word).

Attention! The vertices of degree 0 are not considered by `tabulate`.

EAT all-degree distribution



Erdős and Renyi's random graphs



Erdős and Renyi defined a *random graph* as follows: every possible line is included in a graph with a given probabilty p.

In **Pajek**'s

`Net/Random Network/Erdos-Renyi`

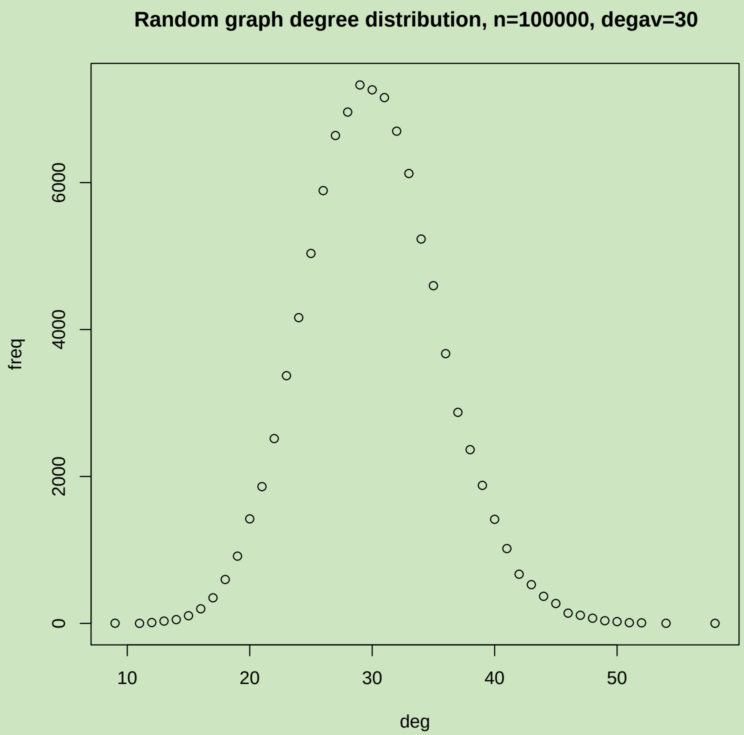
instead of probability p a more intuitive average degree is used

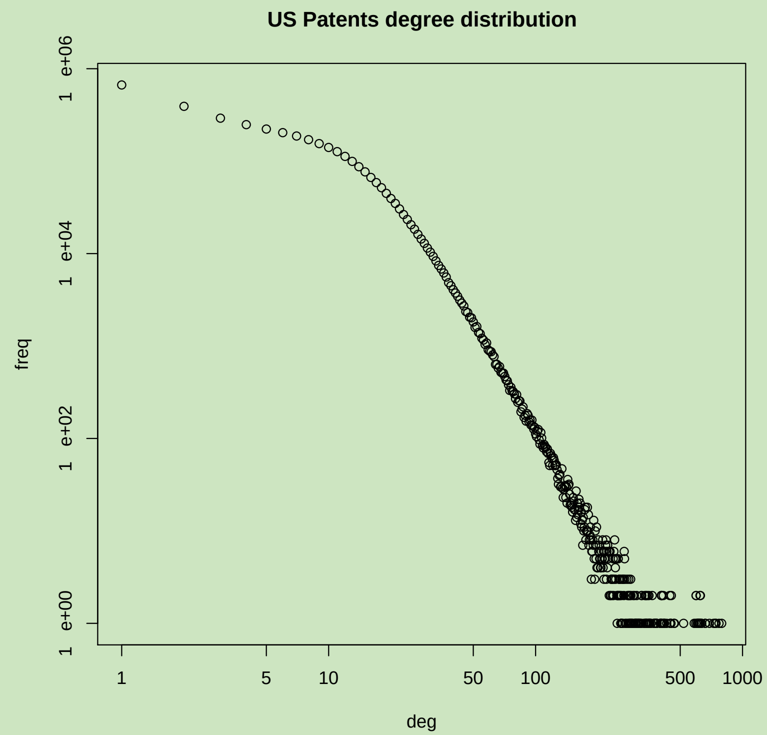
$$\overline{\deg} = \frac{1}{n} \sum_{v \in \mathcal{V}} \deg(v)$$

It holds $p = \frac{m}{m_{max}}$ and, for simple graphs, also $\overline{\deg} = \frac{2m}{n}$.

Random graph in picture has 100 vertices and average degree 3.

Degree distribution





Real-life networks are usually not random in the Erdős/Renyi sense. The analysis of their distributions gave a new view about their structure – Watts (Small worlds), Barabási (nd/networks, Linked).

Example: Snyder and Kick World Trade

The data are available as a **`Pajek`**'s project file

`SaKtrade.paj`

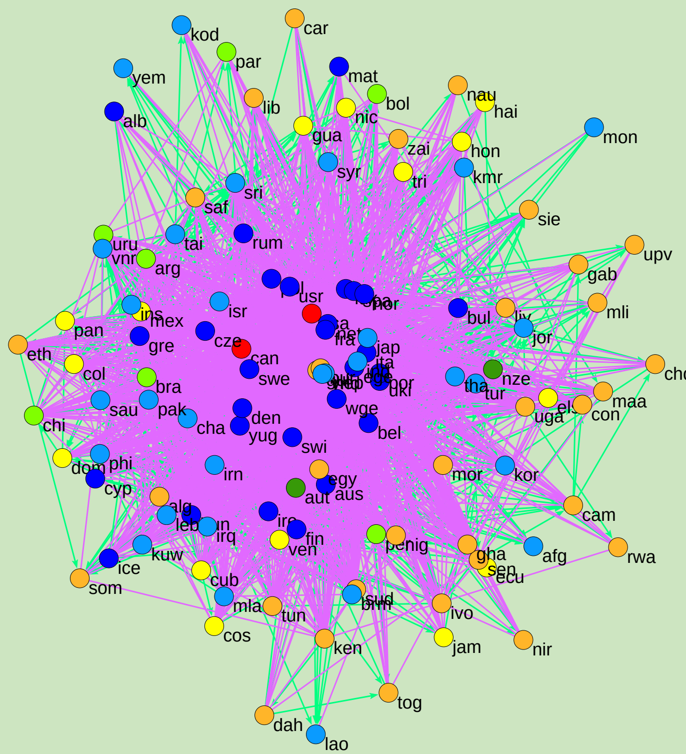
The network consists of trade relations (118 vertices, 515 arcs, 2116 edges). The source of the data is the paper: Snyder, David and Edward Kick (1979). *The World System and World Trade: An Empirical Exploration of Conceptual Conflicts*, Sociological Quaterly, 20,1, 23-36.

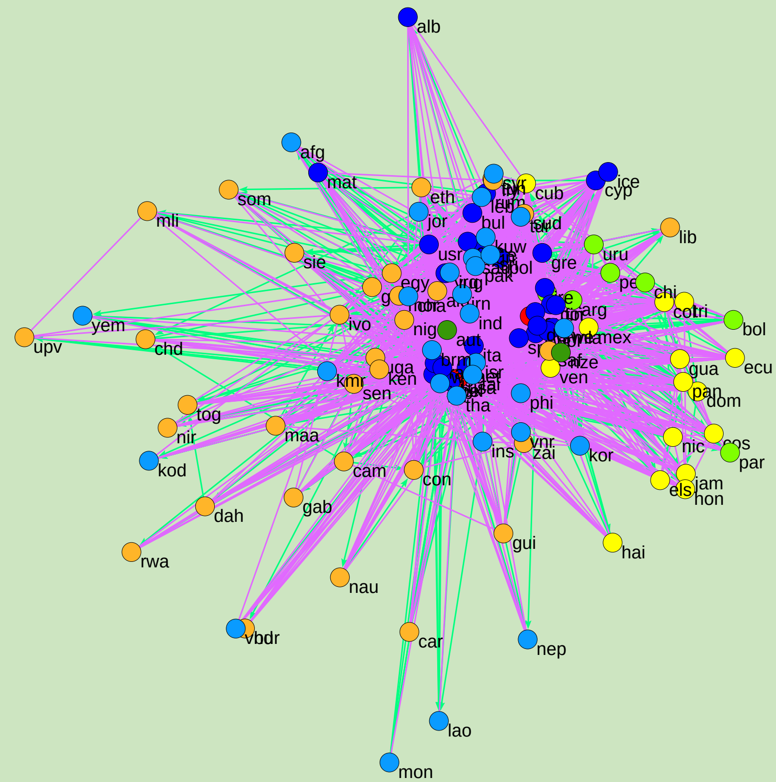
The project file contains also the (sub)continents partition:

1 - Europe, 2 - North America, 3 - Latin America, 4 - South America, 5 - Asia, 6 - Africa, 7 - Oceania.

For latest data see NBER / Feenstra; and for an analysis see Science / Hidalgo et. al.

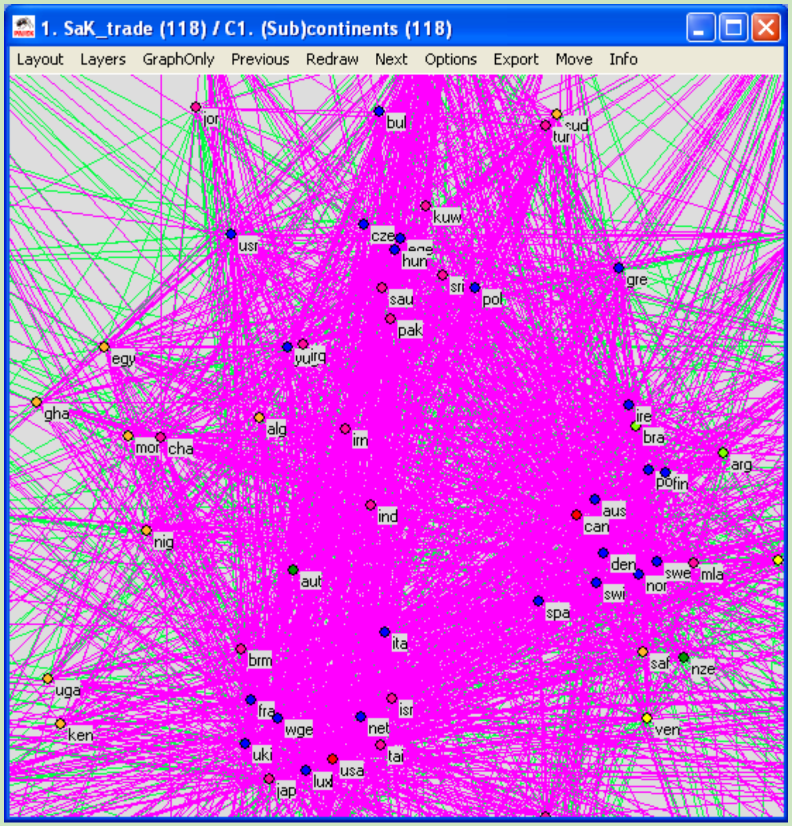
Draw / Partition





```
Draw/Draw Partition
Layout/Energy/Kamada-Kawai/Free
Layout/Energy/Fruchterman Reingold/2D
```

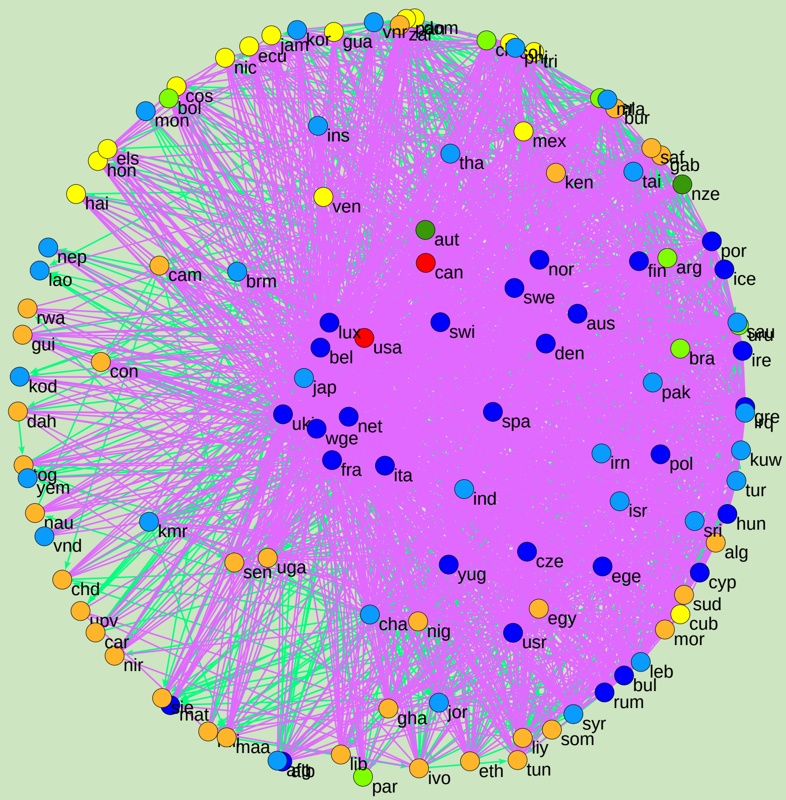
Zoom in

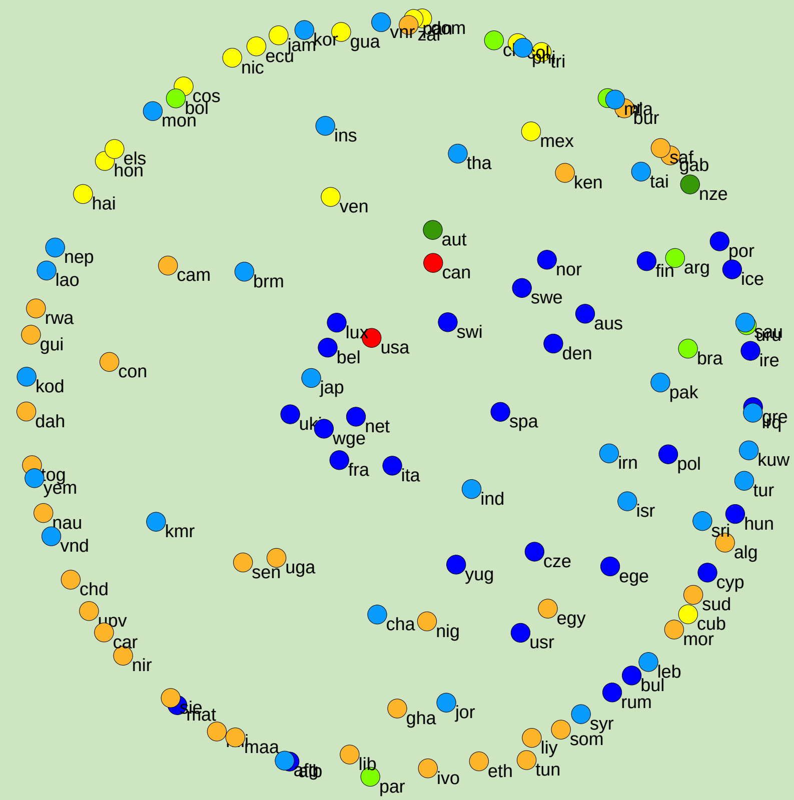


Using right button on the mouse select the zoom area.

To restore the standard view select `Redraw`.

Fruchterman Reingold / factor $= 9$





`Layout/Energy/Fruchterman Reingold/3D`

3D picture / King

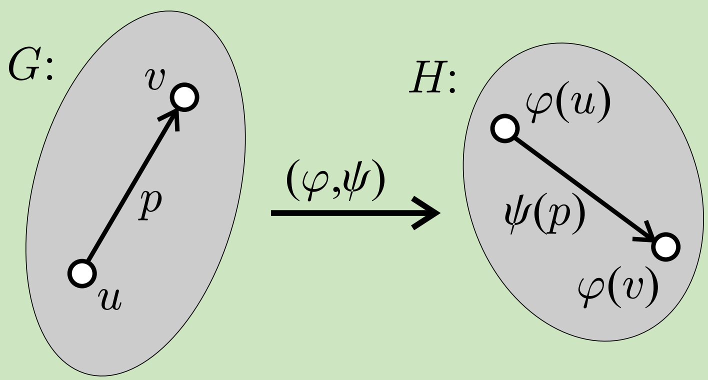
Homomorphisms of graphs

Functions (φ, ψ), $\varphi: \mathcal{V} \to \mathcal{V}'$ and $\psi: \mathcal{L} \to \mathcal{L}'$ determine a *weak homomorphism* of graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ in graph $\mathcal{H} = (\mathcal{V}', \mathcal{L}')$ iff:

$$\forall u, v \in \mathcal{V}\, \forall p \in \mathcal{L} : (p(u : v) \Rightarrow \psi(p)(\varphi(u) : \varphi(v)))$$

and they determine a *(strong) homomorphism* of graph $\mathcal{G}$ in graph $\mathcal{H}$ iff:

$$\forall u, v \in \mathcal{V}\, \forall p \in \mathcal{L} : (p(u, v) \Rightarrow \psi(p)(\varphi(u), \varphi(v)))$$

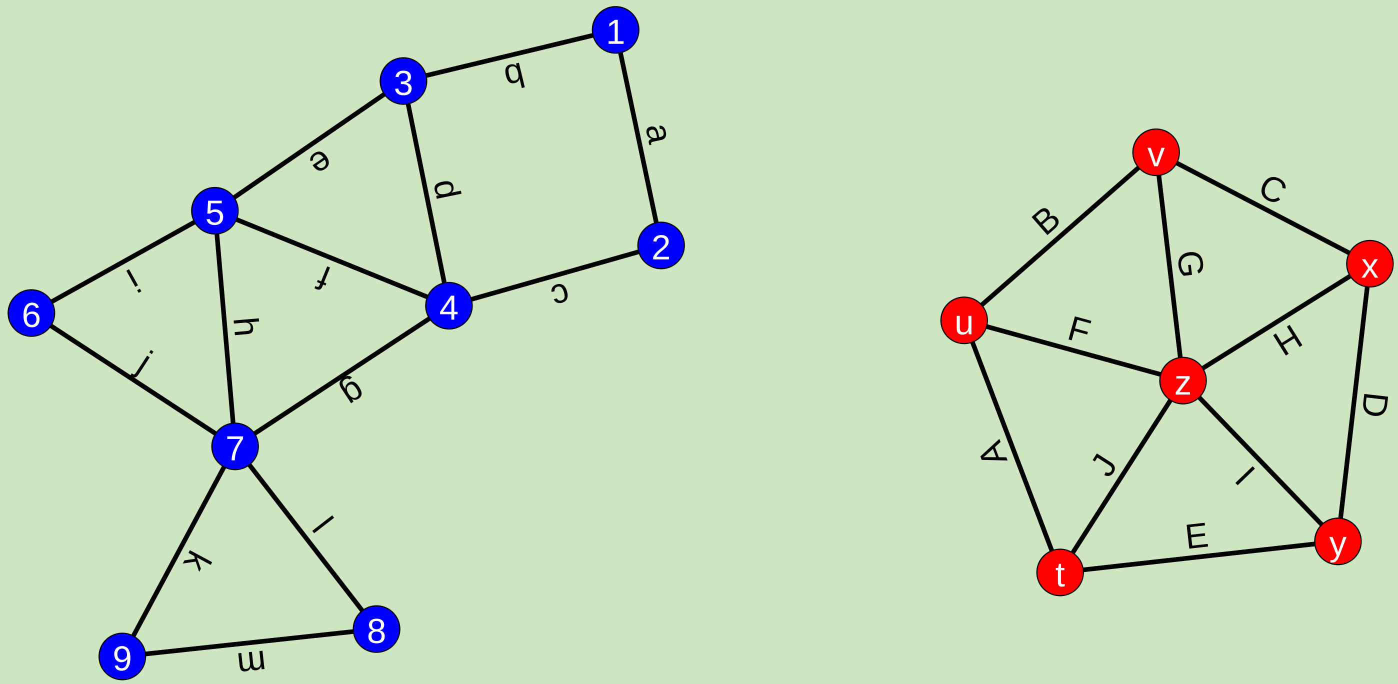


If φ and ψ are bijections and the condition hold in both direction we get an *isomorphism* of graphs $\mathcal{G}$ and $\mathcal{H}$. We denote the weak isomorphism by $\mathcal{G} \sim \mathcal{H}$; and the (strong) isomorphism by $\mathcal{G} \approx \mathcal{H}$. It holds $\approx \subset \sim$.

An *invariant* of graph is called each graph characteristic that has the same value for all isomorphic graphs.

EulerGT

Homomorphism

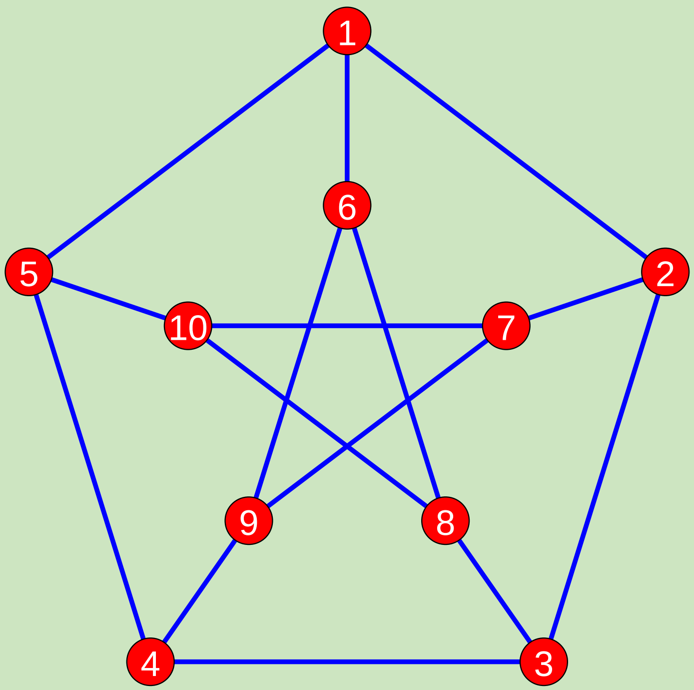


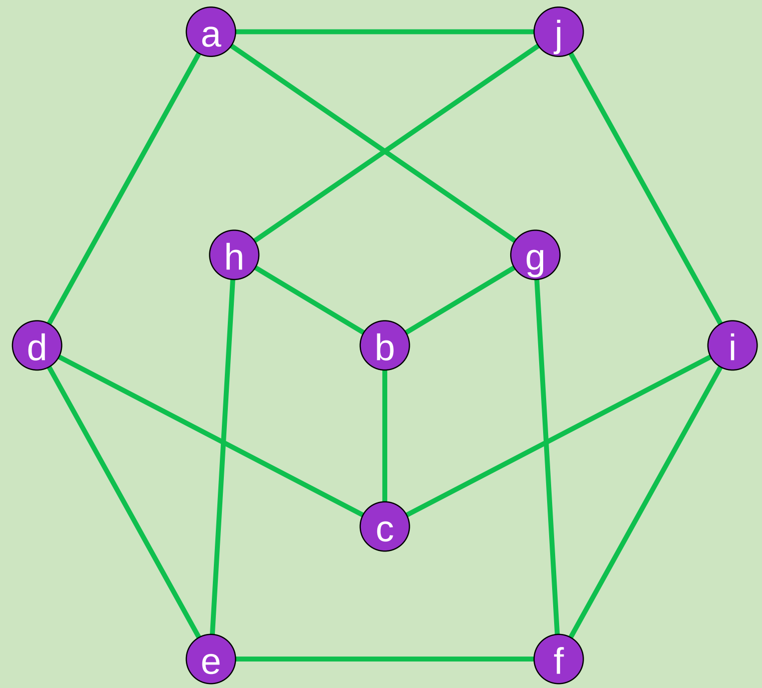
φ	1	2	3	4	5	6	7	8	9
	t	y	z	x	v	u	z	y	t

ψ	a	b	c	d	e	f	g	h	i	j	k	l	m
	E	J	D	H	G	C	H	G	B	F	J	I	E

Pajek: homoEna.net

Isomorphic graphs





φ	1	2	3	4	5	6	7	8	9	10
	b	h	j	a	g	c	e	i	d	f

Pajek: izoPet.net

Clusters, clusterings, partitions, hierarchies

A nonempty subset $C \subseteq \mathcal{V}$ is called a *cluster* (group). A nonempty set of clusters $\mathbf{C} = \{C_i\}$ forms a *clustering*.

Clustering $\mathbf{C} = \{C_i\}$ is a *partition* iff

$$\cup\mathbf{C} = \bigcup_i C_i = \mathcal{V} \quad \text{and} \quad i \neq j \Rightarrow C_i \cap C_j = \emptyset$$

Clustering $\mathbf{C} = \{C_i\}$ is a *hierarchy* iff

$$C_i \cap C_j \in \{\emptyset, C_i, C_j\}$$

Hierarchy $\mathbf{C} = \{C_i\}$ is *complete*, iff $\cup\mathbf{C} = \mathcal{V}$; and is *basic* if for all $v \in \cup\mathbf{C}$ also $\{v\} \in \mathbf{C}$.

Contraction of cluster

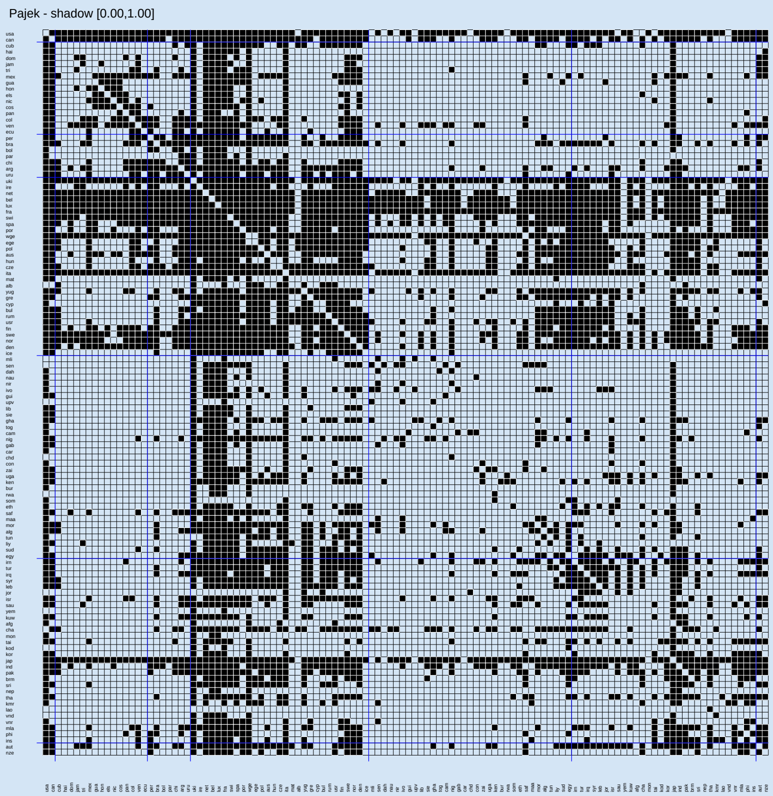
Contraction of cluster C is called a graph $\mathcal{G}/C$, in which all vertices of the cluster C are replaced by a single vertex, say c. More precisely:

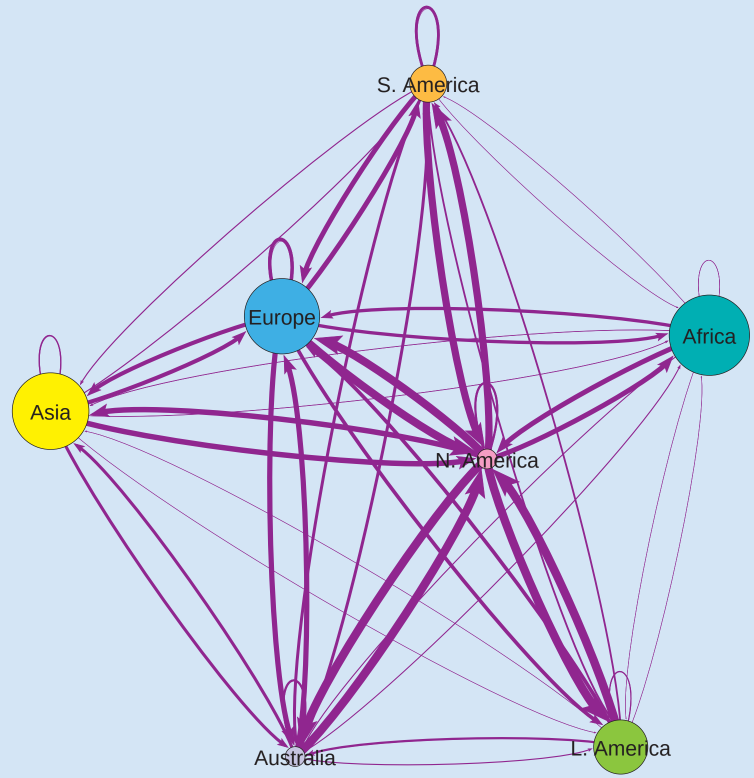
$\mathcal{G}/C = (\mathcal{V}', \mathcal{L}')$, where $\mathcal{V}' = (\mathcal{V} \setminus C) \cup \{c\}$ and $\mathcal{L}'$ consists of lines from $\mathcal{L}$ that have both end-vertices in $\mathcal{V} \setminus C$. Beside these it contains also a 'star' with the center c and: arc (v, c), if $\exists p \in \mathcal{L}, u \in C : p(v, u)$; or arc (c, v), if $\exists p \in \mathcal{L}, u \in C : p(u, v)$. There is a loop (c, c) in c if $\exists p \in \mathcal{L}, u, v \in C : p(u, v)$.

In a network over graph $\mathcal{G}$ we have also to specify how are determined the values/weights in the shrunk part of the network. Usually as the sum or maksimum/minimum of the original values.

`Operations/Shrink Network/Partition`

Contracted clusters – international trade





Snyder and Kick's international trade. Matrix display of dense networks.

$$w(C_i, C_j) = \frac{n(C_i, C_j)}{n(C_i) \cdot n(C_j)}$$

Computing the weights

```
File / Pajek Project File / Read [SaKtrade.paj]
Net / Transform / Remove / Loops [No]
Net / Transform / Edges -> Arcs [No]
Operations / Shrink Network / Partition [1][0]

                1    2    3    4    5    6    7
-------------------------------------------
 #usa  1.       2   30   13   56   42   45    4
 #cub  2.      30   74   25  196   20   37   12
 #per  3.      12   28   33  124   16   36    5
 #uki  4.      55  217  130  695  427  483   41
 #mli  5.      42    8   14  406  122  117   11
 #irn  6.      43   37   43  444  142  307   30
 #aut  7.       4    4    5   39    9   30    2

Partition / Make Permutation
[select partition (Sub)continents]
Operations / Functional Composition / Partition*Permutation
Partition / Count
Partition / Make Vector
Operations / Vector / Put loops
```

…Computing the weights

		1	2	3	4	5	6	7
#usa	1.	4	30	13	56	42	45	4
#cub	2.	30	89	25	196	20	37	12
#per	3.	12	28	40	124	16	36	5
#uki	4.	55	217	130	723	427	483	41
#mli	5.	42	8	14	406	155	117	11
#irn	6.	43	37	43	444	142	337	30
#aut	7.	4	4	5	39	9	30	4
count		2	15	7	29	33	30	2

```
Vector / Create Identity Vector [7]
[select as second vector From partition ...]
Vectors / Divide First by Second
Operations / Vector / Vector # Network / input
Operations / Vector / Vector # Network / output
[edit partition - rename vertices]
```

		1	2	3	4	5	6	7
N.Am	1.	1.00	1.00	0.93	0.97	0.64	0.75	1.00
L.Am	2.	1.00	0.40	0.24	0.45	0.04	0.08	0.40
S.Am	3.	0.86	0.27	0.82	0.61	0.07	0.17	0.36
Euro	4.	0.95	0.50	0.64	0.86	0.45	0.56	0.71
Afri	5.	0.64	0.02	0.06	0.42	0.14	0.12	0.17
Asia	6.	0.72	0.08	0.20	0.51	0.14	0.37	0.50
Ocea	7.	1.00	0.13	0.36	0.67	0.14	0.50	1.00

In **Pajek** sequences of commands can be combined into a macro command using

`Macro / Record`

and

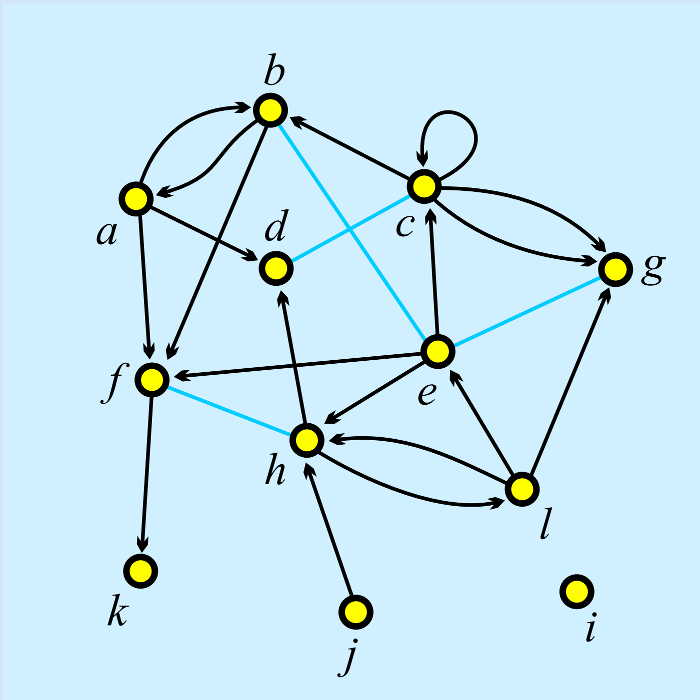
`Macro / Recording...`

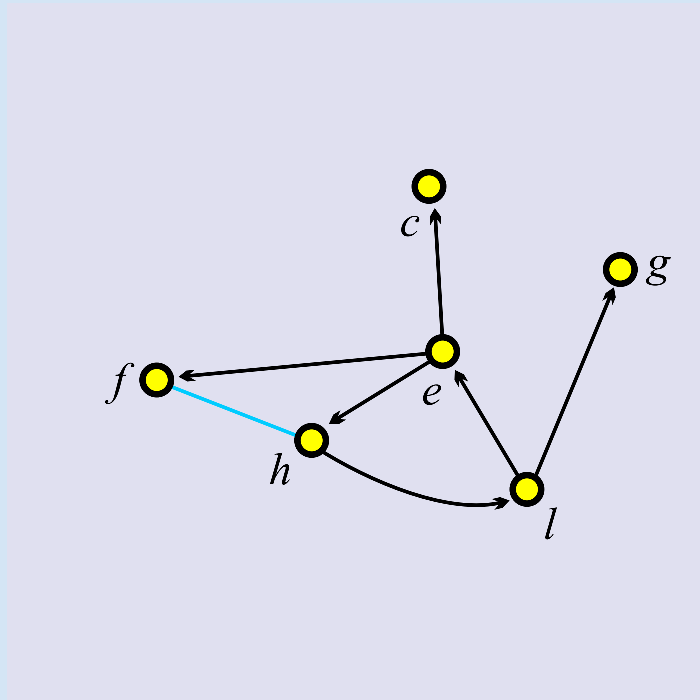
The macro can be activated by

`Macro / Play`

The sequence for computing the weights $w(C_i, C_j)$ is saved in the macro `weights`.

Subgraph

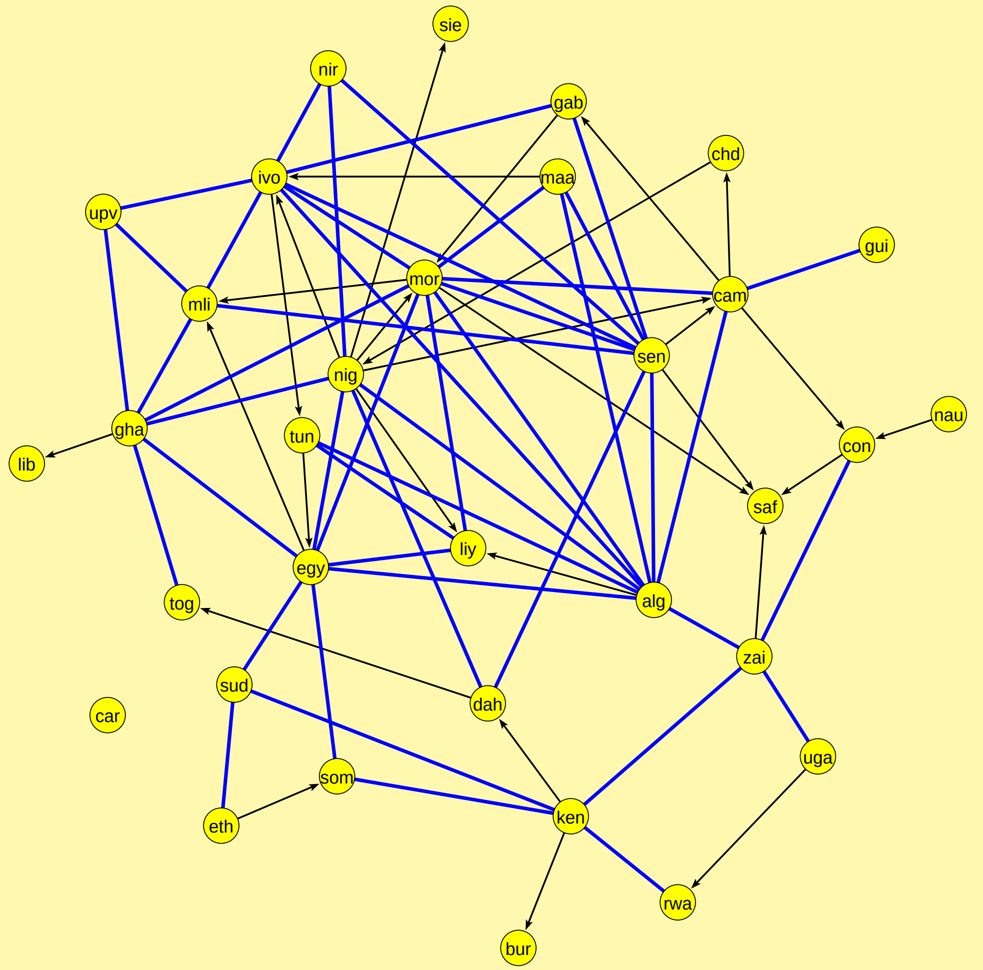




A *subgraph* $\mathcal{H} = (\mathcal{V}', \mathcal{L}')$ of a given graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ is a graph which set of lines is a subset of set of lines of $\mathcal{G}$, $\mathcal{L}' \subseteq \mathcal{L}$, its vertex set is a subset of set of vertices of $\mathcal{G}$, $\mathcal{V}' \subseteq \mathcal{V}$, and it contains all end-vertices of $\mathcal{L}'$.

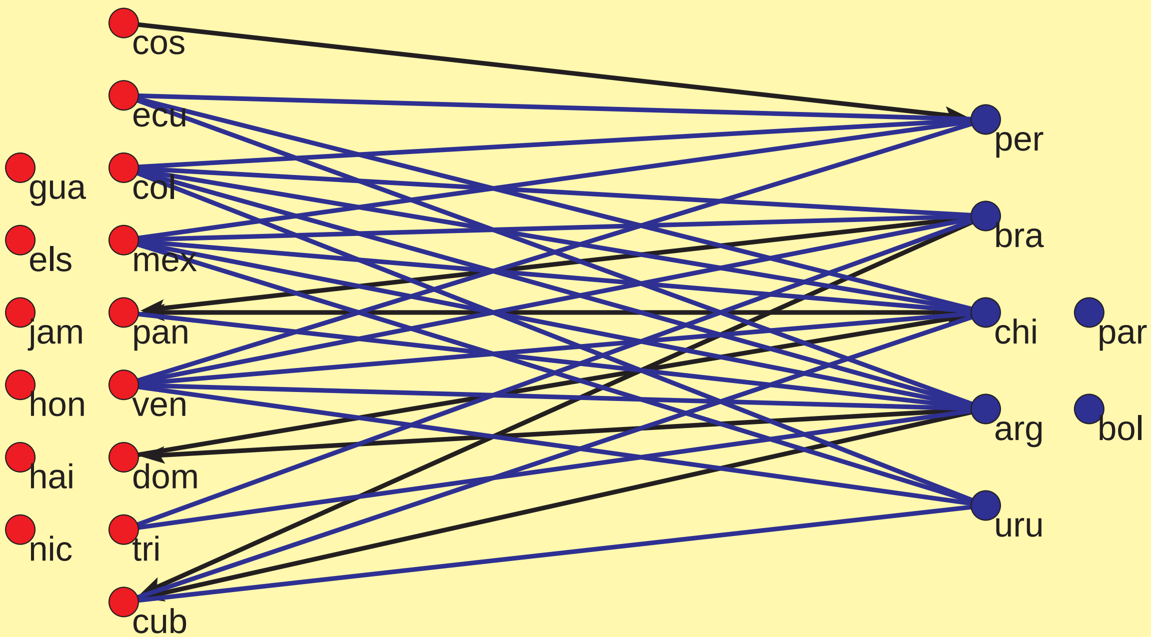
A subgraph can be *induced* by a given subset of vertices or lines. It is a *spanning* subgraph iff $\mathcal{V}' = \mathcal{V}$.

Cut-out – induced subgraph: Snyder and Kick – Africa



`Operations/Extract from Network/Partition 6`

Cut-out: Snyder and Kick
Latin America : South America




```
Operations/Extract from Network/Partition   3,4
Operations/Transform/Remove lines/Inside clusters   3,4

[Draw] Move/Grid
```